Crowdsourced Mapping and Analysis of Literary Character Networks

Syed Ahmed

Master Of Science

School of Computer Science

McGill University

Montreal,Quebec

2014-03-03

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Master of Science

DEDICATION

This document is dedicated to my parents.

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my deepest gratitude to my academic supervisor Professor Derek Ruths without whose guidance and supervision, This work would not be possible. Derek gave me enough freedom to implement my own methods and constantly gave constructive feedback on my work. Any amount of words fall short to express the gratitude that I have for him. He is a true inspiration. I would also like to thank Professor Andrew Piper for his invaluable contributions to this work. He has been the guiding stone behind this project. His insights on the literary aspects of this project have been the single most significant factor in shaping this work. I also want to extend my thanks to Faiyaz Zamal for work on random models. His contributions and insights during our meetings helped me to gain a deeper understanding of the problem we were solving. Finally, I would like to thank my friends who have kept me sane all times and being there for me. You guys are the best.

ABSTRACT

Literary plot and the development of character are strongly driven by social context within the domain of literary theory. Capturing the constructed social world within literature can therefore greatly facilitate our understanding of the essential elements of a story and even a genre. While there have been efforts in the past to derive the social networks embedded in literary works, none have demonstrated the ability to reliably and accurately infer character interactions at scale. In this work, we present a novel crowdsourcing based method for rapidly mapping literary character networks. We apply our method to a corpus of detective and short fictional works and present a number of novel network statistics to capture literary features specific to each genre. These measures provide insights into how the use of social structures in detective fiction differ from those in the broader class of short fiction. As we report, stories that are concerned with narratives of detection are significantly more aligned with features of extensive rather than intensive social relationships, indicating a strong connection between open social networks and the narration of fact-finding. The results are supported by the use of specially-designed random network models which show that short fiction character networks are distinctly "man made".To further our claim, we build a classifier based on features derived from the network structure of the stories. Our proposed technique yields highly accurate character interaction sets with an F1 score of 0.91 which is significantly more accurate than existing methods.

RÉSUMÉ

L'intrigue littéraire et le traitement des personnages sont fortement motivés par les interactions sociales dans la théorie de la littérature. Capturer l'ensemble d'un monde social dans la littérature peut donc faciliter grandement notre compréhension des éléments essentiels d'une histoire et même d'un genre complet. Bien qu'il ya eu des efforts dans le passé pour extraire et analyser la complexité des interactions dans les œuvres littéraires, aucun n'a encore fait ses preuves lorsqu'il s'agit d'inférer de manière fiable et précise les interactions entre personnages à grande échelle. Dans ce travail, nous présentons une nouvelle méthode de crowdsourcing permettant d'analyser rapidement des réseaux d'interactions de personnages littéraires. En appliquant notre méthode à un corpus de romans et d'œuvres de fiction courtes, nous obtenons un grand nombre de statistiques permettant de capturer les caractéristiques littéraires spécifiques à chaque genre. Ces mesures permettent de mieux comprendre comment l'utilisation de structures sociales dans la fiction policière diffère de celle de la catégorie plus générale de fiction courte. Au cours de notre expérience, nous avons constaté que les récits qui se rapprochent le plus du genre policier se distinguent clairement des genres où les interactions sociales sont très nombreuses, indiquant une forte connexion entre un réseau social étendu et la narration de faits. Les résultats sont corroborés par l'utilisation de modèles de réseaux aléatoires spécialement conçus, montrant que les interactions entre personnages dans la fiction sont clairement voulus et non aléatoires. Afin de confirmer nos résultats, nous avons entrepris de construire un classificateur basé sur les caractéristiques dérivées de la structure du réseau des histoires. Notre technique aboutit à des résultats très prometteurs avec un score F1 de 0.91, nettement plus précis que les méthodes existantes.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
Introduction

Characters and their interactions are a fundamental feature of literature. Characters provide us with the opportunity to identify with other imaginary human beings and the ability to model social relationships. Within the domain of literary theory, there is a rich tradition of scholarship on the meaning of character, from the analysis of character typologies (e.g., [28]), to the study of fan fiction and the afterlife of character (e.g., [5]), to more recent work on the affective and cognitive identifications with characters on the part of readers [35, 33]. What unites much of this work is an emphasis on understanding character in the singular.

The vast majority of stories contain more than one trivially engaged character. This is true across medium (e.g., short story, movie, graphic novel) and genre (e.g., mystery, thriller, romance). Thus, it would seem that stories depend on a constructed social universe to achieve plot progression, character development, and myriad other story-telling devices. From a structural perspective, this social universe can be represented as a social network of characters and interactions between them which forms and is revealed over time, providing the scaffolding for character-character interactions.

Introducing social network analysis into the study of character interaction allows us to model both the larger social universe to which characters belong as well as the dynamic evolution of their interactions. Character networks can help us see how characters are not simply types or themes or even vehicles for the affective connections between readers and texts, but instead windows into the social imaginings of writers, periods, or genres.

In this thesis we present a high-throughput technique for mapping the character interaction networks in literary works. Unlike past approaches, our method uses crowdsourcing in order to map interactions in a way that most closely resembles the way in which they are experienced by a human reader — one of the most important (if not the primary) interpretive perspectives to assume. We applied our method to 41 short fictional works and found that our method achieved an F1 score of 0.913 in capturing the interactions detected by a controlled cohort of readers. This represents a significant improvement over existing approaches in several ways.

In order to highlight the merits of our approach and this direction of inquiry, we conduct an analysis of the short fiction corpus that we sequenced in the validation phase, investigating the social signatures of the sub-genre, detective short fiction. We propose a number of novel static and dynamic network statistics which strongly support the thesis that detection narratives explore a distinctly more open social universe than short fiction in general. We also evaluate our generated networks against various random network models and show that the highlighted features in our network are significant Finally, we build a classifier based on the network features and show that it can classify detective fiction from short fiction with 68% accuracy.

1.1 Contributions of the Thesis

While network analysis holds great promise for the study of literature, only very initial attempts have been made to map and study the structure of such social networks [1, 2, 26, 12, 22]. Our work is meant to take this research to a new level of sophistication in three distinct ways.

1. **Reliable extraction of literary networks using crowdsourcing:**

   Deriving social interactions from prose texts in a reliable way is a highly complex act for both humans and machines. It has to-date largely confounded automated methods and is insufficiently analyzed in manual encoding methods [2, 12]. In this work, we propose an approach that uses crowdsourcing to massively parallelize the reading and coding of text by human Amazon Mechanical Turk workers. Crucially, we have found that our approach achieves a high level of accuracy (F1 score of 0.913) which far exceeds any reported by existing automated (or manual) approaches.

2. **Novel network statistics for character networks :** With the exception of Elson et al., prior work in this field has not produced generalizable measures for the study of literary phenomena [12]. Existing research has either focused on the mapping problem without addressing larger literary questions or on literary questions without supporting robust or large-scale quantitative data [1, 26, 22]. Here we bridge these pursuits by presenting, in addition to our novel interaction mapping system, a suite of new network statistics that measure significant social features which contribute to the meaning of a particular genre, in our case detective fiction. As we show, applying social network analysis to the study of literature can produce truly novel insights about the nature and social function of different genres.

3. **Ability to scale to other genres/novels :** Little existing work has explored the value of understanding character networks across a broad array of texts. Besides Elson's study of 60 Victorian novels, analysis has taken place on a maximum corpus size of no more than three texts [12]. Moreover, existing automated methods, while promising, are severely hampered in their ability to correctly map the interaction structures in a text, making large-scale studies impractical at present. Here, we

present analysis of 41 separate works allowing us to compare an entire anthology of a single genre (detective fiction) against a control group of more general canonical short stories. Thus, our work does offer a feasible method for scaling this research to large corpora with appropriate resources.

Overall, we consider the present work to be an exciting and necessary step towards large-scale studies of social structures in literature (and in other media). The proposed mapping method makes a significant improvement to existing interaction mining accuracy without making serious compromises to scalability. Our subsequent analysis of detective and short fiction reveals that characterizing the social dimensions of story-telling is an important part of understanding how different genre and stories function.

1.2 Outline

The reminder of this thesis is organized as follows:

Chapter 2 discusses related work which has been done in analyzing literature using social network analysis. It also gives background information about the random models used in our study and the various classifiers used to evaluate our results. Finally we discuss the different measures we use to evaluate our performance and their interpretation from an interaction network perspective.

Chapter 3 discusses the corpus used in our study and the crowdsourcing method for mapping the interaction network using AMT. We also discuss the post-processing steps required after getting the raw data from AMT to filter out interactions with low agreement. Finally we give the description of the network statistics we measure for each of the network and their interpretation from a literary perspective.

In Chapter 4 we discuss the reliability of our method for extracting interaction networks. We give the sensitivity and specificity scores for various coverages and discuss the comparison between the generated networks and random models using the networks stats that we described in Chapter 3. We also report the performance of various classifiers and compare them to our method.

In Chapter 5 we discuss the literary insights from our analysis. We also discuss future work in augmenting the system, making it more automated and mitigating some of the bottlenecks.

CHAPTER 2
Background

2.1 Related Work

There have been some initial attempts to introduce social network analysis into the study of literature. Character networks have been studied within three major European epics (The Iliad, Beowulf, Tain Bo Cuillange) to understand their relation to contemporary models of social networks [22]; an abridged version of a single well-known literary work (Alice in Wonderland) to test differences between interactions and observations on character centrality [1]; nineteenth-century novels to understand the correlation between dialogue and setting [12]; and the genre of classical drama to better understand the notion of tragic conflict [26].

Each of these works has added to our understanding of the relationship between character and literary form in important ways. And each also faces significant challenges at different phases of the process. For projects that rely on the manual encoding of character interactions (see [22, 1, 26]), insufficient reflection has been given either to the generalizability of the process or the problem of reliability. Manual encoding of character interactions in long literary works is an extremely time-consuming process, which prohibits scalability. It is also highly subjective. "Interaction," however rudimentary a concept, is neither straightforward nor universally identifiable. Agarwal et al. have proposed an important distinction between social interactions and directed observations, while has focused on dialogue as a unique form of interaction [1, 12]. In addition to the variability of what constitutes an interaction, studies have so far not addressed the subjectivity of the manual encoding process. Worth mentioning is our own experience with "expert" human coders during the early stages of this project. Using cohorts of student coders, we found their agreement on interactions averaged near 50%, which served as a significant motivation for this present crowdsourced approach.

The automated extraction of networks has so far fared with limited success. Agarwal et al. reports a maximum F1 score of 0.61 using natural language processing, while Elson's approach to extracting dialogue and attaching it to speakers reports F1 of 0.67 [2, 12]. Social interactions between characters are highly complex acts - they can have a great deal of variability in the naming conventions used to identify characters or in the subtleties of what constitutes an interaction, making it a challenging object to capture in either a manual or machine-learning way.

Finally, with the exception of Elson et al., prior work on literary character networks has so far not produced generalizable measures or insights into literary phenomena [12]. In the case of Mac Carron et al., while it is an interesting question to study the extent to which fictional character networks correspond to real social networks, using contemporary network features to understand historical texts that span a great deal of both time and space (ancient Greece to the medieval British isles) is insufficiently grounded in the realities of historical context and difference [22]. Similarly, referring to social networks from the genre of epic texts as "mythological" represents a significant confusion of literary terminology and the nature of these texts.

Studying literary phenomena requires a careful understanding of the distinctions within that field of study. While Elson et al. limit themselves to one form of interaction, that of dialogue, they offer robust findings about the non-correlation between the amount of dialogue and the setting of Victorian novels [12]. This interestingly contradicts much accepted literary wisdom that urban novels indicate an increase in the number of characters and a decrease in social connectivity (as a form of social alienation that comes with modern urbanization). Agarwal et al. introduces a potentially productive distinction between social interactions and directed observations, and shows how this impacts notions of character centrality, but offers no larger literary claims about the significance of this distinction, which would indeed be interesting to pursue [1]. Moretti uses no network measures to ground his insights [26].

2.2 Literary perspectives on character

A great deal of literary theory addresses the meaning of character for narrative structure or reader's imaginative identification with texts (whether fictional or non-fictional). For the Russian formalists, characters were thought of principally as "types" that served to give meaning to a particular genre, such as highly formulaic ones like the fairy tale [28]. In contrast, for the school of French structuralists, character was understood as nothing more than an aggregation of rhetorical features a character was not to be confused with a real person, but was instead the sum of the descriptive language used to convey that character [32]. Later approaches attempted to integrate these thematic and mimetic understandings of character the way characters often function as thematic types but are also constrained by their real-world nature [27].

