Linguistic **I**ssues in **L**anguage **T**echnology – **LiLT**
Submitted, October 2015

Clustering of Novels Represented as Social Networks

Mariona Coll Ardanuy
Caroline Sporleder

Published by CSLI Publications

LiLT volume 12, issue 4 October 2015

Clustering of Novels Represented as Social Networks

Mariona Coll Ardanuy, *Trier University, mcollardanuy@gmail.com*
Caroline Sporleder, *Trier University,sporledc@uni-trier.de*

Abstract

Within the field of literary analysis, there are few branches as confusing as that of genre theory. Literary criticism has failed so far to reach a consensus on what makes a genre a genre. In this paper, we examine the degree to which the character structure of a novel is indicative of the genre it belongs to. With the premise that novels are societies in miniature, we build static and dynamic social networks of characters as a strategy to represent the narrative structure of novels in a quantifiable manner. For each of the novels, we compute a vector of literary-motivated features extracted from their network representation. We perform clustering on the vectors and analyze the resulting clusters in terms of genre and authorship.

LiLT Volume 12, Issue 4, October 2015.
Clustering of Novels Represented as Social Networks.

> The study of the novel as a genre is distinguished by peculiar difficulties. This is due to the unique nature of the object itself: the novel is the sole genre that continues to develop, that is as yet uncompleted. [...] The generic skeleton of the novel is still far from having hardened, and we cannot foresee all its plastic possibilities.
>
> Mikhail Bakhtin, *Epic and Novel: Toward a Methodology for the Study of the Novel*, 1941

1 Introduction

Within the field of literary analysis, few branches are as confusing as that of genre theory. This confusion arises mostly from the lack of consensus when it comes to defining what a literary genre is. The *Oxford Dictionary of Literary Terms* (ODLT) (Baldick, 2008) describes the literary genre as "a recognizable and established category of written work employing such common conventions as will prevent readers or audiences from mistaking it for another kind". This vague definition gives a sense of the difficulty that lies behind the task of categorizing literary works according to their genre. Labels such as 'prose', 'novel', 'tragedy', and 'detective novel' are used to indicate the genre of a literary work, even though they do not refer to the same level of classification. In this paper, we focus on the study of novelistic subgenres. Henry James once referred to novels as "large loose baggy monster[s]",[1] due to their long, unwieldy and unconfined nature. They are complex objects that do not lend themselves to easy classification. Many different taxonomies of novelistic subgenres exist. Thus, assigning a subgenre to a novel, very often subjective, is the fruit of intuition rather than the product of detailed and systematic analyses. This subjectivity is, to nobody's surprise, a source of disagreement.

Until quite recently, most computational studies of literature had focused mainly on form and content. Narrative structure, considered a key dimension of the novel, had largely been ignored – mostly because of the complexity of representing it in a quantitative manner. In this study, we explore the degree to which the character structure of a novel can be used to represent its genre. We also examine how this approximation is representative of the genre of a novel and, in less depth, how characteristic the depicted community is of the style of its author. We represent novels as static and dynamic social networks of characters. From such networks, we extract graph-based literary-motivated features that can be used to cluster documents according to the structural similarity of the novels.

[1] James (1908) referred particularly to long nineteenth-century novels.

2 Background

2.1 Genre theory and the theory of the novel

Literary genre studies began as an effort to organize the space of literary production, by grouping together literary texts with similar characteristics (Madsen, 1994). One of the first attempts to create a literary taxonomy is *Poetics* by Aristotle (2007). However influential, it should not be used to organize the space of modern literary production, because genres have mutated and evolved considerably since it appeared in ca. 335 BCE. In our study, we focus on the modern novel, a relatively young literary form which began developing almost two millennia after Aristotle wrote his *Poetics* treatise,[2] and is considered the most important literary genre of the modern age. There have been many, thus far unsuccessful, attempts to find a conclusive – or at least, widely-accepted – categorization of the spectrum of novelistic subgenres. The novel stands out among other literary forms due to its unconstrained style and structure. Because of this, but also because of the lack of consensus on the classification criteria, new genres proliferate, largely in an attempt not to leave any novel uncategorized. The ODLT, for example, states that different subgenres arise from differences in characters, settings, plots or structures; Spang (1993) considers that they develop around form, content or both; and according to Fowler (1982) they are categorized by subject matter or motifs, substance, configuration, and the influence of neighboring genres.

The Russian literary critic Mikhail Bakhtin has been one of the most influential voices in the study of the novel. His main contribution was probably the concept of the *chronotope*, a literary device "that defines genre and generic distinctions" (Bakhtin, 1981b). The term *chronotope* – from Greek *χρόνος* 'time' and *τόπος* 'place' – refers to the "intrinsic connectedness of temporal and spatial relationships that are artistically expressed in literature" (Bakhtin, *ibid.*); it is, in other words, a narrative time-space unit. It is this amalgamation of space and time, in which the representation of the plot and the structure of characters lies, that Bakhtin considers to be at the heart of the distinction of different novel subgenres. Novels of the same subgenre share the same chronotopes, which account for the differences in representation of plot and characters. Abbott (2008) introduces the notion of 'masterplot' as the recurrent skeletal story of a genre that occurs between the axes of time and space, corresponding to Bakhtin's chronotope. As Emer-

[2] According to the acclaimed literary critic Harold Bloom, the first modern novel would be *Don Quixote* by Miguel de Cervantes, published in 1605 (Bloom, 2003), even though earlier claimants exist.

son (1986) indicated, the representation of the characters in a novel varies according to its chronotope(s) and, thus, according to its subgenre. The structuralist theorist Vladimir Propp analyzed the structure of 100 Russian fairy tales in his most famous study, *Morphology of the Folktale* (Propp, 1968). In another of his studies, *Theory and History of Folklore* (Propp, 1984, p. 41), he discusses the problem of defining genre. Focusing on the classification of folklore genres, he considered characters to be key elements for the identification of a genre. According to Propp, each genre has a different structure closely related to the plot, and since plot is realized by characters, in some instances genre classification should be possible in terms of characters.

2.2 Unsupervised document classification

Unsupervised document classification consists in automatically grouping a set of documents by the similarities among them. Unlike its supervised counterpart, it requires neither labeled training data nor prior knowledge of the classes into which the texts are to be categorized. Instead, similar documents – represented as vectors of features – are grouped together to yield a clustering that depends on the features chosen to characterize the document. Without supervision, there is no guarantee that the resulting clusters correspond to the classes in which we are interested (Zhang, 2013).

As described by Andrews and Fox (2007), content is traditionally the grouping criterion in document clustering. This may be achieved by representing the content of the document as a bag of words (BoW) (Willett, 1988, Steinbach et al., 2000). More sophisticated approaches have also explored content-based document clustering using such different techniques as Nonnegative Matrix Factorization, Vector Space Model, Latent Semantic Analysis, Self-Organizing Map or Locality-Preserving Projection (Shahnaz et al., 2006, Basili et al., 2008, Wang et al., 2011, Margonari, 2011).

In the literature domain, document clustering has mostly been applied to unsupervised authorship analysis.That is the task of automatically grouping texts according to their author, by determining the set of features that distinguish one author from another. The early studies employ mainly stylometric features such as punctuation and function word frequencies (Ledger and Merriam, 1994, Holmes and Forsyth, 1995, Baayen et al., 1996, Aaronson, 2001). More recent works use content-based features, such as BoW representations (Akiva and Koppel, 2012, Layton et al., 2011). Pavlyshenko (2012) explicitly brings document clustering by author to the literature domain. Semantic fields are proposed as a way of capturing the author's idiolect and serving as

the basis for the vector space.

Much less effort has been devoted to the task of clustering documents according to their genre; see Gupta et al. (2005), Poudat and Cleuziou (2003), Bekkerman et al. (2007) for the examples of generic document clustering by genre. The work of Allison et al. (2011) was one of the first (and, to date, one of the few) to address the problem of clustering by genre in the literary domain. Their study proposes the use of stylometric features, mainly based on frequency counts, to recognize the different literary genres. Even though some strong genre clustering could be observed for certain genres, the authors realized that the classification was not only obeying genre criteria. The stylistic signature of every document was carrying a strong 'author' signal, which would sometimes conceal the 'genre' signal. An excerpt from the paper exemplifies it thus: "when, say, Dickens moves from the industrial novel *Hard Times* to the urban multiplot of *Little Dorrit*, the historical *Tale of Two Cities*, or the *Bildungsroman* of *Great Expectations* – what happens is that his plots change, but his style doesn't".

Continuing the work of Allison et al. (*ibid.*), Jockers (2013, chapter 6, "Style") presents a deeper investigation of how the genre signal is affected by the presence of alternative factors such as author, time period, nationality, or author gender. The deep analysis presented in this chapter supports the conclusion that some genres are more formulaic than others, and ask for stronger stylistic patterns.

2.3 Quantitative literary analysis

Section 2.2 lists work on document clustering that uses either stylometric or bag-of-words-based features. Novels, however, should not be reduced to punctuation, morphology, syntax, and bag-of-word representations. This literary form has a depth, a complex structure of plot and characters. The Russian structuralist school considers the plot of a novel to be modeled by its collection of characters and the actions they carry out (Bakhtin, 1981a, Propp, 1968). Moretti (2011), concerned with plot quantification, explores extensively the effect that characters have on the plot. He creates a social network of William Shakespeare's *Hamlet*, in which the characters are the nodes. Several experiments (the removal of the protagonist, isolated nodes or a connecting character from the network) allow him to show how the plot changes according to the alteration in the structure of characters. Sack (2012) proposes social networks of characters as a mechanism for generating plots artificially. Alberich et al. (2002) made one of the first attempts to combine social networks and literature. They built a social network from the Marvel comics, in which characters are the nodes linked by their co-occurrence

in the same book. The authors note that the resulting network is very similar to a real social network. Newman and Girvan (2003) used a hand-built social network with the main characters of Victor Hugo's *Les Misérables* to detect communities of characters, densely connected, that reproduced the subplot structure of the novel.

Elson et al. (2010) introduced a new approach to create networks from novels: characters are linked if they converse, instead of being linked if they occur in the same window of text. The networks are built automatically, and heuristics are used to generate variations of the names of the characters and to cluster the coreferent names. The analysis of the networks is then used to refute some literary hypotheses. Jayannavar et al. (2015), revisiting the work of Elson et al., revised the set of hypotheses, which they validated using networks based on social events (observations and interactions). Celikyilmaz et al. (2010) extracted dialogue interactions in order to analyze semantic orientation of social networks from literature. In order to perform large-scale analyses of the works, both Rydberg-Cox (2011) and Suen et al. (2013) extract networks from structured text: Greek tragedies the former, plays and movie scripts the latter.

All the approaches mentioned above produce static networks which are flat representations of the novel. Past, present and future are displayed simultaneously in such networks: time turns into space. The recent work by Elsner (2012) and Agarwal et al. (2012) questions the validity of static network analysis. Agarwal et al. introduce the concept of dynamic network analysis for literature, motivated by the idea that static networks can distort the characters' importance (exemplified by a case analysis of Lewis Carroll's *Alice in Wonderland*). A dynamic social network is but the collection of independent networks for each of the parts into which the novel is divided.

2.4 Contributions

Our paper makes several contributions. It is one of the first attempts to solve in a quantitative fashion the challenging task of clustering novels according to genre, and the first one that does so by means of its structure of characters. The same method is then used to assess how representative the network structure of a novel is of the style of its writer, which gives us some insight into the manner in which authors imagined a community. The creation of the social network itself essentially builds on (Elson et al., 2010), even though not completely. Our person-name coreference resolution module, for example, does not generate possible variations for each name, but takes as a basis all the names in the novel and clusters them together, exploiting the advan-

tages of novels as confined objects. In order to perform the clustering, we use literary-motivated features from the static and dynamic social networks created for each novel, which we use to cluster the novels according to the genre they belong to and according to their author.

3 Turning novels into social networks

A social network is a structure that captures the relations among a set of actors. In a novel, the actors are its characters. In order to obtain the list of characters of a book, we extract the list of all person names (section 3.1) and perform coreference resolution[3] to identify the characters to which those names refer (section 3.2). Once we have a list of characters, we link them according to our definition of interaction and construct the network (section 3.3).

3.1 Extraction of person names

In order to extract the list of person names from a novel, we use the `Stanford Named Entity Recognizer` (*Stanford NER*).[4] In the absence of training data from the literary domain, we used the model trained on news that comes by default with the software. The performance of Stanford NER is very high when in-domain,[5] but it decreases when applied to literature (see Table 1).

Novels have certain characteristics that facilitate the recognition of person entities. We propose the following post-processing steps in order to enhance the performance of the recognizer.