More recent research on character has emphasized its affective or identificatory function for reading. Characters are the vehicles through which we emotionally identify with and invest in stories. The example of fan fiction, which dates at least back to the eighteenth century (Brewer), is a good example of characters efficacy in generating readers' responses to literary material. Character has also been understood to be the means through which readers in the past came to terms with new social experiences such as the introduction of consumer culture (Lynch). Newer research drawing on cognitive psychology and theories of mind has emphasized the way characters are useful for modeling cognitive behavior [33, 35]. We enjoy reading about characters because it is a way for us to navigate the complexity of other people's minds. As Lisa Zunshine writes regarding detective fiction, "We can thus enjoy being lied to in the highly structured world of a murder mystery because it offers us a safe setting in which to relieve our anxieties about the uncertainties and deceptions of real life" (122).

Where our work and other recent attempts at introducing social network analysis to the study of literature differs from this tradition is through the emphasis on dynamic interactions as a key to understanding the narrative function of character. Whether exploring the afterlife of fan fiction, theories of mind, affective identification, or the typologies of character, what all of this work has in common is an emphasis on an understanding of character in the singular. Even recent work on character space, which models characters through their descriptive prominence, does not account for characters through a sense of their interconnectedness [34]. Social network analysis by contrast argues that the meaning of any character is a function of his or her relationships with respect to all of the other characters introduced over the course of a story. Characters offer a way to study not simply types or themes or affective connections between readers and imaginary people, but the ability to understand the social imaginings of writers and genres.

2.3 Crowdsourcing & AMT

Crowdsourcing is a distributed problem solving mechanism which is being recently seen as an alternate to worker-employee type of model. Jeff Howe who coined the word defines crowdsourcing as [20]

> "Simply defined, crowdsourcing represents the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call. This can take the form of peer-production (when the job is performed collaboratively), but is also often undertaken by sole individuals. The crucial prerequisite is the use of the open call format and the large network of potential laborers."

Crowdsourcing differs from a standard worker-employee model as the participation is mostly voluntary and the time of participation is also not mandated. The timeframe may vary between one-time to several years and there are no fixed hours. There is a relaxed sense of responsibility and general expectations from the workers are low. The people who participate in a crowdsourced system do it for some sort of gain. It can be economical (getting paid), social (achieving some status) or self-satisfaction (sense of contribution). The task is given is either collaborative (i.e. every person knows and contributes towards a fixed goal) or non-collaborative where the users may not be actively involved in the goals and direction of the project. Crowdsourcing can be applied to a wide array of problems which can vary from simple tasks like identifying objects in images to very complex tasks like neuron mapping [24].

Because of the nature of crowdsourcing, the tasks are designed such that they are engaging, short and require a short timespan from the users. A single task, can be given to multiple people and their responses can be combined. If a task is too large to be completed in a small amount of time, the task must be sub-divided into to smaller sub-tasks which are crowd-friendly.

The applications of crowdsourcing are vast and far-reaching. Online crowdsourcing has been very successful during the past decade. Wikipedia

& Linux are prime examples of collaborative crowdsourcing. In recent years, crowdsourcing has also been applied to create a viable source of capital for various projects. Termed as crowd-funding, this method uses small contributions of capital from various users to fund a specific project. Examples include Kickstarter, IndiGogo.

2.3.1 Amazon Mechanical Turk

Developed by Amazon in 2005, Amazon Mechanical Turk (AMT) is a crowdsource market which lets users create and manage the crowdsourcing of various tasks done by a large community of crowd-workers.

AMT allows two kinds of accounts (1) requesters who post tasks and (2) workers who complete the tasks. Each task called a HIT, created by the requester, can be assigned to one or more workers. The requester assigns the amount of money he is willing to pay for each HIT and also sets the criteria for a worker to be able to accept the hit. The criteria can be based on several parameters such as % of accepted HITs, number of HITs completed and number of rejected HITs. The requester may also set special requirements for the HIT which requires the workers to clear a qualification criteria in order to accept the HIT.

Workers can search for HITs which fit their criteria and select which they want to work on. Each HIT has a time period designated which if exceeded will result in disqualification of that HIT. Once the workers have completed their HIT, the requester can approve or deny their HIT based on the answer or agreement among the workers. The requester can also select to auto-approve after certain time has elapsed. Once a HIT is accepted, the worker gets paid. Amazon charges 10% commission for the task as AMT fee.

There are a special class of workers call “Master” users in AMT which are workers with demonstrated accuracy in specific types of HITs. Amazon charges 20% extra for using masters.

2.4 Levenshtein Distance

The Levenshtein distance is a measure of similarity between two sequences. It is distance between two sequences is defined as the number of edits (inserts, deletes, substitutions) required to change one sequence to another.

The Levenshtein distance between two strings, a and b is given by $\mathrm{lev}_{a,b}(|a|,|b|)$ where

$$\mathrm{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if} \min(i,j) = 0, \\ \min \begin{cases} \mathrm{lev}_{a,b}(i-1,j)+1 \\ \mathrm{lev}_{a,b}(i,j-1)+1 \\ \mathrm{lev}_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

where $1_{(a_i \neq b_j)}$ is the indicator function equal to 0 when $a_i = b_j$ and 1 otherwise. When comparing different strings for similarity, we use the Levenshtein ratio which defined as

$$\frac{|a| + |b| - lev_{a,b}}{|a| + |b|}$$

We use the Levenshtein ratio instead of length as it is normalized over the sum of lengths of both the strings which keeps it stable for larger strings. The ratio has a value of 1 if both strings are similar and 0 otherwise.

2.5 Random Models

In our study, we use variants of two different random network generation models to create synthetic interaction networks for comparison against the AMT generated networks. This section gives a brief introduction to these models.

2.5.1 ER Model

Proposed by Paul Erdos and Alfred Renyi, in the Erdos-Renyi model for random graph generation, each edge is given equal probability of being added to the random graph [13]. Let us consider the generation of a random graph. Suppose we have V nodes and we want to create a random graph with K edges, There are a total of $\binom{V}{2}$ possible edges. In the ER model, each edge has equal probability of being selected. We do a uniform random selection K times among the $\binom{V}{2}$ edges and the resultant set of edges is the random graph. Note that once an edge is selected, it is not removed from the pool of available edges. This means that we could select the same edge multiple times. In this case, the weight associated with that edge increases.

2.5.2 BA Model

The Barabasi-Albert model for random graph generation follows the preferential attachment mechanism in which new nodes have a tendency to connect to nodes with a high degree [3]. The degree distribution from the resulting graph follows a power law. Consider the iterative generation of a random graph using the BA model with V nodes and K edges. For each iteration, we select two nodes and create an edge between them. We do this for K iteration. Initially, all nodes have equal probability of getting an edge. We uniformly choose two nodes (u, v) with a probability $\frac{1}{|V|}$ and create an edge e. Once the edge is created, the probability of both nodes to get selected for the next iteration changes to $\frac{2}{|V|+2}$ thus after n iterations, the probability of node u

getting selected is $\frac{d(u)+1}{|V|+2n}$ where $d(u)$ is the degree of node u in the graph of the $(n-1)^{th}$ iteration. The resultant graph has “hubs” which are connected to many nodes. This model mimics a real world social network where popular people are connected to many other people. Twitter celebrities are hubs which everyone follows. They have a high degree whereas people connected to them, on average, have lower degree.

2.6 Classifiers

This section gives a brief overview of the classifiers we use in our study. Of these, the Naive Bayes and the Labeled Latent Dirichlet allocation (LLDA) are language-based probabilistic classifiers and Support Vector Machine (SVM) is a linear non-probabilistic binary classifier.

2.6.1 Naive Bayes Classifier

The Naive Bayes classifier is a probabilistic classification model which relies on Bayes rule [30]. In our work we use a variation of the Naive Bayes classifier called the Multinomial Naive Bayes.

For our classification exercise, consider we have a number of documents $D = (d_1, d_2 ... d_m)$. Let $C = (c_1, c_2 ... c_n)$ be the set of classes to which these documents belong and $x^i_1, x^i_2 .... x^i_k$ represent the tokens of the document d^i.

To classify a new document d, we need to find its probability of being in a class $c \in C$ and choose the class which maximizes the probability $P(c|d)$.

$$\hat{c} = \underset{c}{\operatorname{argmax}}\, P(c_i|d), c_i \in C$$

consider $P(c_i|d)$ as per Bayes rule we know

$$P(c_i|d) = \frac{P(d|c_i)P(c_i)}{P(d)}$$

To calculate $P(d|c_i)$, we make the Naive Bayes assumption that all individual tokens are independent of each other. We tokenize our document to individual tokens(words) and remove any stopwords. To represent the whole document, we use a bag-of-words model which counts the occurrence of each token in the document. $P(d)$ is constant for all classes and can be ignored. We have

$$P(c_i|(x_1, x_2, ..., x_k)) \propto P((x_1, x_2, ..., x_k)|c_i)P(c_i)$$

Applying the Naive bayes assumption

$$P(c_i|(x_1, x_2, ..., x_k)) \propto P(x_1|c_i)P(x_2|c_i)...P(x_k|c_i)P(c_i)$$

To calculate $P(x_i|c_i)$, we combine all the training documents into one big bag-of-words which has the frequency of each word and $P(c_i)$ is the relative frequency of the class c_i in the training set.

$$P(x_i|c_i) = \frac{\text{frequency of } x_i \in \{d_i \mid Class(d_i \in D) = c_i\}}{\text{sum of all tokens in } d}$$

$$P(c_i) = \frac{\text{\# of documents of class } c_i}{\text{total number of documents}}$$

2.6.2 LLDA Classifier

Labeled Latent Dirichlet Association (LLDA) is a variant of Latent Dirichlet Association (LDA) which is a probabilistic generative topic model. LDA is used to automatically discover topics in a set of documents [4]. It is an unsupervised learning method meaning the algorithm finds the topics and the words belonging to the topics automatically. The only input is the number of

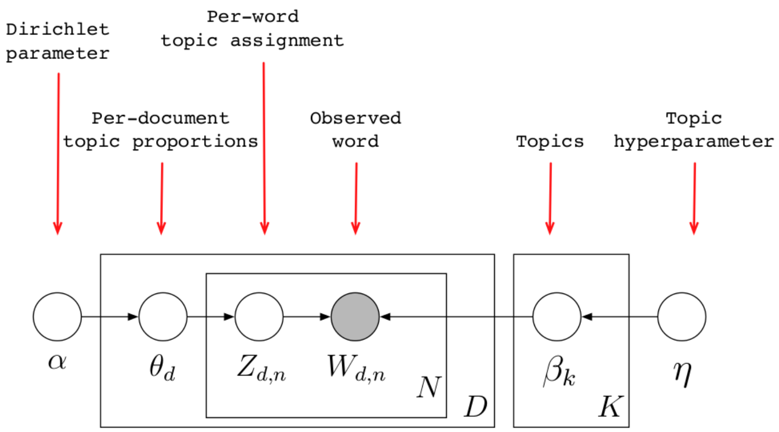


Figure 2–1: LDA model represented using the plate notation. Figure from [4]

topics to find. LDA is an unsupervised method which means it clusters words into abstract topics. If a topic model is already known, a variation of the LDA called labeled LDA is used which uses the prior topic model for classification. [29]. The labeled LDA variation is where the algorithm is made supervised by supplying topic labels.

The idea behind LDA is that each document is generated from a mixture of topics. Consider a set of M documents $D = (d_1, d_2....d_M)$ called "corpus" where each document is in turn a collection of words $d = (w_1...w_N)$. Let $Z = (z_1, z_2..., z_k)$ be the set of topics for D. α, β and ζ are constants for random distributions.

The only observed variables we have are the words in the documents themselves. If we work backwards from the words, LDA assumes that each word in each document is derived from some distribution of words over the topics z_n. For example we may have a topic "mammals" having words like "mouse" with probability 0.03 and "eggs" with probability 0.001. Each document is generated by a mixture of such topics. Now the topics themselves are derived from a topic distribution denoted by θ_d. There will be one topic mixture for each document in our set. The topic mixture is a Dirichlet distribution over all the possible topics.

Figure 2–1 shows the LDA model using plate notation. The nodes (circles) represent hidden variables and edges (arrows) represent dependency of one random variable on the other. The shaded nodes represent observable variables and the non-shaded nodes represent hidden variables. The boxes around the nodes represent replicated structures basically that node is replicated X number of times where X is denoted and the bottom of each plate. The following are the probability distributions for various variables in LDA.