- **Recognition patterns typical from the literary domain**: a list of 178 honorifics and titles (such as 'Sir', 'Lady', or 'Professor') and a list of 83 verbs of utterance (such as 'say', 'complain', and 'discuss', both in the present and the past) indicating the immediate presence of a person.
- **Re-tagging of the file**: Stanford NER tags entities sequentially and thus inconsistencies may occur (e.g. whereas 'Leicester' is at first identified as a person, it is recognized as a location just three sentences below, as shown in Figure 1). Given that a novel represents a small universe in which it is rarely the author's intention to confuse the reader, we assume that a proper name will always refer to the same entity. If a name has been tagged inconsistently, and if the previous filtering step has recognized it as a person somewhere in the

[3] This is not full coreference resolution: we do not resolve definite noun phrases and pronouns, only link person names.

[4] http://nlp.stanford.edu/software/CRF-NER.shtml

[5] Finkel et al. (2005) note the F_1 score of 92.29 for *person* entities on the CoNLL 2003 named entity recognition dataset.

text, we add it to the list of person names unequivocally extracted by Stanford NER; otherwise, we add it to this list only if, throughout the file, it has been tagged as a person in most of the cases. This list of person names is used to re-tag the file, making sure that there are no more inconsistencies.

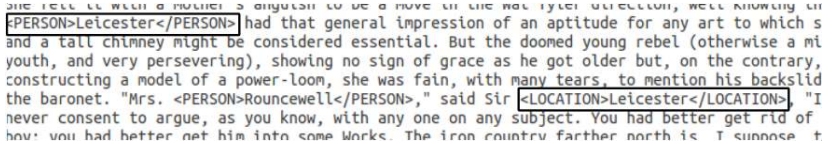

FIGURE 1 Snapshot of the Stanford NER tagging of an excerpt from Charles Dickens' *Bleak House*, before applying any literary-specific filtering.

	Precision	*Recall*	F_1 *Score*
StanfordNER-Eng	0.9684	0.8101	0.8822
FilteredNER-Eng	0.9816	0.9970	0.9892
StanfordNER-Trn	0.9287	0.7587	0.8351
FilteredNER-Trn	0.8589	0.8277	0.8430

TABLE 1 Evaluation of person recognition.

Table 1 shows the evaluation of the person name recognizer in novels originally written in English and in translated novels, both before (`StanfordNER`) and after (`FilteredNER`) the post-processing step. The performance is improved notably in the case of English literature, and only slightly in literature originally written in languages other than English. We evaluated eight chapters randomly selected from eight different novels.[6] In this task, precision is the proportion of person names (tokens) identified by the system that are correct, and recall is the proportion of relevant person names (tokens) that are retrieved.

3.2 Person name coreference resolution

The list of person names which appear in a novel is by no means a list of characters. The names 'Miss Lizzy', 'Miss Elizabeth', 'Miss Elisabeth Bennet', 'Lizzy', 'Miss Eliza Bennet', 'Elizabeth Bennet', and 'Elizabeth' in Jane Austen's novel *Pride and Prejudice* all refer to the same

[6] *Little Dorrit* and *The Pickwick Papers* by Charles Dickens, *Pride and Prejudice* by Jane Austen, *Dr. Jekyll and Mr. Hyde* by R. L. Stevenson, *The Hunchback of Notre-Dame* by Victor Hugo, *The Phantom of the Opera* by Gaston Leroux, *War and Peace* by Leo Tolstoy, and *Don Quixote of La Mancha* by Miguel de Cervantes. In total, 27,392 tokens were annotated, of which 827 were person names.

entity, its protagonist. In order to create a social network of the novel, we need to link the characters, and thus it is crucial first to group all coreferents together.

Parsing person names

We used an extended version of the `python-nameparser`[7] software to parse the recognized names into their different components, namely *title*, *first name*, *middle name*, *last name*, and *suffix*. For example, a name like 'Detective Sherlock Holmes' is parsed into the *title* 'Detective', the *first name* 'Sherlock', and the *last name* 'Holmes'. This is a rule-based approach which relies on lists of titles and suffixes and on some strong assumptions, for instance that every two-token name consists of a first name and a last name unless one of the tokens is a title or a suffix. A common error arises for single-token person names, such as 'Holmes' or 'Sikes', which are always considered as first names. This does not affect our coreference resolution module, since our algorithm does not differentiate between the first and last name in single-token names.

Assigning gender to names

Each name is assigned a gender (*male*, *female*, or *unknown*). We use four lists: typical male titles ('Sir', 'Lord', etc.), typical female titles ('Miss', 'Lady', etc.), 2579 uniquely male first names,[8] and 4636 uniquely female first names.[9] In order to assign a gender to a person name, we first consider the title. If the title is empty or not informative of the gender (such as 'detective', which applies to males and to females), the first name is considered. If none are informative, the immediate context is considered: a counter keeps track of the count of 'his' and 'himself' (on the one hand), and of 'her' and 'herself' (on the other) appearing in a window of at most three words to the right of the name. Depending on which of the two counters is higher, the person name is assigned one gender or the other. If the conditions are not met, the assigned gender is *unknown*. We evaluate the gender assignment module on the total number of person names from three novels originally written in English and three novels translated into English[10] (see Table 2).

[7] http://code.google.com/p/python-nameparser/

[8] http://www.cs.cmu.edu/Groups/AI/areas/nlp/corpora/names/male.txt

[9] http://www.cs.cmu.edu/Groups/AI/areas/nlp/corpora/names/female.txt

[10] *Oliver Twist* by Charles Dickens, *Sense and Sensibility* by Jane Austen, *The Hound of the Baskervilles* by Arthur Conan Doyle, *Around the World in Eighty Days* by Jules Verne, *The Phantom of the Opera* by Gaston Leroux, *On the Eve* by Ivan Turgenev. There were 328 person names in English novels, and 324 in foreign novels.

	Precision	*Recall*	F_1 *score*
EnglishLit	0.9725	0.8676	0.9171
ForeignLit	0.9603	0.5734	0.7180

TABLE 2 Evaluation of gender assignment.

Matching names

A five-step matching algorithm is responsible for grouping the different coreferents from the least ambiguous to the most ambiguous:

1. Select names with **title**, **first name**, and **last name** (e.g. 'Miss Elizabeth Bennet') and add them to an empty list of nodes.
2. Select names with **first name** and **last name** (e.g. 'Elizabeth Bennet'). If the name partially matches an existing node, they are grouped together. Otherwise, the name is added as a new node to the list of nodes.
3. Select names with **title** and **first name** (e.g. 'Miss Elizabeth') and apply the procedure from step 2.
4. Select names with **title** and **last name** (e.g. 'Miss Bennet') andand apply the procedure from step 2.
5. Select names with only **first name** or **last name** (e.g. 'Elizabeth' or 'Bennet') and apply the procedure from step 2.

At each step, we take into account the following considerations:

- A first name can appear as a nickname ('Lizzy' is 'Elizabeth').[11]
- A first name can appear as an initial ('J. Jarndyce' is 'John Jarndyce').
- Gender of the names must agree ('Miss Sedley' matches 'Amalia Sedley', but not 'Jos Sedley').

If a referent is still ambiguous after these steps, we group it together with its most common match. 'Mr. Holmes', for example, might refer to both 'Sherlock Holmes' and his brother, the minor character 'Mycroft Holmes'. According to our algorithm, 'Mr. Holmes' matches both entities. In these cases, we assume that the ambiguous name refers to the more frequent of the two characters, in this case 'Sherlock Holmes'.

Assessing the performance of our coreference resolution method is not an easy task, since different characters have different relevance in the novel, so that the effect of a misidentification correlates with the relevance of the incorrectly identified character. The evaluation that we propose for this task takes into consideration only the 10 most men-

[11] We use a list of hypocoristics at https://metacpan.org/source/BRIANL/Lingua-EN-Nickname-1.14/nicknames.txt.

tioned characters from 10 different novels.[12] Here, precision is the proportion of predicted co-referents of the character that are correct, and recall is the proportion of correct co-referents out of all the co-referents that should have been predicted.

It is important to note that for this evaluation we only considered correctly recognized person names, because we were interested specifically in the performance of the matching algorithm. Table 3 shows the results of the evaluation.

	Precision	*Recall*	$F_1 Score$
English Literature	0.9866	0.9371	0.9612
Foreign Literature	0.9852	0.9086	0.9454

TABLE 3 Evaluation of coreference resolution.

The observed errors mostly fall into two categories: (i) a name can refer to more than one character, as the aforementioned Holmes brothers; and (ii) the name has no surface indication of the character to which it refers, as is the case of *Oliver Twist*'s character Jack Dawkins, mostly referred to as the Artful Dodger. In the first case, in an attempt to minimize the error, our system considers that, whenever a name can refer to two or more characters, we can expect that most of the times it refers to the most important of them. This assumption certainly minimizes error in the case of the Holmes brothers, but is more problematic in Jane Austen's *Sense and Sensibility*, where the two Dashwood sisters are the coprotagonists of the novel. At the moment, we do not offer a solution for this.

Finally, we can see that, even though the task was originally conceived for evaluation only on literature originally written in English, the difference in the F_1 score between English literature and foreign literature is small. As already stated, we have performed coreference resolution evaluation over the correct results from the person name recognizer, and therefore the overall performance of character recognition is lower in the case of foreign literature.

[12] Two novels have been used for development: *Pride and Prejudice* by Jane Austen, and *Bleak House* by Charles Dickens. The 10 novels that have been evaluated are: *The Mystery of Edwin Drood* by Charles Dickens, *The Hound of the Baskervilles* by Arthur Conan Doyle, *Vanity Fair* by William M. Thackeray, *Oliver Twist* by Charles Dickens, *Sense and Sensibility* by Jane Austen, *Around the World in Eighty Days* by Jules Verne, *The Phantom of the Opera* by Gaston Leroux, *Les Misérables* by Victor Hugo, *The Three Musketeers* by Alexandre Dumas and *Madame Bovary* by Gustave Flaubert. The total number of person names used for evaluation is 1630 for English literature and 2269 for foreign literature, most of which false negatives since we only considered the ten most frequent characters.

3.3 Construction of the social network

As noted in section 2, there are two main approaches to creating character networks from literary fiction. In the first approach (hereafter called **conversational network**), an edge connects two characters edge whenever there is an explicit spoken interaction between them. In the second approach (hereafter called **co-occurrence network**), an edge connects two characters whenever they co-occur in the same window of text, be it a sentence, a paragraph, or a longer stretch of text.

A conversational network is well-suited to highly structured literary forms, such as plays, in which each social interaction is represented by scripted dialogue between characters. In novels, much of the interaction takes place off-dialogue through a description by the narrator, and thus a conversational network might not suffice to capture it completely. In particular, this method would impose severe limitations on novels with little or unmarked dialogue (such as Cormac McCarthy's *The Road*) or no dialogue at all (such as in Margaret Yourcenar's *Memoirs of Hadrian*). Unlike conversational networks, a co-occurrence network also captures non-spoken interactions (such as 'give', 'kiss', or 'kill'), which are crucial for example in novels in which silence is forced upon the characters (such as Margaret Atwood's *The Handmaid's Tale* or George Orwell's *Nineteen Eighty-Four*).

Figures 2 and 3 show respectively the manually created conversational[13] and co-occurrence networks[14] for chapter 7 of Charles Dickens' *David Copperfield*. The two networks look relatively dissimilar. The cause of this contrast is to be found in their definition. A conversational network aims to capture all spoken interactions marked with quotation marks, whereas a co-occurrence network aims to capture all characters who co-occur in the same scene. Both kinds of networks have advantages and drawbacks. On the one hand, conversational networks fail to capture a huge part of the interactions of a novel, namely unmarked spoken interactions (e.g. "Steerforth [...] was very angry with Traddles, and said he was glad he had caught it"), non-spoken interactions (e.g. "Mr. Creakle, looking hard at Mr. Mell, put his hand on Tungay's shoulder"), and lack of interaction but acknowledgement of presence ("I [...] found Mr. Creakle in the midst of us, with Tungay at his side, and Mrs. and Miss Creakle looking in at the door"). On the other hand, a co-occurrence network would connect characters that do not interact nor share presence in the scene, but who happen to

[13] Created in the manner of Elson et al. (2010).
[14] Using a paragraph as text window.

be mentioned in the same window of text by either the narrator or a third-party (e.g. "I was almost tempted that evening to tell Steerforth about pretty little Em'ly"). We consider that the advantages of using a co-occurrence network outweigh the disadvantages, since two characters mentioned in the same window of text will rarely be wholly unrelated.[15]

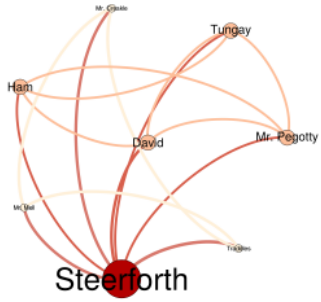

FIGURE 2 Conversational network of chapter 7 from *David Copperfield*.