1. The topic distribution for each topic β_k. This is a Dirichlet with a parameter η given by $P(\beta_k \mid \eta)$. The topic distribution is independent as it only depends on the Dirichlet parameter η.
2. The topic mixture distribution for each document d in the corpus. This is also a Dirichlet with parameter α given by $P(\theta_d \mid \alpha)$.
3. The topic assignment $z_{d,n}$ for each word of the document given by $P(z_{d,n}|\theta_d)$.

4. Finally, we have the probability of the n^{th} word from a document d, $w_{d,n}$ given a topic mixture for that document z_d and the overall topic distribution β_k given by $P(w_{d,n} \mid z_{d,n}, \beta_k)$.

The joint probability distribution of observed and hidden variables over all the documents in the LDA is given by:

$$P(\mathbf{w}, \mathbf{z}, \theta, \beta|\alpha, \eta) = \left(\prod_{k=1}^{K} P(\beta_k \mid \eta)\right)\left(\prod_{d=1}^{D} P(\theta_d \mid \alpha)\left(\prod_{n=1}^{N} P(z_{d,n}|\theta_d) P(w_{d,n} \mid z_{d,n}, \beta_k)\right)\right)$$

The generative model for LDA generates a set of documents from the topic mixture distribution and the distribution of words in a topic. The following are the steps that LDA uses to generate a single document. The assumption here is that the topic distributions β_k are already known.

1. Choose the number of words for the document N
2. Choose the topic mixture θ from a Dirichlet distribution $Dir(\alpha)$
3. For each of the N words w_n
   (a) Choose a topic z_n from the topic mixture θ
   (b) Each topic has a probability distribution over the words it can produce, we now choose a word w_n based on this distribution conditioned on the topic z_n, $P(w_n|z_n, \beta_k)$

Example: Let us consider generation of a document with the LDA model

1. Choose $N = 5$, the number of words in the document
2. From the topic mixture θ we choose the topics "Arts" with probability 1/3 and "Education" with probability 2/3
3. Generate each word w_n from the topics by selecting topics based on their probabilities
   (a) "Education" selected word produced: "School"
   (b) "Arts" selected word produced: "Music"
   (c) "Education" selected word produced: "Students"
   (d) "Education" selected word produced: "Public"
   (e) "Arts" selected word produced: "Actor"
4. The final document produced "School Music Students Public Actor"

Given an initial estimate of the model parameters η and α the topic distribution can be inferred iteratively using various methods like gibbs sampling, variational bayes approximation and expectation propagation [8, 9, 25]. Note that LDA does not consider the order of words, it uses a bag-of-words representation for the documents similar to the Naive Bayes classifier.

Learning: We have a set of topics Z and a set of documents D. We want to be able to assign the topics to documents such that it follows the LDA topic generation model. We use the covered gibbs sampling method for inferring topics [11]. The following illustrates the algorithm

Algorithm 1: LDA using covered gibbs sampling

```
Input: Set of documents $D$
Output: topic assignments $\mathbf{z}$ for each document word $w_n \in d$ for each
        document $d \in D$
Assign random topic to each word in every document ;
initialize ;
$M \leftarrow$ Number of documents $D$ ;
$V \leftarrow$ Number of words in all documents (vocabulary length) ;
$K \leftarrow$ Number of topics ;
$\alpha \leftarrow$ Dirichlet parameter for topic mixture ;
$\eta \leftarrow$ Dirichlet parameter for per topic distribution;
counters $n_{m,z}, n_{z,t}, n_z$ ;
foreach iteration do
    for $m = 0 \rightarrow M - 1$ do
        $d \leftarrow D_m$ ;
        for $n = 0 \rightarrow length(d) - 1$ do
            $t \leftarrow D_{m,n}$ ;
            $z \leftarrow \mathbf{z}_{m,n}$;
            $n_{m,z}[d, z] -= 1, n_{z,t}[z, t] -= 1, n_z[z] -= 1$;
            for $k = 0 \rightarrow K - 1$ do
                $p(z = k) \leftarrow (n_{m,z}[d, t] + \alpha)\frac{n_{z,t}[k,t]+\eta}{n_z[k]+\eta V}$;
            $z' \leftarrow$ sample from $p(\mathbf{z})$ ;
            $n_{m,z}[d, z'] += 1, n_{z,t}[z', t] += 1, n_z[z'] += 1$;
```

Classification: Once we have the topic label for each word in the training set, to classify a new document d_{new} we go through each word in the document and check if it is available in our vocabulary, if present, we calculate the probability of that word belonging to each topic. Finally we make the assumption of independence among words similar to naive bayes and calculate the score for each topic. We select the one with the highest store as the topic for the document d_{new}

2.6.3 Support Vector Machine (SVM) Classifier

The SVM classifier is a non-probabilistic binary classifier [17]. Unlike the previous classifiers, classic SVMs can only classify data into two classes. If each document can be considered as a point in a N dimensional space of features, an SVM tries to create a hyperplane which best separates the data. There may be many hyperplanes which may separate the data, we choose the one which maximizes the distance from both classes. This is called the maximum margin hyperplane.

If we have a set of documents $D = (d_1, d_2 ... d_n)$ which belong to classes $C = (1, -1)$ where each $d_i \in D$ has features $F_i = (f_1, f_2 ... f_n)$ We represent each document as a vector in an N dimensional space. We want to find a $N-1$ dimensional hyperplane which separates points with $c_i = 1$ with $c_i = -1$.

The equation for a hyperplane can be written as

$$\boldsymbol{W}.\boldsymbol{F_i} - b = 0$$

where $\boldsymbol{W}$ is a vector normal to the plane and b is a constant that determines the distance from the origin. We want to get the value of $\boldsymbol{W}$ and b that maximizes the margin between the data. The margin can be represented by two hyperplanes which are parallel to the current hyperplane but on opposite directions of it.

$$\boldsymbol{W}.\boldsymbol{F_i} - b = 1$$

$$\boldsymbol{W}.\boldsymbol{F_i} - b = -1$$

We also add the constraint that no data points should cross into the margin region so we have the following

$$\boldsymbol{W.F_i} - b > 1\,\forall\, F_i \in C_1$$

$$\boldsymbol{W.F_i} - b < -1\,\forall\, F_i \in C_2$$

The distance between the hyperplanes is given by $\frac{2}{||\boldsymbol{W}||}$. Hence, to achieve maximal margin, we want $||\boldsymbol{W}||$ to be minimum.

So finally we have the following

$$\text{Minimize } ||\boldsymbol{W}||$$

$$\text{given } C_i(\boldsymbol{W.F_i} - b) > 1$$

Here $||\boldsymbol{W}||$ is computationally expensive as it involves a square root, we replace it with $\frac{1}{2}||\boldsymbol{W}||^2$

We now have

$$\text{Minimize } \frac{1}{2}||\boldsymbol{W}||^2$$

$$\text{given } C_i(\boldsymbol{W.F_i} - b) > 1$$

Multi-class SVM: While the generic SVM is a binary classifier, there are various methods to classify multi-class data using SVM. The method used depends on the type of data. If we have a case where the samples in the input data can belong to more than one class, then we build a classifier for each class where training set consists of the documents in the class (positive cases) and documents not in the class (negative cases). We then apply each classifier and select the K-best classes. However, if the data has classes which are mutually exclusive, we apply a one-vs-all approach. We build a classifier for each class like in the previous case but here we choose the class which has the highest score. We then remove that class and recursively keep running the classifier until all classes are exhausted.

Non-linear Classification: In many real world examples, the data to classify is not linearly separable, in such cases, we map the input data space into a linear feature space using some non-linear kernel function. This makes the data linearly separable and we can apply an SVM on this. A commonly used kernel is the RBF kernel(gaussian) which is defined as

$$k(\mathbf{F_i}, \mathbf{F_j}) = \exp(-\frac{1}{2\sigma^2}\|\mathbf{F_i} - \mathbf{F_j}\|^2)$$

where $\|\mathbf{F_i} - \mathbf{F_j}\|^2$ is the squared distance between the feature vectors $\mathbf{F_i}$ and $\mathbf{F_j}$.

CHAPTER 3
Methods & Data

3.1 Datasets

The dataset we collected and used in this study performed three separate functions.

1. It permitted us to select appropriate values for parameters that affected the performance of the character network mapping method (Section 4.1).
2. It enabled us to evaluate the overall performance characteristics of the method (Section 4.2.1).
3. Finally, the dataset served as the subject for the literary analysis we performed to demonstrate the value of large-scale analysis of interaction structures across and between genres (Chapter-5).

Table 3–1: The dataset used for our study. The dataset was taken from "Longman Anthology of Detective Fiction" [23] and includes 21 detective fiction and 20 short fiction works

Detective Fiction

Title (Year)	**Author**	**Word length**
Revised Endinkgs (1998)	Burke,Jan	2267
The House In Goblin Wood (1947)	Carr,JohnDickson	7010
The Witness For The Prosecution (1925)	Christie,Agatha	6382
The Hunt Ball (1943)	Crofts,Freeman Wills	4707
Cold Turkey (1992)	Davidson,Diane Mott	6477
The Speckled Band (1892)	Doyle,Arthur Conan	9898
The Parker Shotgun (1986)	Grafton,Sue	6602
The Gutting Of Couffignal (1925)	Hammett,Dashiell	11717
And Pray Nobody Sees You (1995)	Haywood,Gar Anthony	5005
Chee's Witch (1986)	Hillerman,Tony	3854

Under Suspicion (2000)	Howard,Clark	8510
Deborah’s Judgment (1991)	Maron,Margaret	7319
Sadie When She Died (1972)	Mcbain,Ed	11221
Nine Lives To Live (1992)	Mccrumb,Sharyn	6474
Skin Deep (1987)	Paretsky,Sara	4801
The Purloined Letter (1844)	Poe,Edgar Allen	7198
My Queer Dean (1955)	Queen,Ellery	1958
The Dean Curse (1992)	Rankin,Ian	8074
Missing In Action (2000)	Robinson,Peter	7500
The Haunted Policeman (1938)	Sayers,Dorothy	8455
Inspector Maigret Deduces (1959)	Simenon,Georges	4141

Short Fiction

Title (Year)	**Author**	**Word length**
Sarah Cole (1984)	Banks,Russell	9291
Caviar (1979)	Boyle,T.C.	6954
The Ceiling (2002)	Brockmeier,Kevin	5179
Paul’s Case (1905)	Cather,Willa	8416
The Lost Phoebe (1916)	Dreiser,Theodore	6589
Communist (1985)	Ford,Richard	6838
Tiny, Smiling Daddy (1997)	Gaitskill,Mary	5530
Young Goodman Brown (1835)	Hawthorne,Nathaniel	5219
The Snows of Kilimanjaro (1936)	Hemingway,Ernest	9380
The Real Thing (1892)	James,Henry	10470
A Temporary Matter (1999)	Lahiri,Jhumpa	7291
Two Blue Birds (1927)	Lawrence,D.H.	5537
The Vane Sisters (1959)	Nabokov,Vladimir	5338
The Translation (1993)	Oates,Joyce Carol	8217
Everything That Rises Must Converge (1965)	OConnor,Flannery	6511
The Half-Skinned Steer (1999)	Proulx,E. Annie	6579
My Shape (2004)	Silber,Joan	6452
Generous Wine (1914)	Svevo,Italo	6733
The Private History of A Campaign That Failed (1885)	Twain,Mark	7857
Nineteen Fifty-Five (1981)	Walker,Alice	5613

For the purpose of literary analysis, we required a sample of literary works that spanned comparable genres and were representative of both genres. Our dataset consisted of 41 short stories representing two principal groups shown in Table 3–1 : 21 detective stories taken from the “Longman Anthology of Detective Fiction” [23], a standard handbook in the field that brings together detective fiction written in English between 1844-2002 divided into three subcategories (The Amateur Detective, The Private Investigator, and The Police) and 20 short stories that represent canonical examples of the genre in English from different anthologies across the same time period (1835-2004). In addition to being representative, selections were matched with respect to length: the mean word length of each group was 6,753 and 6,936 words respectively. This dataset proved to be diverse and large enough to also satisfy the requirements for tuning and testing our interaction network mining system.