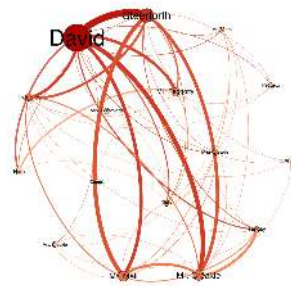

FIGURE 3 Co-occurrence network of chapter 7 from *David Copperfield*.

In our approach, the characters are linked if they co-occur in the same window of text, which we set to be a paragraph. The networks are **undirected** (the direction of the interaction is ignored) and have **weighted nodes** (the number of occurrences of the corresponding characters) and **weighted edges** (the number of paragraphs in which two characters co-occur). In novels with the first-person point of view, the narrator has been manually identified with the character who performs the narration and the off-dialogue occurrences of the pronoun "I" have been added to this node. We took this measure to avoid pushing the protagonist of a first-person novel to the background, but it is still far from optimal: the narrator may describe unwitnessed events, as Elson et al. (2010) remark.

We extract static and dynamic co-occurrence networks. A **static network** is a network that does not take the time dimension into account. It allows a better visualization of the novel as a whole, but the features extracted from it do not capture the evolution of characters throughout the novel. In our method, a **dynamic network** is a succession of subnetworks, one for each of the chapters into which the novel is divided. It incorporates the temporal dimension, and therefore the features that are extracted from a dynamic representation correspond to

[15]Only 10 of 403 mentions in this chapter are to characters absent from the scene.

an analysis of the development of the characters throughout the novel. We used the Python library `Networkx` to construct the networks, and the network analysis software `Gephi` to visualize them.[16]

4 Network analysis

The aim of extracting social networks from a novel is to turn it, a complex object, into a schematic representation of its core structure, taken from the interactions between its characters. In section 2, we mention the use of networks as a strategy to find the skeletal structure of a novel's plot. In this section we analyze whether networks can be said to represent a novel. We show how the static and dynamic networks together can provide a rough overview of the novel's narrative structure.

4.1 Static networks

In Figures 4-7, we show our static networks of, respectively, *Pride and Prejudice* by Jane Austen, *Vanity Fair* by William M. Thackeray, *The Island of Dr. Moreau* by H. G. Wells, and *Peter Pan* by J. M. Barrie.[17] Just a glimpse at the networks is enough to realize that these novels are very different when it comes to their character structure.

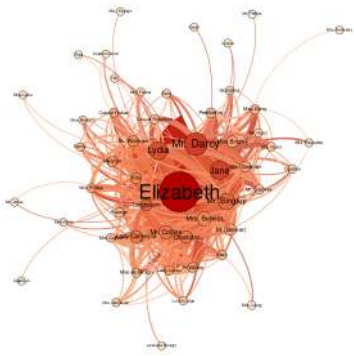

FIGURE 4 Static network of *Pride and Prejudice*, by Jane Austen.

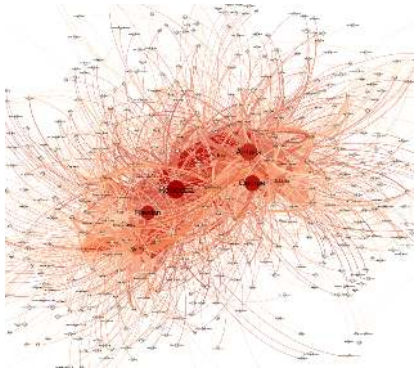

FIGURE 5 Static network of *Vanity Fair*, by William Thackeray.

Pride and Prejudice is the archetypal romantic comedy and is also often placed in the category of *Bildungsroman* with a heroine. It is not difficult to understand, just by looking at the network, that the heroine of this novel is the character who corresponds to the node 'Elizabeth', which is the most central and connected node of our graph. The second

[16] http://networkx.github.io and http://gephi.org

[17] We show network centers, ignoring all the isolated satellite nodes.

most central node, 'Darcy', is the protagonist's romantic interest. A more meticulous study of the graph gives us a good insight into how closely related each two characters are, and allows to identify small communities of characters (such as the Bennet family, the community around Bingley, and the community at Rosings Park).

The community represented in ***Vanity Fair*** could hardly be more different. This is a satirical novel with many elements of social criticism. Through the ironic voice of the narrator, the reader becomes aware of his describing much more than just the adventures and misfortunes of a collection of invented characters. As can be understood from the graph, the novel does not revolve around one only character as in the case of *Pride and Prejudice*, but instead there are four foci (the nodes 'Rebecca', 'Amelia', 'Rawdon' and 'George') and a large number of minor characters. *Vanity Fair* wants to depict in a satirical manner the society of the period, mostly in order to criticize it.

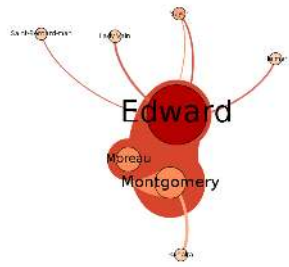

FIGURE 6 Static network of *The Island of Dr. Moreau*, by H. G. Wells.

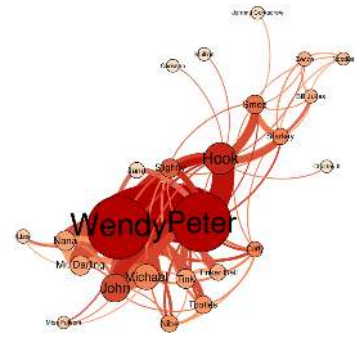

FIGURE 7 Static network of *Peter Pan*, by J. M. Barrie.

The Island of Dr. Moreau is a science-fiction and adventure novel in which two characters head into the unknown. There are three main characters in the novel, which can only communicate with each other due to the hostility surrounding them. This is a novel with a first-person narrator, 'Edward', who is also the protagonist.

Finally, ***Peter Pan*** is an example of children's literature. The community of this novel is again not too complex: there are two main nodes, 'Peter' and 'Wendy', and a third node with a high degree: 'Hook'. Around these three main characters, the Lost Boys, the Pirates and Wendy's family are organized according to their interactions with them.

A static network shows the skeleton of a novel. By looking at a static

representation of a novel as a graph, we can see its actors, with their relative importance, centrality in the plot, and interactions with each other, and we can understand the size and interconnectedness of the community of characters in the novel.

4.2 Dynamic networks

A dynamic network incorporates a key dimension of a novel: time, represented by a succession of chapters. Unlike static networks, dynamic networks allow us to have an insight into how characters appear, disappear and evolve. Bakhtin's concept of the *chronotope*, described in section 2, is presented here as the amalgamation of time (represented by the sequence of chapters) and place (represented by the contextual minor characters who surround the protagonists in each of the different locations of the novel). We illustrate this with an example.