In order to process these texts, our interaction mapping system, like all others to date, required a dictionary for each work that resolves various aliases of a character to the canonical name of the character itself (used as a proxy for the unique identity of the character) [12]. Accordingly, we produced a dictionary for each of the 41 works in our dataset. These were generated by literature students who read the works and compiled a list of names (aliases) used to refer to each character in the book. It is worth noting these alias lists were later discovered to be somewhat incomplete — as should be expected whenever an interpretive task is undertaken manually. As we will highlight later, our crowdsourcing method provided a way to discover additional aliases that had been missed. This alias discovery functionality is a unique and valuable feature of our interaction mapping system.

3.2 Network Statistics

To capture properties which hold a literary significance in our social networks, we propose various statistics. Given a social network of interactions N where nodes represent the characters and the edges represent the interactions between them, the following is a list of network statistics that we compute for each work in our study.

Number of edges: It is the number of unique interactions which characters have among themselves. This does not account for the frequency of interactions among characters.

$$\text{NumEdges}(N) = |E|$$

Number of nodes: This is the number of unique characters present in the text. This is computed after all the aliases have been resolved.

$$\text{NumNodes}(N) = |V|$$

Average degree: It denotes the average number of unique interactactions characters have over the length of the story. The average degree for the network is given by:

$$\text{AvgDeg}(N) = \frac{|E|}{|V|}$$

Degree-weighted heaviest edge score: This measure checks if the heaviest edge (edge with the highest weight) connects the strongest nodes (nodes with the highest degree). This measure tries to answer if there is a strong interaction between the protagonist and the next most important character. Eg (Holmes and Watson in Sherlock Holmes). It is the ratio of the sum of degrees of nodes connected by the strongest edge to the sum of weights of the strongest nodes in the graph. If $e^x = (u^x_{s_a}, v^x_{s_b})$ is the x_{th} heaviest edge of N which connects $u^x_{s_a}$, the a_{th} strongest node and $v^x_{s_b}$ the b_{th} strongest node, and $d(u)$ denotes the degree of a node u then the Degree-weighted heaviest edge score is defined as:

$$\text{DegEdgeScore}(N) = \frac{d(u^1_{s_a}) + d(v^1_{s_b})}{d(u_1^{x_1}) + d(u_2^{x_2})}$$

Heaviest edge fraction: It is the ratio of the strongest edge to the total number of interactions that occur in the story. This tries to capture how significant is the strongest interaction usually between the protagonist, is when compared to other interactions that occur in the story. If $w(e^1_s)$ denotes the weight of the edge e and $|I|$ is the total number of interactions that happen throughout the story, the heaviest edge fraction is given by:

$$\text{HeaviestEdgeFraction}(N) = \frac{w(e^1_s)}{|I|}$$

Average 2-clustering: It is a measure of dispersion of neighborhood of a particular node. For a given node it is the number of 2-hop connectedness of the neighborhood of that node. It measures the ability of the node to explore the social network.

$$\text{Avg2Clustering}(N) = \frac{1}{|V|} \sum_{x \in V} \frac{|\{(u,v) : u, v \in \mathcal{N}(x), \sigma_{N/x}(u,v) \leq 2\}|}{|\mathcal{N}(x)|(|\mathcal{N}(x)| - 1)}$$

2-clustering along the heaviest edge: This measure calculates how connected the neighbors of the strongest edge in the graph are.

$$\text{Avg2ClusteringHeaviestEdge}(N) = \sum_{x \in \mathcal{N}(u^1_s)/v^1_s} \frac{|\{(x, v^1_s) : \sigma_{N/u^1_s}(x, v^1_s) \leq 2\}|}{|\mathcal{N}(u^1_s)| - 1}$$

Max/avg degree ratio: It is the ratio of the degree of strongest node to the sum of all degrees in the network. This measures captures the importance of a particular character compared to all the other characters in the story.

$$\text{MaxAvgDegreeRatio}(N) = \frac{d(u_c^1)}{\sum_{v \in V} d(v)}$$

Time-to-edge-complete: This measures captures how far into the story do we see the last relation appear. It is the time at which we see the last edge dropped on the graph. Some stories have all the edges dropped early into the story and the story revolves around having interactions between already established relations whereas some stories add the last relation close to the end of the story revealing a surprise relationship. We divide our stories into blocks of approximately 250 words. if b_{max} denotes the maximum blocks of a network N, The story progresses from the first block to b_{max}. If $b_{lastEdge}$ is the block in which we see the last relation(edge) being added to the network, the time to complete edge is defined as:

$$\text{TimeToCompleteEdge}(N) = \frac{b_{lastEdge}}{b_{max}}$$

Density: The density of a graph is a measure of how close the graph in becoming a complete graph. It is the ratio of the edges present to all possible edges in the network.

$$\text{Density}(N) = \frac{2|E|}{|V|(|V|-1)}$$

Degree-center neighborhood fraction: It is the ratio of the degree of the strongest character (protagonist) by the total number of edges present. It captures how many interactions directly involve the protagonist.

$$\text{DegreeCenterNeighbourFraction}(N) = \frac{d(u_c^1)}{|E|}$$

Diameter It is defined as maximum length of the the longest shortest path between any two vertices in N.

$$\text{Diameter}(N) = \max_{u,v \in V} \sigma_N(u, v)$$

Closeness vitality: Calculated for the strongest node(protagonist) it is the change in the sum of distances between all node pairs when excluding the node that it is being computed for. It captures the effect of removal of the central character in the story. If the network is very sparse, removing the central character may make many of the nodes not being able to connect to other nodes.

$$\text{ClosenessVitality}(N) = \sum_{x \neq y \in V/u_1^x} \frac{\sigma_{N/u_c^1}(x, y) - \sigma_N(x, y)}{(|V| - 1)(|V| - 2)}$$

Time-to-node-complete: This measure captures how far into the story do we see the last character being introduced. We calculate this similar to Time-to-edge-complete, if b_{max} denotes the maximum blocks of a network N, and $b_{lastNode}$ is the block in which we see the last character(node) being added to the network, the time to complete node is defined as:

$$\text{TimeToCompleteNode}(N) = \frac{b_{lastNode}}{b_{max}}$$

time-to-interaction-complete: This measure captures how far into the story we see the last interaction. This is similar to the Time-to-complete-edge, the difference is that this captures when in the story do we see two characters interact last. Here we are concerned with when the last weight to an existing edge is added. In cases the last edge may be the same as last interaction. If b_{max} denotes the maximum blocks of a network N, and $b_{lastInteraction}$ is the block in which we see the last interaction being added to the network, the time to complete node is defined as:

$$\text{TimeToCompleteInteraction}(N) = \frac{b_{lastInteraction}}{b_{max}}$$

Average weight: It is the ratio of sum of weights of all the edges to the number of edges.

$$\text{AvgWeight}(N) = \frac{\sum_{e \in |E|} w(e)}{|E|}$$

Heaviest edges ratio: It is the ratio of the weights of the heaviest edge to the 2_{nd} heaviest edge. This tends to capture the strength of the interaction between the most frequent interaction usually between the protagonist and a character to the second strongest interaction.

$$\text{HeaviestEdgeRatio}(N) = \frac{w(e_s^1)}{w(e_s^2)}$$

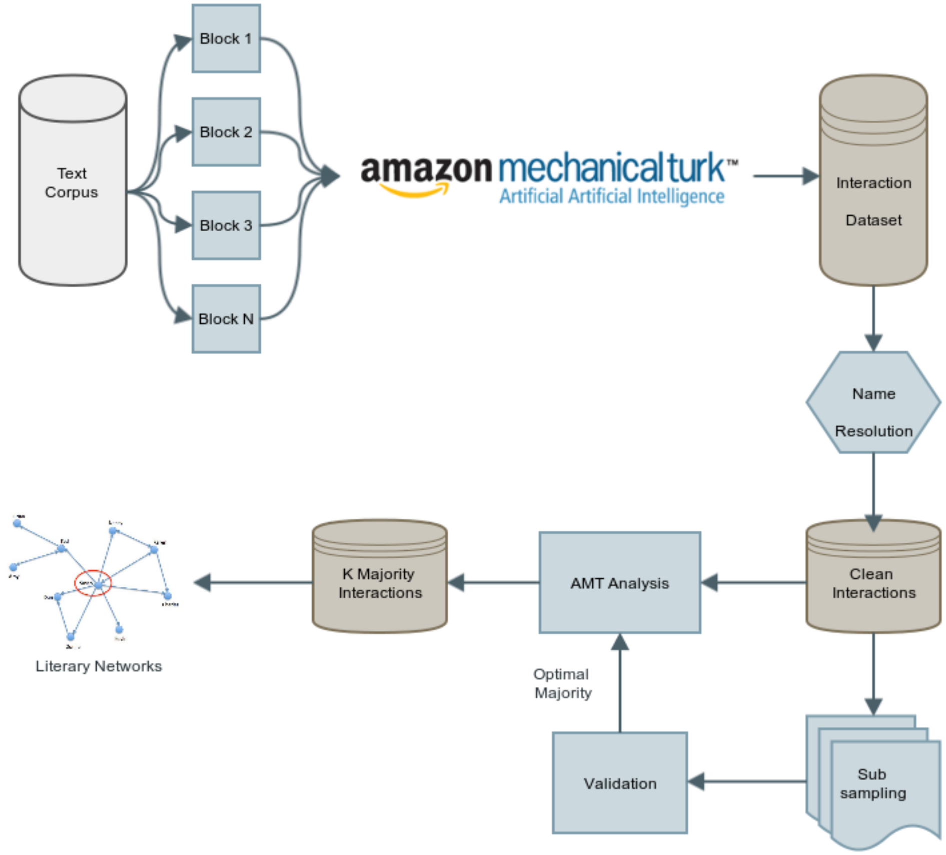


Figure 3–1: Setup of our interaction network pipeline.

Average clustering: It is the average of ratio of number of triangles that are present between any three nodes to the total number of possible. Let us define $A_{uv} = 1$ iff $(u, v) \in E$ else 0. then the average clustering is given by

$$\text{AvgClustering}(N) = \frac{\sum i < j < k \in V A_{ij}A_{jk}A_{ik}}{\sum_{i<j<k\in V} A_{ij}A_{ik}}$$

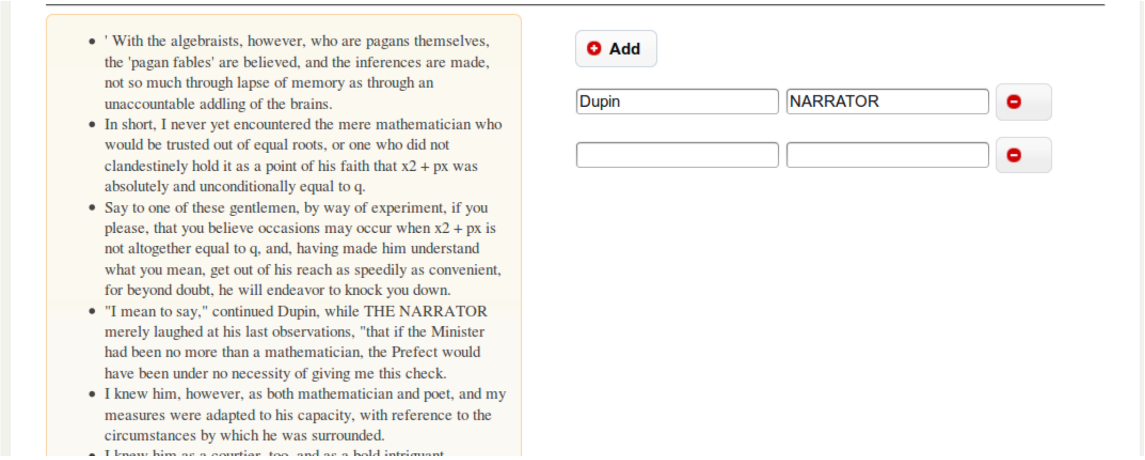
3.3 Mapping Character Networks

Following existing work, we initially conceived of a near fully-automated system for mining character interactions from text. Quickly confronted by the issues also identified in the literature (character name instability, the nuanced typology of interaction, and the encoding of interactions in complex grammar/syntax), we revisited the fundamental objective of the mapping problem: capturing all character interactions as they are encountered in the reading of the text. We realized that, were it possible, human readers would be ideal for mapping such interactions. crowdsourcing presented an opportunity to enlist the aid of human readers in a massively parallel and highly standardized way.