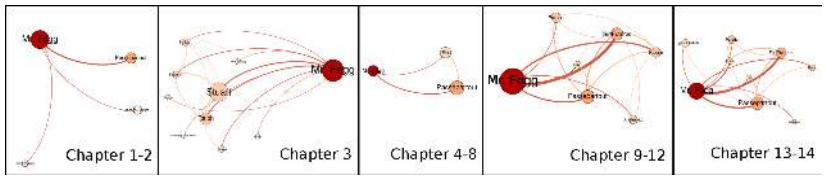

FIGURE 8 Dynamic representation of chapters 1-14 of *Around the World in Eighty Days*.

Figure 8 shows the dynamic network for the first fourteen chapters of Jules Verne's *Around the World in Eighty Days*. Unlike static networks, the weight and degree of the nodes varies according to the relevance of the characters in the present chapter, and characters absent from a chapter are absent from the corresponding network. For the purpose of illustration, we have grouped the chapters. The first two chapters take place in Mr. Fogg's house, where the protagonist is introduced to his new valet, Passepartout. In chapter 3, Mr. Fogg goes to his club, where he meets the other members (they are represented in the left hand of the graph). It is in this chapter that Mr. Fogg places the famous bet that he can travel around the world in 80 days. Chapters 4 to 8 correspond to travel preparations and Mr. Fogg's travel only with his valet Passepartout, but followed by Inspector Fix. Chapters 9-12 take place in India and thus we do not see any of the characters whom Mr. Fogg left in London. In chapter 13, Aouda appears for the first time. She will be Mr. Fogg's companion for the rest of his journey and the reader's companion for the rest of the novel (37 chapters in total).

This short description of the first chapters is possible only through a

dynamic network. In the static network, Aouda is a node permanently present in the network, rather than a character who does not appear until the second half of the novel and becomes one of the main characters from that moment on. In the static network, the group of very static gentlemen of a London club are sitting very close from the consul of Suez, a judge of Calcutta, or a captain of a transatlantic boat. All these characters would never co-occur (other than by mentions) in a dynamic network. The two kinds of networks are complementary, and together provide a naked structure of the novel, which, even if devoid of goals and actions, can provide a trustworthy approximation.

5 Selection of literary features

The features that we use in our clustering experiments are extracted from the static and dynamic networks of the novels and are meant to capture their narrative structure. They are intended to answer questions such as how central the protagonist is, how the minor characters interact with the protagonist, how the characters develop throughout the novel, etc.[18] The features extracted from a novel must form a faithful representation of the static and dynamic distributions of the novel's characters. That is why the definition of features is crucial to our work. A detailed list of features appears in the appendix.

5.1 Static network features

We define 40 features drawn from static networks. They can be arranged into four groups.

- **Features that describe the graph.** This group of features aims at capturing the character distribution of the network as a whole: how dense and wide the community is, how many isolated characters are introduced, what is the diameter and radius of the network, what is the proportion of eccentric and central nodes, etc.

[18] Alex Woloch, in his *The One vs. the Many* (Woloch, 2003), proposes a series of similar questions with which he means to redefine narrative characterization, an oft-neglected aspect of literary theory: "How often, at what point, and for what duration does a character appear in the text? How does she enter and exit specific scenes? [...] How are her appearances positioned in relation to other characters and to the thematic and structural totality of the narrative? Why does a particular character suddenly disappear from the narrative or abruptly begin to gain more narrative attention? How does the text organize a large number of different characters within a unified symbolic and structural system?" Such questions can be asked of any novel. Turning specifically to Homer, Woloch asks: "Is Achilles the controlling figure or only the most important of many vital characters who are all essential to *The Iliad*? Does centrality within the thematic and narrative structure of the *Iliad* distinguish Achilles quantitatively, or only by degree, from such memorable characters as Hector, Diomedes, Sarpedon, or Patroklos?"

- **Protagonist, minor characters, and isolates.** This group of features focuses on specific nodes and captures their presence and relevance in the static representation of the novel. It keeps information on how central and relevant the protagonist is in comparison to the other characters, the weight of the isolated nodes, the immediate connections of the protagonist, etc.
- **Metadata.** Each novel is annotated with metadata, including the title, size, number of chapters, and point of view of the novel. This group of features contains information drawn from the metadata.
- **Character gender.** As explained in section 3, the gender of each character is kept as an attribute for each node. The features of this group capture information about the gender of the characters of each novel, such as what is the percentage of male characters, or what is the gender of the protagonist.

5.2 Dynamic network features

As we already noted, we consider a dynamic network to be a sequence of networks, each matching a division of the novel (chapter, part, book and so on). Novels are divided in a very arbitrary fashion. Whereas some novelists opt for few divisions and thus long chapters, others prefer more divisions and shorter chapters. **Freytag's Pyramid** (Freytag, 1863) is a classical technique to analyze the structure of literary fiction, considering that each work of literary fiction can be divided into five acts: *exposition*, *rising action*, *climax*, *falling action*, and *dénouement*. In order to facilitate the comparison of dynamic networks, we have grouped each novel's chapters into these five acts, assuming that each act has the same length. This is of course a simplification: Freytag's Pyramid assumes that every novel follows a linear narrative structure, with no details about what percentage of the work belongs to what act.

There is, to our knowledge, no quantitative study on the dramatic structure of novels, so we simply consider each part to be of the same length.[19] For novels that have fewer than five chapters, dynamic networks could not be created and thus the features were set to 'unknown'. We define 15 features drawn from dynamic networks, which show the continued or discontinued presence of the protagonist throughout the novel, the proportion of characters in each of the five acts, and the proportion of characters appearing in only one act.

[19] Matthew Jockers has recently released an R package which reveals plot arcs in a narrative (http://www.matthewjockers.net/2015/02/02/syuzhet/). We believe that it contains interesting ideas on dealing with the problem of variable lengths in novels. We plan to implement them in future experiments.

6 Experiments

At the beginning of this paper we ask ourselves whether the structure of the network of characters can be used to identify literary genres and whether it carries the fingerprints of its author. We propose two main experiments, which we describe below.