A crucial detail when undertaking the mapping of character interactions involves defining what is meant by an "interaction." Clear definitions of the term are still largely missing in work in this area, although Agarwal, Elson, and Woloch have begun to create useful taxonomies [1, 12, 34]. We formulated our definition of interaction in two different ways: formally and operationally (so as to enable participants in our crowdsourced efforts to understand and act on it).

Formally, we define an interaction as *any concrete action between two characters that requires physical proximity.* This excludes more abstract interactions such as thinking about a person or remembering someone as well as remote interactions mediated by letters or other messages. It also excludes the act of one character simply mentioning another. Our notion of interaction is grounded in a theory of co-presence between characters, which in the philosophical literature would fall under the heading of mutual recognition or theories of "acknowledgement" [19, 14].

Because our interaction mapping method involved the use of a large population of individuals with varying (and generally low) degrees of familiarity with literary analysis, we chose to operationalize our definition through the following instructions provided in the crowdsourcing interface. Participants were told that



- ' With the algebraists, however, who are pagans themselves, the 'pagan fables' are believed, and the inferences are made, not so much through lapse of memory as through an unaccountable addling of the brains.
- In short, I never yet encountered the mere mathematician who would be trusted out of equal roots, or one who did not clandestinely hold it as a point of his faith that x2 + px was absolutely and unconditionally equal to q.
- Say to one of these gentlemen, by way of experiment, if you please, that you believe occasions may occur when x2 + px is not altogether equal to q, and, having made him understand what you mean, get out of his reach as speedily as convenient, for beyond doubt, he will endeavor to knock you down.
- "I mean to say," continued Dupin, while THE NARRATOR merely laughed at his last observations, "that if the Minister had been no more than a mathematician, the Prefect would have been under no necessity of giving me this check.
- I knew him, however, as both mathematician and poet, and my measures were adapted to his capacity, with reference to the circumstances by which he was surrounded.
- I knew him as a courtier, too, and as a bold intriguant

Dupin	NARRATOR

Figure 3–2: An example of the interface presented to Amazon Mechanical Turk workers. The left pane provides the text block to be coded. Interactions are entered into the right pane, one interaction per line. In the present study, interactions are undirected, so the ordering in which names are given for an interaction do not matter.

> "An interaction can be any sort of action between two characters (talking, looking, touching, eating, etc.). An interaction cannot be any other abstract action like thinking (about the character), mention of the character when he is not physically present, etc."

This definition represents an attempt to both capture the essence of our formal definition and also to provide concrete examples that aid comprehension.

3.4 Crowdsourced Interaction Detection Setup

Using the Amazon Mechanical Turk (AMT) platform, we designed a coding interface (see Figure 3–2) in which workers were presented with individual text blocks for which they had to report all interactions between characters that occurred in the text [6].

Figure 3–1 gives the outline for our process. First, the text to be sequenced is broken into fixed block of 250 words each and each block is posted to AMT. Once we get the interaction data from AMT we apply name resolution to resolve all alias and disambiguate the results. We then clean the interactions by removing self-edges and unresolved names. To find the optimal majority, we subsample interactions and verify them manually using an interface similar to Figure 3–2. Finally we build the interaction network using that majority threshold. The following sections give more detailed information on each of these steps.

3.4.1 Text block length

We experimented with text block length and found that 250 words allowed a text block long enough to contain numerous interactions without overwhelming the worker with text. Note that, while the target length for each text block was 250 words, we actually terminated the text block at the end of the sentence after the 250 word mark was reached. This was done in order to preserve semantic meaning at the end of the text block.

It is important to note that this block-based approach created a discrete and regular measure of interaction pacing in stories. Under this model, each work is a sequence of non-overlapping blocks. Furthermore, in a given block, only one interaction can be coded for a distinct pair of characters, effectively setting a resolution limit for the detection of recurring interactions among characters.It is also important to note that this method would miss interactions which occur at the boundaries of blocks ie. one of the character is in one block the other in the next one. Certainly, an interesting question for future work would investigate the utility and value of exploring alternative formalisms, for example overlapping blocks.

3.4.2 Worker instructions.

The AMT coders were given instructions to read a block of text and identify the names of characters who are interacting in the block. As mentioned earlier, we define an interaction as an activity where two or more characters engage physically, e.g. talking, touching, and looking (crucially, the interaction does not need to be symmetric). In addition, we instructed coders to skip pronouns (e.g., "he" or "she") and general references (e.g., "the student") whenever the character to which they referred could not be resolved within the context of the text block.

3.4.3 Coverage and cost.

In order to evaluate the extent to which coding a text block multiple times (called *coverage*) improved the quality and coverage of detected character interactions, we required each text block to be coded by 10 different AMT workers. Initially, we only permitted "Masters" AMT workers to perform the tasks. This, however, proved too stringent a criterion as jobs completed slowly. As a result, we relaxed our criteria to include workers who have at least 95% acceptance rate and at least 1000 accepted tasks. Each task (called a *HIT* in AMT) was assigned 15 minutes for completion and paid the worker $0.10 USD. As a result, a single text block cost one US dollar to code. Notably, as will be discussed later, we found that 10x coverage was not always necessary in order to guarantee high accuracy, so the actual coding cost per text block could be reduced.

3.4.4 Narrator representation.

Early on we discovered that AMT workers struggled to correctly code interactions drawn from stories written with a first-person narrator. In particular, since all ambiguous pronoun references were skipped, AMT workers necessarily left out all interactions involving the narrator. To fix this, in first-person narrator texts, we replaced all "I" references made by the narrator with a special character called "THE NARRATOR." AMT workers were then instructed to treat this special character as any other character in the story. Note that this replacement was done semi-manually (one of the authors and Find/Replace functionality). While this is a clear performance bottleneck in our proposed system, we consider this step to be a natural direction for future work for which NLP tools already exist to approach the problem.

3.5 Interaction Post-Processing

The AMT coding exercise for a given work yielded a set of text blocks with several sets of reported interactions. At this stage, an interaction consisted of the names of two characters, each name being a free text entry field in the AMT interface. In order to construct the canonical character interaction sequence (and the network), then, these names and aspects of the interactions in which they participated needed to be processed such that all names operate as aliases to the correct underlying character.

3.5.1 Name resolution

Name resolution involved solving three different problems: (1) spelling/copying mistakes, (2) unique character aliases (e.g., "Sherlock" and "Mr. Holmes"), (3) ambiguous character aliases (e.g., "my good man" in reference to Dr. Watson at some points in the story and Mr. Holmes in others).

To handle the first two cases, a work's character-alias dictionary (described in Section 3.1) was used and extended. Given the unique alias dictionary, many AMT-provided names could be mapped to canonical character names using the following checks on candidate name x:

1. if x is the canonical character name c, match to c;
2. if x is in the alias list for character c, match to c;
3. if $\text{levenstein}(x, c) > 0.8$ for some canonical character name c, then match to c;
4. if $\text{levenstein}(x, a) > 0.8$ for an alias a belonging to canonical character name c, then match to c;
5. else, x must be an ambiguous alias.

Note that steps 3 and 4 are responsible for correcting spelling and copy errors. Also note that this process of discovering new character aliases is enabled by the use of human readers who bring a level of literary perception to the task which automated methods, to date, are unable to achieve.

All ambiguous aliases (e.g., "the criminal") were resolved through manual inspection: a literature student revisited that text block and determined the character to which the alias corresponded in that context. Like the character alias dictionary construction, this manual process clearly represents a bottleneck in scaling our method. We consider this another important problem for future work in which computation or crowdsourcing could yield good solutions.

3.5.2 Self-interaction removal

On occasion, the name resolution process will reveal that an interaction reported by AMT was actually an interaction between a single character with himself (e.g., Holmes-Holmes). Or in an interaction between three characters, to other characters refer to this character by different names and the AMT coder puts down an interaction between the different alias of the same character which gets resolved to a self-interaction. Such self-interactions reflect situations where the worker was unable to recognize that the two character names resolved to the same character. More generally, in our analysis, we did not include even valid self-interactions (whatever that might mean). As a result, we removed all self-interactions from the AMT interaction data as a final post-processing stage.

3.6 Interaction selection

Given the uncontrolled population of workers who participated in the study, the interpretive nature of the coding exercise, and the non-negligible chance for human error, it would not be prudent to accept all reported interactions as true interactions (in the sense that the interaction actually occurred between two characters in the text). A common strategy for overcoming these crowdsourcing specific issues is to require some proportion of the AMT workers who coded the same text block to report the same interaction in order for it to be accepted as a true interaction [6]. Our 10x coverage allows us to apply different thresholds ranging from 2+ agreement[1] to 10 agreement. The right choice of threshold is, itself, an empirical question which we consider in the next section.

3.7 Network inference.

The threshold-selected interactions provide the basis for directly constructing the character network. Note that the network constructed will be

[1] We use the “2+” notation (as opposed to simply “2”) to indicate that interactions on which 2 *or more* coders agreed should be included.

weighted since we can assign to edge (x, y) the number of interactions reported between characters x and y. Furthermore, because, in this study, we did not enforce a convention in character ordering within an interaction, the network will be undirected. As a direction for future work, it will be interesting to consider how directionality might be captured and encoded in order to reflect the directionality implied in some kinds of interactions (e.g., looking and speaking).

3.7.1 Manual Annotation of AMT Results

In order to use the short fiction corpus described in Section 3.1 for threshold selection and method performance evaluation, we required ground-truth against which to compare interactions discovered by our method. Using a web interface similar to the one used in our crowdsourcing system, a literature student annotated 150 text blocks, selected at random from the set of AMT tasks generated from the entire corpus. For each text block, the student performed two tasks: (1) reporting whether each AMT interaction existed in the text block and (2) reporting all interactions in the text block that were not identified by an AMT interaction. Note that the student was not responsible for correcting spelling mistakes — they were instructed to forgive clear spelling or copy errors.

In order to make this annotated interaction set representative of AMT performance over the entire corpus, the number of text blocks selected from a work was proportional to its length (relative to the rest of works in the corpus). Furthermore, at least one text block was drawn from each work. Hereafter, we refer to this set of 150 text blocks, their annotated AMT interactions and flagged missing interactions as the *annotated dataset*.

3.8 Classifiers

In order to test the effectiveness of the networks statistics proposed in Section 4.4 and to identify characteristic of a genre based on their social network, we build an SVM based classifier using the statistics as features for the classifier. Note that this classifier is purely based on the network statistics and does not account for any language models on which the genres are based on. We use libsvm [2] for implementing our classifier. Training was done using 10-fold cross validation which used an RBF based kernel. The parameters for the kernel were optimially selected by using a grid search over the parameter space.

To compare the performance of the classifier, we implement two language model based classifiers. The first one is a Naive Bayes classifier which depends on the frequency of some words occurring more frequently in a particular genre. For example SF would, on an average, have more words like "murder", "victim" etc when compared to SF.

The second classifier is built using LLDA which models topics in a particular genre. The LLDA assigns each word a weight which reflects probability of that word belonging to a particular topic. We fix the topics as the classes that we want to identify i.e. SF and DF. We use the weight as a proxy for genre probability and classify based on the combined weight of all the words in a text.

[2] `http://www.csie.ntu.edu.tw/∼cjlin/libsvm/`

CHAPTER 4
Results

4.1 Threshold and Coverage Selection

To determine the correct agreement threshold we evaluated how different choices of agreement threshold affected important performance statistics: sensitivity, specificity, precision, and accuracy. These statistics, shown in Table 4–1, were derived from the annotated dataset (described in Section 3.7.1) using the following sets:

- $P = \{(x, y), ...\}$ is the set of interactions, (x, y), between characters that the literature student annotator indicated as existing.
- $N = \binom{C}{2} - P$, where C is all characters mentioned in the text block, is the set of interactions that don't exist in the text block. This formula takes the full set of all interactions that *could* have occurred, given character names present in the text block (i.e., $\binom{C}{2}$), and removes all interactions that *did* occur.
- P^{x+}_{AMT} is the set of interactions in the text block reported by at least x AMT users.
- $N^{x+}_{AMT} = \binom{C}{2} - P^{x+}_{AMT}$ is the set of all possible interactions that could have happened which AMT (with agreement level $x+$) reported as not occurring.

Given these terms, then, we can define true positives as $TP^{x+} = P \cap P^{x+}_{AMT}$, false positives as $FP^{x+} = N \cap P^{x+}_{AMT}$, true negatives as $TN^{x+} = N \cap N^{x+}_{AMT}$, and false negatives as $FN^{x+} = P \cap N^{x+}_{AMT}$. Note that these are the counts for a single block, so the counts used for the entire annotated dataset are summed

Table 4–1: The performance of the AMT-based interaction mapping system when assessed on the annotated dataset. Each row reports the performance of the system under the assumption that a certain number of agreements among independent AMT coders is required to support the inclusion of an interaction. For example, 2+ agreement reports performance statistics if one includes only interactions on which at least 2 AMT coders agreed. As can be seen, the 2+ agreement threshold gives the best overall performance. Also noteworthy is the fact that there were *no* interactions on which 10 workers agreed (thus, the 10 agreement threshold is omitted from all tables).

Agreement	**Spec.**	**Sens.**	**Prec.**	**Acc.**
2+	0.901	0.893	0.922	0.896
3+	0.970	0.653	0.964	0.795
4+	0.985	0.477	0.974	0.711
5+	0.990	0.339	0.976	0.643
6+	0.991	0.230	0.964	0.596
7+	0.991	0.083	0.905	0.524
8+	0.995	0.031	0.875	0.503
9+	1.000	0.009	1.000	0.494

over the 150 blocks. The statistics reported in Table 4–1 were computed using their standard formulations [31].

As can be seen, increasing the required level of agreement increases specificity, the fraction of AMT interactions that are, indeed, interactions in the text. Requiring even moderate levels of agreement, however, have disastrous effects on sensitivity, the fraction of interactions in the text that are identified by AMT: a minimum agreement level of 4 is already missing more than one half of all interactions in the text. Fortunately, the most modest agreement level, 2+, yields very high levels of sensitivity and specificity, making this the clear choice of agreement threshold. For the remainder of the study, this is the threshold we use.

4.2 Optimal coverage

The annotated dataset also afforded the opportunity to consider the best choice of AMT coverage of a text block to achieve best interaction discovery rates. At the outset, it was unclear as to what level of coverage

Table 4–2: Optimal performance of M randomly selected AMT workers. M ranges from 2 to 10. For each M we have K agreement thresholds where K ranges from 2 to M

Agreement	**Spec.**	**Sens.**	**Prec.**	**Acc.**
2 Random workers				
2+	0.992	0.559	0.950	0.903
3 Random workers				
2+	0.972	0.688	0.952	0.845
3+	0.996	0.089	0.917	0.706
4 Random workers				
2+	0.967	0.715	0.962	0.830
3+	0.987	0.441	0.962	0.752
4+	0.996	0.058	0.900	0.625
5 Random workers				
2+	0.967	0.766	0.965	0.858
3+	0.986	0.484	0.968	0.747
4+	0.991	0.192	0.943	0.643
5+	1.000	0.048	1.000	0.600
6 Random workers				
2+	0.922	0.828	0.932	0.869
3+	0.986	0.564	0.978	0.761
4+	0.986	0.341	0.963	0.657
5+	0.996	0.133	0.966	0.585
6+	1.000	0.038	1.000	0.546
7 Random workers				
2+	0.922	0.832	0.933	0.871
3+	0.986	0.568	0.978	0.763
4+	0.986	0.350	0.963	0.661
5+	0.991	0.144	0.939	0.578
6+	0.996	0.048	0.909	0.546
7+	1.000	0.010	1.000	0.533
8 Random workers				

2+	0.922	0.832	0.933	0.871
3+	0.986	0.568	0.978	0.763
4+	0.986	0.350	0.963	0.661
5+	0.991	0.144	0.939	0.578
6+	0.996	0.048	0.909	0.546
7+	1.000	0.010	1.000	0.533
9 Random workers				
2+	0.911	0.877	0.929	0.892
3+	0.975	0.639	0.969	0.790
4+	0.985	0.443	0.972	0.695
5+	0.991	0.314	0.973	0.641
6+	0.991	0.107	0.923	0.544
7+	0.995	0.036	0.889	0.515
8+	1.000	0.009	1.000	0.503
10 Random workers				
2+	0.901	0.893	0.922	0.896
3+	0.970	0.653	0.964	0.795
4+	0.985	0.477	0.974	0.711
5+	0.990	0.339	0.976	0.643
6+	0.991	0.230	0.964	0.596
7+	0.991	0.083	0.905	0.524
8+	0.995	0.031	0.875	0.503
9+	1.000	0.009	1.000	0.494

would optimize the coding result for a given agreement threshold (e.g., 2+). Specifically, given that we conducted the coding with 10x coverage, would less coverage have provided effectively the same result?

There are several factors that make this investigation worthwhile. Most practically, there is the question of cost. Coding the entire corpus at 10x coverage cost $\$1,125$. If, for example, 5x coverage could have given us an equally good result, money could have been saved. There is also reason to suspect that a smaller coverage might result in better performance for a given agreement threshold: as one increases the number of people voting on a solution, the likelihood of X+ people's answers agreeing by chance increases. Thus, for our

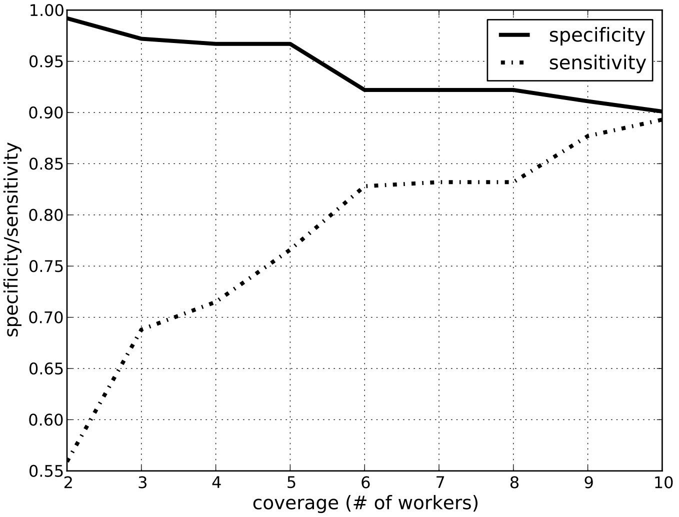


Figure 4–1: The effect of changing the number of workers who code the same text block on the sensitivity and specificity with which interactions are identified in the text. While specificity is relatively stable over all choices of coverage, sensitivity increases dramatically until 6x coverage is reached.

selected 2+ agreement threshold, is there a fold coverage less than 10x that would provide equal or better performance?

To investigate this, we simulated coverage of 2x, 3x, up to 10x by taking only a portion of the completed coding tasks done for text blocks in the annotated dataset (which was done at 10x) such that for each text block we had the desired coverage. We then computed the specificity and sensitivity for this simulated dataset imposing an agreement threshold of 2+.

Figure 4–1 reveals that any coverage value smaller than 6 will have a significant impact on sensitivity. Additionally, this analysis reveals some interesting trends in how coverage differentially affects specificity and sensitivity. Specificity is relatively stable, remaining above 90% and degrading gradually as coverage increases. This indicates that even with few workers per block, the AMT system is quite good at not reporting false interactions (though, admittedly, this is likely due to the relatively large number of negatives present in each block). Sensitivity, on the other hand, dramatically improves by over 35%

as additional coverage is added. Together this suggests that the harder problem for our crowdsourcing platform is identifying true interactions — which is consistent with the challenges of reasoning about the existence and nature of a relationship between the mentions of two characters in a text block.

It is also worth noting that, while the curves are flattening out at 6x coverage, it would be interesting in future work to carry that curve out and determine if or when increasing coverage actually hurts performance. This said, we expect that because our curves meet at 10x coverage that any increase will continue to negatively impact specificity at the expense of sensitivity gains.

4.2.1 Validation

Given that an agreement threshold of 2+ yields the best performance on the annotated dataset, Table 4–1 allows us to determine the performance we should expect from our system: an F1 score of 0.913. We define the $F1$ score of a x_{i+} majority agreement as:

$$F1_{x_{i+}} = 2 \cdot \frac{Precision_{x_{i+}} \cdot Recall_{x_{i+}}}{Precision_{x_{i+}} + Recall_{x_{i+}}}$$

This favorably compares to all existing methods, the best of which have reported F1 scores of 0.609 [2] and 0.67 [12], with the latter method being restricted to only dialogue. It is noteworthy that an F1 score close to 0.67 either means that precision and recall are both around 0.67 or that the performance is imbalanced and one of the statistics is significantly lower, placing it closer to 0.5. In either case, the error rate of such methods will be exceedingly high — either missing true or including false interactions at a rate of between 30% and 40%. As a result, our method stands out as the only currently viable method for mapping interactions both accurately and with a significant degree of scalability.

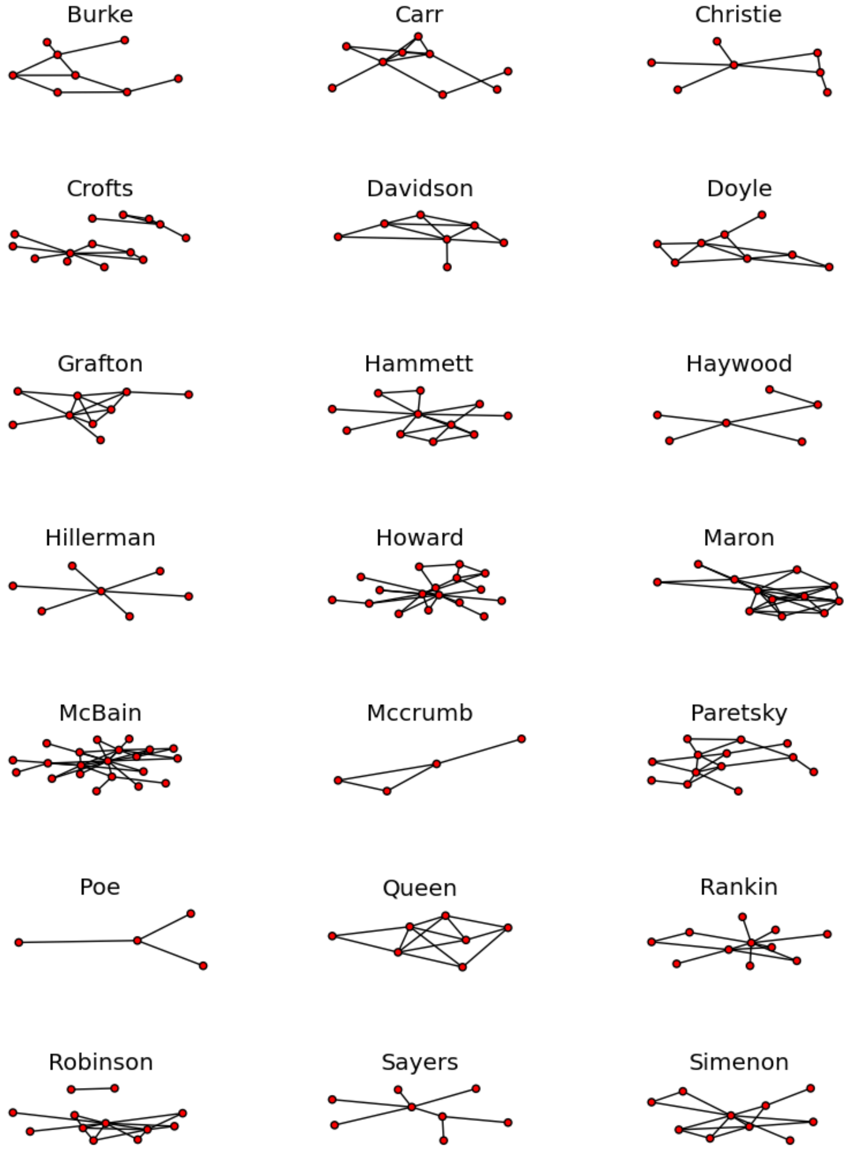


Figure 4–2: Detective fiction networks

4.2.2 AMT-specific Observations

One unexpected and rewarding aspect of our crowdsourcing platform was the enthusiasm it generated among Amazon Mechanical Turk workers. After AMT workers completed the batch of tasks corresponding to the selected corpus, we received no less than 10 emails from AMT workers expressing interest in the work and performing similar tasks in the future (e.g., *"I really enjoyed*