6.1 Document clustering by genre

Data collection.[20] In order to create a dataset for this experiment, we have tried to collect a representative sample of the most influential novels of the Western world since the beginning of the novel as a literary form. The resulting dataset consists of 238 novels obtained from Project Gutenberg.[21] Each novel has been annotated with the genre to which it belongs. This was not a trivial task. As we note throughout the paper, there is no such thing as the one correct genre for a novel. Sources differ in categorizing the same novels, some novels are labeled with more than one genre, and some novels are not categorized at all. The process of building and labeling the corpus has therefore been long and laborious.

The decision on the number of genres was based on observation, resulting in **11 most seen genres**: adventure, historical, romance, satirical, *Bildungsroman*, picaresque, mystery, gothic, social criticism, science fiction, and children's fiction. In order to annotate the data, we compared various sources, among them the study guides from Spark Notes and Schmoop,[22] the social-cataloging website for books Goodreads,[23] Wikipedia,[24] and a variety of literary research studies for each particular novel. Each novel has been annotated with a maximum of three genres in those cases when the sources did not agree on one.

Experimental setup. Treating our problem as an unsupervised task brings a definite advantage: training assumes that the categories are known. In genre identification there is no consensus, no agreement, on the ideal category (or categories) for each novel. The unsupervised approach allows us to categorize the novels without bias, avoiding all the subjectivity which the training labels might contribute. We propose four different set-ups: the **enCorpus** is the set of 184 novels originally written in English; the **trCorpus** is the set of 54 novels originally not written in English, in their translated version; the **alCorpus** is the whole dataset, 238 novels; and the **19Corpus** is a subset of 118 British

[20] The complete list of works used for the genre experiment can be found in http://www.coli.uni-saarland.de/∼csporled/SocNetNovels/corpusGenres.pdf
[21] Source: http://www.gutenberg.org
[22] http://www.sparknotes.com/ and http://www.shmoop.com/literature/
[23] http://goodreads.com/
[24] http://www.wikipedia.org/

novels from the 19th century.

6.2 Document clustering by author

Data collection.[25] We could not use the same dataset as for clustering by genre, because few authors have more than one novel in it. Instead, we collected 45 novels by seven 19th-century authors: five British writers (Jane Austen, Charles Dickens, Elizabeth Gaskell, George Eliot and William Thackeray), and two Russian realists (Ivan Turgenev and Fyodor Dostoyevsky). We also included the fantasy series of *Harry Potter* novels, by the contemporary British author J. K. Rowling.

Experimental setup. Again, we propose four different set-ups, according to the author's country of origin or epoch. In Table 4, we show the detailed list of authors per each segment of the corpus.

Corpus#	Authors
`Corpus1`	Austen, Dickens, Thackeray, Eliot, Gaskell
`Corpus2`	Austen, Dickens, Thackeray, Eliot, Gaskell, Dostoyevsky, Turgenev
`Corpus3`	Austen, Dickens, Thackeray, Eliot, Gaskell, Rowling
`Corpus4`	Austen, Dickens, Thackeray, Eliot, Gaskell, Dostoyevsky, Turgenev, Rowling

TABLE 4 Authors in each corpus segment.

7 Results

The goal of the experiments was to investigate the extent to which the genre and author of a novel can be detected from its network of characters. To this end, we compared our network-based clustering with a clustering based on the novel's content using a bag-of-words approach – a strong baseline. To assess the overall difficulty of the task, we also provided a simpler baseline: assign all novels to the most frequent genre (or author). We apply the **EM** clusterer,[26] taken from Weka (Hall et al., 2009), pre-defining the number of clusters to the expected number.[27]

Clustering is not trivial to evaluate, because the labels of the returned clusters are not known. We perform **Classes to clusters** evaluation. In this mode, Weka first ignores the class attribute and generates the clustering. In the test phase, it assigns classes to the clusters based

[25] The complete list of works used for the author experiment can be found in http://www.coli.uni-saarland.de/∼csporled/SocNetNovels/corpusAuthors.pdf

[26] http://weka.sourceforge.net/doc.dev/weka/clusterers/EM.html

[27] The expected number is 11 in clustering by genre – see section 6.1. In clustering by author, it is 5, 7, 6 or 8 for corpora 1-4 respectively – see section 6.2.

on the majority value of the class attribute in each cluster (as long as no other cluster has a higher proportion of items of the same class).

Corpus#	**All-In-One**			**BoW**			**Our approach**		
Metric	Pr	Re	F_1S	Pr	Re	F_1S	Pr	Re	F_1S
`enCorpus`	0.18	1.00	0.31	0.25	0.40	0.31	0.27	0.42	0.33
`trCorpus`	0.31	1.00	0.48	0.21	0.35	0.27	0.21	0.34	0.26
`alCorpus`	0.20	1.00	0.34	0.26	0.46	0.34	0.21	0.37	0.27
`19Corpus`	0.25	1.00	0.39	0.27	0.47	0.34	0.31	0.54	0.40

TABLE 5 Genre clustering evaluation.

Table 5 shows the results for the baselines and for our approach. When a novel is classified into one of the correct classes, we consider the classification to be correct. We can see that the absolute numbers are low. The performance is slightly better for works originally written in English (`enCorpus` and `19corpus`). The reason why the `19Corpus` performs better than the rest of the collections is probably the fact that it is the most local collection – it only contains British novels – and covers the shortest time span. The other collections contain documents from very different ages (up to five centuries between the first and the last novel) and countries of origin. A novel can be said to a mirror of the society of the moment, so it is not surprising that the more local a collection of texts, the better our method performs.

Corpus#	**All-In-One**			**BoW**			**Our approach**		
Metric	Pr	Re	F_1S	Pr	Re	F_1S	Pr	Re	F_1S
`Corpus1`	0.31	1.00	0.48	0.88	0.69	0.77	0.71	0.67	0.69
`Corpus2`	0.24	1.00	0.39	0.64	0.54	0.59	0.65	0.67	0.66
`Corpus3`	0.26	1.00	0.42	0.66	0.73	0.69	0.74	0.77	0.75
`Corpus4`	0.21	1.00	0.35	0.63	0.48	0.54	0.67	0.60	0.63

TABLE 6 Author clustering evaluation.

Table 6 shows the results of clustering by author. As can be seen, the performance of both the BoW baseline and our approach in clustering by author is much higher than by genre. Interestingly enough, both return parallel results in both clustering tasks even though their respective feature vectors could hardly be more different. We see how the performance of the BoW baseline approach decreases as the corpus becomes less local, but also as the number of authors increases. Our approach does not suffer considerably from the increasing number of classes into which to cluster.