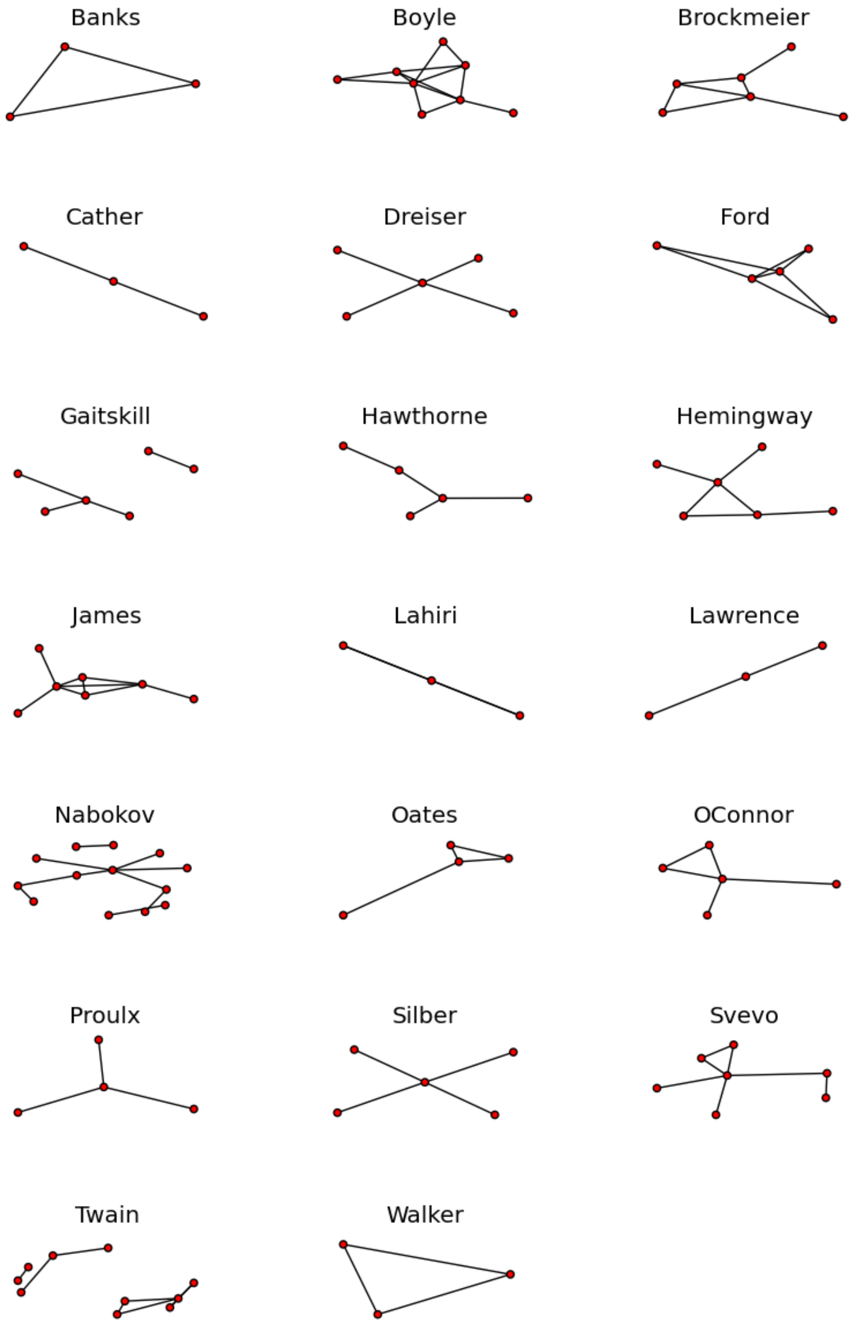


Figure 4–3: Short fiction networks

doing your tasks that you posted last month. I was just curious to know if you will be posting more work in the near future.”). As we will discuss later, this degree of enthusiasm for the coding work suggests that our character interaction mapping system might be successfully converted into a citizen science platform.

4.3 Random Models

Random network models provide another way of investigating the social processes that produce observed networks. To this end, we developed three generative random network models inspired by conventional Erdos-Renyi and preferential attachment processes but which allowed us to simulate the construction of a literary network through the arrival of interactions over the timescale of the work [15]. In all our models, the original interaction timing schedule (when interactions happened in the story) was respected and the number of characters (nodes) was fixed to the original character set size. The models implemented different mechanisms for deciding on the pair of characters that would be involved in a given interaction. In the *uniform* model, the two characters were chosen with uniform random probability for each interaction. In the *uniform-preferential attachment* (UPA) model, one node is chosen with uniform random probability and one node is chosen with probability proportional to their current degree. In the *double preferential attachment* (DPA) model, the two characters were chosen with probability proportional to their current degree in the growing network. We considered these models to express three different plausible mechanisms (random character interaction, important-random character interaction, and interaction based on importance alone) that might drive the growth of social networks in DF and SF.

For each model, we simulated 1000 random networks for each work in our corpus. For each statistic of interest, we computed the p-value of the work's statistic value in its true network against the distribution of statistic values in the networks produced by each model. Table 4–3 shows the fraction of DF/SF works that had a significant (p-value < 0.01) value for each statistic when compared against the three different models. Thus, in this table, a larger value (closer to 1) for a given statistic-model pair indicates that that more works in the genre have a value which significantly deviates from what would be expected under that particular model.

Overall, for both DF and SF, DPA is, by far, the best fit for the statistics considered, evident by comparing the average deviation scores for each model within genre. Between the two genre, SF has statistic values that are better modeled by DPA. This fact becomes more pronounced when one considers the stand-out statistics for DF and SF: the statistics for which that genre has a greater deviation from the DPA model. While DF and SF have an even number of such statistics, the average deviation for DF is much higher than for SF (the average deviation among the more different statistics is 0.66 and 0.44 for DF and SF, respectively). Note that this is not guaranteed to happen due to statistic selection: the criteria that guided the selection of statistics was their ability to distinguish DF and SF networks. The fact that certain of these statistics are more or less similar to DPA networks is independent or, at least, not obviously dependent on this other criterion.

That preferential attachment appears to be a better model for short fiction corroborates our earlier characterization (through feature analysis) of short fiction as a genre marked by social networks that consist of a fewer number of more intense relationships. Moreover, detective fiction's greater deviation from a preferential attachment model suggests that the genre may place increased emphasis on other social network formation processes (such as a preference for reaching distant parts of the network).

4.4 Random network analysis

While the roots of detective fiction extend back into the nineteenth century, it was only at the turn of the twentieth century when detective fiction emerged as one of the quintessential modern genres. For both early and later theorists, narratives of detection were imagined to serve the purpose of sense-making, vehicles for trying to navigate a world that was increasingly less familiar — an experience that would be appreciated by contemporary readers [21, 10]. This could take the form of re-establishing moral codes through the strong binary of good versus evil that pervades detective fiction (Grafton). Or it could take a more conservative tack, as George Grella has argued, where detective fiction is thought to be about restoring a lost social order [16, 18]. Central to these approaches is an emphasis on the singularity of character, the charismatic detective at the heart of the story who allows for this identificatory process of discovery. The detective, whether male or female, private eye or police detective, serves as a stand-in for the uncertain reader [35].

Here we're interested in studying the extent to which detective fiction can be understood less as the experience of a single, charismatic individual (the great detective) and more as an articulation of the social processes through which a shared understanding of the truth comes to be known. In this section we investigate this idea through the hypothesis that detective fiction consists of a more open social network, which dramatizes the navigation of more complex social space in order to arrive at socially accepted truths.

The notion of an open or closed social network is one question (of many) which can be investigated using the character interactions and inferred networks constructed for each text in our corpus. From an operational perspective, by "open" we refer to a network in which the truth seeker (the detective, in this case) favors using interactions to *explore* a social network rather than invest in building a small number of strong relationships. Our hypothesis is that this is a high-level property that appears in detective fiction (DF) and is much less common in general short fiction (SF). As this is a multi-dimensional property, we have used a combination of established and novel network statistics (see Section 3.2) to evaluate this hypothesis.

DF has larger, sparser networks As a starting point, basic node and edge count statistics reveal that DF interaction networks involve more characters and more interactions than short fiction (this is true even when controlling for text length, not shown). However, despite having more edges (and an overall higher average degree), the networks are less dense (the number of edges cannot keep up with the factor of n potential edges created every time a node is added). Consider that a larger, sparser network is a natural pre-condition for having a more open network in which there is space for independent social structures to emerge which the detective must explore.

Certainly, a protagonist who choses to interact with more characters over the course of a story will produce this pattern, supporting our hypothesis. It is also possible, however, that this trend is simply a product of there being more characters in the story: if every character has some interactions (with the main character or otherwise), then this will, itself, drive up the number of edges present. The lower density and greater diameter of DF networks, however, suggests that the greater number of edges is not simply a matter of all characters having more interactions (which would lead to a more fully connected network, driving up the density and decreasing the diameter). Thus, the increased number of edges is not necessarily a product of a larger set of arbitrarily (or uniformly) interacting characters.

DF has less indirectly connected neighborhoods The average clustering coefficient — the connectedness of a node's neighborhood — is not statistically significantly different between DF and SF. Moreover, the amount of clustering is relatively low, suggesting that in both genre neighborhoods are not highly connected. 2-clustering, on the other hand is significantly lower in DF. 2-clustering is the 2-hop (rather than 1-hop) connectedness of a node's neighborhood. Even if the clustering coefficient is low, high dispersion suggests that in a two step random walk, it is relatively common to end up back in the starting node's neighborhood. While both DF and SF show a strong tendency towards high dispersion, DF maintains distinctly lower values, indicating that a detective's walk through the social network will tend to lead to new places rather than back to previously visited neighborhoods.

Notably, the 2-clustering along the strongest edge (which always links to the detective in DF) shows a higher deviation between genre. The higher value for short fiction indicates that the social circles of the central character (by degree) and her most important neighbor are more densely connected (i.e., almost always completely connected). For example, classic short fiction is often driven by very strong single relationships around which the stories revolve, as in the boy narrator and his stand-in father Glen in Richard Ford's "The Communist," or Helen and Harry in Hemingway's "The Snows of Kilamanjaro." This is less true for the detective and the detective's immediate strongest neighbor, implying that the detective more often fills a structural hole in the network, a position well-established to have important information acquisition and control properties [7]. Thus, not only does the detective inhabit a less tightly knit social universe, but within her portion of it, she holds a position that ties it together.

Detectives don't invest in strong relationships A significant part of the open network thesis rests on the intuition that detectives favor spending their interactions to explore new parts of the social network. If this is the case, then we should expect to find that detectives tend to connect to less heavy edges. Two statistics confirm this in different ways. The heaviest edge fraction shows that the heaviest edge (which invariably involves the detective in DF) accounts for a much smaller proportion of all interactions than the heaviest edge in short fiction. The degree-weighted heaviest edge score captures the extent to which the heaviest edge connects the most degree-central nodes in the network: this is more true in SF, indicating that in DF the detective is making different decisions about where to invest her interactions. In practice, sometimes this can have a circular structure, as in Clark Howard's "Under Suspicion," in which the detective Frank Dell moves through a wide array of interactions with suspects and colleagues only to have it revealed at the end that it was his partner who killed the young woman found dead at the opening of the story (who was also the partner's daughter and Dell's former lover).

Detectives aren't the center of the social universe For a detective to be able to explore a social network, there must be a network to explore. This suggests that, in an open network, there should be more going on *outside* of the central character's experiences that matter to the fact-finding process. Two of our statistics directly support this idea. Normalized closeness vitality measures the average increase in shortest path lengths in the network when the most degree-central node is removed. The increase in distance is distinctly less in detective as opposed to general short fiction. Thus, a degree central node in SF is more important in connecting paths of information flow through the network than a degree central node in DF. Said differently, this indicates that detectives (the most degree central node in a DF network) is less central in the network than well-connected nodes in SF. Additionally, if we consider the proportion of the network that the degree central node is connected to, the degree-center neighborhood fraction, we find that this, too, is lower for DF. Certainly, this is likely related to the fact that DF involves more characters; however, regardless, it is notable that the interactions in DF are not allocated to connecting the detective more directly to all parts of the network. These statistics support the idea that the detective — and perhaps the act of detection — is not just about encountering the greatest number of people, but in encountering and exploring, albeit incompletely, more complex social networks. For example, in Margaret Maron's "Deborah's Judgment," at stake in the story is the untangling of complex ties between a series of interconnected and inter-related characters who knew each other from a previous period. Understanding those ties provides the resolution to the crime.

DF takes longer to build/reveal the entire network In contrast to the statistics used above, here we consider the question of how quickly the social network is formed. To investigate this, we formulated the time-to-node/edge/interaction-complete statistics, which capture the percentage of the story required in order to encounter the final node/edge/interaction in the network. We find that the time-to-node-complete and time-to-interaction-complete statistics are not notably different between detective and short fiction. In the case of nodes, it would seem that in DF and SF all characters have been introduced between 50% and 75% of the way through the story. In the case of interactions, the tendency towards late final interactions (i.e. 88% and 80% through the text) likely reveals in both genre the important role that social interactions play in moving a plot forward.

However, the time-to-edge-complete statistic is statistically significant, indicating that DF tends to reveal or add its final edge far later in the story than SF. This suggests that an important part of detective fiction is the late arrival of a relationship which may carry important information for the detective's (and the reader's) truth seeking endeavor. In Georges Simenon's classic "Maigret Deduces," for example, the final relationship established is between the secondary detective and the murderer that generates the confession of the crime, whereas in Diane Davidson's "Cold Turkey" the final edge of the story is between the murderer and the corpse (the victim), constituting an edge that brings resolution for the reader.

In the latter case, the late arrival of a relationship is not only coincident with the proper identification of perpetrator and victim, but more importantly signals the narrative resolution of the two different temporal planes that are, according to a classic essay by Tzvetan Todorov, essential to detective fiction: the narrative present in which the detective operates and the narrative past (of the crime) that must be reconstructed [32].

Table 4–3: A comparison between literary social networks from detective and short fiction and three novel generative network models: uniform random attachment (Uniform), uniform-preferential attachment (UPA), and double preferential attachment (DPA). Values in the table are the fraction of works in that genre that deviate from the random model (for a given statistic). This is computed as the percent of works in a particular genre that, for a given statistic and model, had a statistic value that was statistically different (p-value < 0.01) from those observed in the model networks build for that work. The bold values flag the genre stand-out statistics: statistics for which that genre has greater deviance from the DPA model. Notably, the average deviation of these stand-out statistics is 0.66 for DF versus 0.44 for SF.

	Detective Fiction			**Short Fiction**		
Statistic	*Uniform*	*UPA*	*DPA*	*Uniform*	*UPA*	*DPA*
# of edges	0.9	0.86	**0.62**	0.6	0.65	0.2
average degree	0.9	0.86	**0.62**	0.65	0.65	0.25
degree-weighted heaviest edge score	0.52	0.52	0.52	0.95	0.95	**0.95**
heaviest edge fraction	0.81	0.76	0.48	0.75	0.85	**0.55**
average dispersion	0.43	0.19	0.19	0.3	0.3	**0.3**
max/avg degree ratio	0.86	0.86	**0.76**	0.85	0.65	0.45
density	0.9	0.86	**0.62**	0.6	0.65	0.2
degree-center neighborhood fraction	0.86	0.86	**0.76**	0.85	0.65	0.45
dispersion along the heaviest edge	0.1	0.1	0.1	0.3	0.3	**0.3**
diameter	0.57	0.43	0.24	0.65	0.6	**0.35**
closeness vitality	0.81	0.81	**0.57**	0.75	0.7	0.55
heaviest edges ratio	0.14	0.19	0.1	0.45	0.45	**0.45**
average clustering	0.52	0.43	0.19	0.4	0.35	**0.2**
average deviation	0.64	0.59	0.44	0.62	0.59	0.40

4.4.1 Classifier Performance

With the Naive Bayes classifier we get an average accuracy of 58.4% which is slightly better than random. This suggests that mere word frequency may not be a good indicator for classifying a genre. The LLDA classifier gives the best average accuracy of 92%. The SVM classifier which uses our network features gives an average accuracy of 68.0%. This gives us better performance than Naive Bayes but it is nowhere near the LLDA classifier accuracy.

Though our classifier does not beat the LLDA based classifier. An accuracy of 68% gives a signal that the structure of a network carries withitn itself information which is characteristic of a particular genere. A viable future work would be to develop new statistics around network features which can serve as better features for genere classification.

CHAPTER 5
Discussion & Future Work

This work represents a cross-disciplinary effort to (1) tackle the hard problem of mapping character interaction networks at scale and (2) demonstrate that such computational systems and their quantitative products can provide substantive insight into questions with significant literary merit. The results we have obtained and reported raise a number of larger ideas and questions about this present work and future directions of research.

1. **AMT interaction mapping.** Our method demonstrates the best performance of any method published to date. Additionally, the crowdsourced component enables us to scale mapping efforts to large corpora. This said, the entire mapping pipeline is not yet automated (or crowdsourced), leaving clear and important directions for future work that aim to improve the overall scale of datasets that can be processed. Three problems stand out. First, techniques must be developed for normalizing narrator representation in texts in order to make first-person narrators easy to code. Second, we require techniques for constructing character-alias dictionaries that do not require a single (or handful of) expert(s) to sit and hand-code name-character relationships. Third, post-hoc name resolution after the crowdsourced interaction sets have been returned needs to be automated.

All three of these are interpretive exercises. One exciting research direction, then, would focus on hybrid solutions in which automated, crowdsourcing, or expert-based systems coordinate to perform the tasks mentioned above.

2. **Alias discovery.** Where the construction of character-alias dictionaries is concerned, our method presents some valuable improvements over automated systems. Unlike automated systems which require such a dictionary and then process only names that match entries in the dictionary provided, humans code interactions through a more natural and sophisticated reading of the text which can yield names and aliases that were not included (for whatever reason) in the character-alias dictionary. While this necessitates the name resolution phase, our crowd-based system has the ability to correct errors and omissions in the dictionary, which overcomes a potentially serious source of error in character mapping studies.

3. **Citizen science.** Given the remarkable accuracy we obtain through crowdsourced coding of interactions and the enthusiastic responses from the AMT workers who did the work, we believe that our method is well suited to be rolled out as a citizen science initiative. The successful launch of such a platform would eliminate much of the cost associated with mining books, would provide a dedicated population of interaction coders, and could attract the attention of individuals more familiar with literature and literary analysis (providing a larger population of experts

who could be tasked with the harder problems such as building and curating character-alias dictionaries).

4. **Devising new network statistics.** There has been a tendency in past work to apply only existing network statistics in pursuit of a literary hypothesis. However, attacking the complex questions and ideas that arise in literary analysis requires a willingness to develop and use new network statistics. Our analysis of detective and short fiction is an example of this: the standard network statistics did not completely address the core question of interest, leading us to develop additional measures to support our thesis.

5. **Literary insights.** This analysis has generated valuable literary insights at multiple levels. First, we have been able to show that social networks are a good indicator of genre. Until now, this correlation between form and social dynamics in literature had been assumed but not proven.

   Second, the use of social network analysis has led to distinctly new insights into the potential meanings and social functions of our selected genre of detective fiction. Our statistics indicate that far from being exclusively about issues of play, morality, charisma, or even suspense, one of detective fiction's indicative features is its ability to dramatize the navigation of more complex and open social networks. We find that detective fiction can be understood as a genre designed to generate new kinds of social order, ones that consist of what we would call extensive versus intensive social networks [16]. The function of detective fiction as a genre is to facilitate the imaginary navigation of an increasing social openness and simultaneously generate a sense of consensus within such openness. It remains to be seen whether these results hold for different languages and cultures (does German detective fiction or the more recent rise of Mexican border detective fiction exhibit similar features?) as well as different genres. Whether social networks are indicative of different sub-genres of the novel, for example, or between novels and other long prose works such as the epic or romance marks an important area of new research.

6. **Random models for character interactions.** Detective fiction's lack of fit to the models considered suggests that different social processes are at work that are not accounted for by these random models and indicate an interesting area for further exploration. More broadly, our findings (and the general success of random models in network science literature) confirm that random models can not only tell us how constructed or "intentional" a literary social network is, but they can also give us insights into the social processes that are unique to different literary genres. We consider the application and innovation of random network models in support of literary analysis to be an exciting direction for the field.

CHAPTER 6
Conclusion

In this thesis we present a novel crowdsourcing based method for extracting interaction networks from literary texts at large scale. We use detective fiction and short fiction texts in our study and show that we can reliably and accurately reconstruct the social network of characters. We also propose several statistics that we show are significant for different generes.

We compare our networks with randomly generated networks and find that the extracted networks have features which significantly differ from random networks showing that these networks are man-made. We also justify the literary meaning behind the stats and why they are significant. Finally we build an SVM based classifier from our stats and get 68% accuracy.

Our methods are generic and scalable enough to be extended to other generes & larger corpora of texts like novels. They can also be used in cases where we need to discover an interaction between two entities not limited to text. Because of the enthusiasm and positive response that we got from AMT users, We believe that this work can be converted to a citizen science project where people will participate voluntarily and help in creating more networks.

REFERENCES

[1] A. Agarwal, A. Corvalan, J. Jensen, and O. Rambow. Social network analysis of alice in wonderland. In *Proceedings of the NAACLHLT 2012 Workshop on Computational Linguistics for Literature*, pages 88–96, 2012.

[2] A. Agarwal, A. Kotalwar, and O. Rambow. Automatic extraction of social networks from literary text: A case study on alice in wonderland. In *the Proceedings of the 6th International Joint Conference on Natural Language Processing (IJCNLP 2013)*, 2013.

[3] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[5] D. A. Brewer. *The Afterlife of Character, 1726-1825*. University of Pennsylvania Press, 2011.

[6] M. Buhrmester, T. Kwang, and S. D. Gosling. Amazon's mechanical turk: A new source of inexpensive, yet high-quality data? *Perspectives on Psychological Science*, 6(1), 2011.

[7] R. Burt. Structural holes and good ideas. *American Journal of Sociology*, 110(2):349–399, 2004.

[8] G. Casella and E. I. George. Explaining the gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.

[9] V. Cevher, D. Kahle, K. Tsianos, and T. Saleem. Variational bayes approximation. *Rice University*, 2008.

[10] G. Chesterton. A defence of detective stories. 1901.

[11] W. M. Darling. A theoretical and practical implementation tutorial on topic modeling and gibbs sampling. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 642–647, 2011.

[12] 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*, pages 138–147. Association for Computational Linguistics, 2010.

[13] P. Erdős and A. Rényi. On the evolution of random graphs. *Magyar Tud. Akad. Mat. Kutató Int. Közl*, 5:17–61, 1960.

[14] N. Fraser and A. Honneth. *Redistribution or recognition?: a political-philosophical exchange.* Verso, 2003.

[15] A. Goldenberg, A. X. Zheng, S. E. Fienberg, and E. M. Airoldi. A survey of statistical network models. *Foundations and Trends in Machine Learning*, 2(2):129–233, 2010.

[16] G. Grella. The formal detective novel. *Detective Fiction: A Collection of Critical Essays*, pages 84–102, 1980.

[17] M. A. Hearst, S. Dumais, E. Osman, J. Platt, and B. Scholkopf. Support vector machines. *Intelligent Systems and their Applications, IEEE*, 13(4):18–28, 1998.

[18] M. Holquist. Whodunit and other questions: metaphysical detective stories in post-war fiction. *New Literary History*, 3(1):135–156, 1971.

[19] A. Honneth and A. Margalit. Recognition. *Proceedings of the Aristotelian Society, Supplementary Volumes*, pages 111–139, 2001.

[20] J. Howe. *Crowdsourcing: How the power of the crowd is driving the future of business.* Random House, 2008.

[21] P. Hühn. The detective as reader: Narrativity and reading concepts in detective fiction. *MFS Modern Fiction Studies*, 3(3):451–466, 1987.

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

[23] D. Mansfield-Kelley and L. A. Marchino. *The Longman Anthology of Detective Fiction.* Longman, 2005.

[24] V. Marx. Neuroscience waves to the crowd. *Nature methods*, 10(11):1069–1074, 2013.

[25] T. P. Minka. Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc., 2001.

[26] F. Moretti. Network theory, plot analysis. *New Left Review*, 2011.

[27] J. Phelan. *Reading people, reading plots: Character, progression, and the interpretation of narrative.* University of Chicago Press, 1989.

[28] V. Propp. *Morphology of the Folktale.* University of Texas, 1968.

[29] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 248–256. Association for Computational Linguistics, 2009.

[30] I. Rish. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

[31] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall, 2009.

[32] T. Todorov. The typology of detective fiction. *The Poetics of Prose*, pages 42–52, 1977.

[33] B. Vermeule. *Why do we care about literary characters?* JHU Press, 2011.

[34] A. Woloch. *The One vs. the Many: Minor Characters and the Space of the Protagonist in the Novel.* Princeton University Press, 2009.

[35] L. Zunshine. *Why we read fiction: Theory of mind and the novel.* Ohio State University Press, 2006.