7.1 Analysis of the genre experiment

Genres are not clear and distinct classes; we have already discussed the difficulty of creating an annotated corpus of novels. We selected the eleven genres in our annotation by considering the most frequently seen novelistic subgenres. It was thus no big surprise that our clusters would not correspond completely to the external classification. However, by observing the 'incorrectly labeled' cases from our network-based approach, we find some interesting patterns. In order to conduct a closer analysis of the clustered data, we decided to run binary experiments.

Two genres that our method conflates almost completely are *mystery* and *adventure*. A binary clustering using only these two genres fails completely at differentiating them, since both have in average similar values for each feature. A similar confusion of genres occurs with the *historical*, *social*, and to a lesser degree *satirical* genres. Novels of these genres are often misclassified into one another. We can see how these three genres are somewhat intertwined: social criticism may be carried out through a satirical novel (as in *Vanity Fair*) or as a historical novel (as in *War and Peace*). Our method tends to classify these three genres indistinctly together, and this might well be because of their similar structural characteristics.

Three genres that our method distinguishes well are *social criticism*, *children's fiction* and *science fiction*. Social novels usually have low density, and removing the protagonist barely affects the density of the network (when compared to science fiction, there is a clear-cut division in terms of difference of density with and without the protagonist). The high proportion of eccentric nodes also indicates a social novel. Dynamic features do not have a large and distinctive contribution to the general clustering, but they do contribute when one looks at particular pairs of genres, such as *picaresque* and *children's fiction*, where the proportion of characters in each act provides a clear indication of the genre.

7.2 Analysis of the author experiment

The clustering of novels according to their authors does not have any of the complexities of the clustering by genre, but very interesting patterns can be found when the clusters are analyzed in detail. One can learn, for instance, that the network structure of Jane Austen novels and of William Thackeray novels are very different, as a look at Figures 4-5 already suggested. These two authors are, alongside J. K. Rowling, the easiest to identify. In fact, a clustering of only the novels by these three authors results in a clear-cut grouping with no misclassifications. The authors most difficult to identify are Dickens and Eliot.

An in-depth study of the role of each feature in the clustering task

provides a very interesting view of the literary work of each author. In our dataset, female writers (in particular Austen and Gaskell) depict a society with a higher proportion of female characters than male writers (in particular Dickens, Turgenev, and Dostoyevsky), whereas Thackeray and Rowling reproduce a more equal society. The very low graph density of Thackeray's novels contrasts with the high density of Austen's and Turgenev's novels, whereas Gaskell's novels have all a strikingly similar graph density.

Now, graph density alone does not distinguish the different authors, and sometimes it can even vary within the same author, as is the case of the *Harry Potter* books, the first ones being considerably denser than the last ones, as the community represented in them becomes broader and less tightly knit with every new book. The role of the protagonist in novels seems to also be an important author choice. It is very prominent in the works by Austen, Gaskell and Rowling, in which its presence is constant throughout the novel. Turgenev's protagonists are also very strong, even though their presence varies along the novel. Thackeray gives less weight to the main characters, whereas minor characters and isolates assume larger roles. Indeed, in his novels, the border between main and secondary characters is more blurred than in other authors.

The dynamic features show the different distributions of characters over the time of the novel. We can see how Rowling introduces most of her characters in the rising action, while it is not until the falling action in the case of Austen, or the dénouement in the case of Eliot. Dynamic features help to see how a high proportion of characters in Thackeray's novels appear in only one stage of the novel, and then disappear. On the other side of the spectrum, Austen's and Dostoyevsky's characters arrive in the novel to stay.

7.3 Discussion

The main difference between clustering by genre and clustering by author is that, whereas there is no (or, should we say, little) doubt who wrote a novel, there is on-going discussion on what genre it belongs to. We proposed a tentative ground truth, which worked well enough for this project. The preceding subsections show some of the most evident conclusions that can be drawn from looking at the clusters and the effects that the features have on them. It is by no means an exhaustive or complete analysis. Even so, if we analyze in depth the contents of the clusters and the role that each feature plays, we may gain better understanding of the nature of our classification and of the task at hand. A logical next step would be to combine both kinds of features, content-based and network-based, to see if this yields better results.

We could also ask whether we worked with a suitable corpus. Attempting a classification of the Western literary canon may always lead to frustration, for many of the novels that entered it broke the mould of the literary production of their time, setting new standards. As future work, we plan to apply our method to present-day literature, which we expect in most of the cases to yield a more clear-cut classification of the novels.

Finally, we have seen how the author's fingerprints are visible in the social networks of novels. Bamman et al. (2014) take first steps toward accounting for the influence of unwanted signals (such as, in our genre experiment, the author of the novel) over the wanted information. Any further work should take this into consideration.

8 Conclusion

This work is a contribution to the field of quantitative literary analysis. We have presented a method of building static and dynamic social networks from novels as a way of representing structure and plot. Our main goal was to understand the role that the network structure of a novel plays in determining the genre to which the novel belongs. A secondary goal was to learn to what extent the network structure of the novel is an indication of its author's style. Two experiments have been designed to address these problems, treated as unsupervised document classification tasks. In the case of clustering according to genre, the results were on par with the bag-of-words baseline, and, when analyzed qualitatively, show that the approach is promising, even though much remains to be explored. The second experiment, clustering according to the author, again produced results similar to those of the bag-of-words baseline (and even slightly higher), which indicates that the representation of novels as social networks carries the author fingerprints. Authorship attribution is mostly used for either forensic purposes or plagiarism identification. We have shown, however, that an analysis of the features and clustering can also be used to explore structural similarities between authors.

Acknowledgments

We would like to thank the anonymous reviewers for their detailed and insightful comments.

References

Aaronson, Scott. 2001. Stylometric Clustering: A Comparison of Data-Driven and Syntactic Features. Tech. rep., Computer Science Department, University of California, Berkeley.

Abbott, H. Porter. 2008. *The Cambridge Introduction to Narrative*. Cambridge Introductions to Literature. Cambridge University Press.

Agarwal, Apoorv, Augusto Corvalan, Jacob Jensen, and Owen Rambow. 2012. Social Network Analysis of Alice in Wonderland. In *Workshop on Computational Linguistics for Literature, Association for Computational Linguistics*, pages 88–96.

Akiva, Navot and Moshe Koppel. 2012. Identifying Distinct Components of a Multi-Author Document. In *Proceedings of the 2012 European Intelligence and Security Informatics Conference*, pages 205–209.

Alberich, Ricardo, Josep Miró-Julià, and Francesc Rosselló. 2002. Marvel Universe looks almost like a real social network. Preprint, Department of Mathematics and Computer Science, University of the Balearic Islands.

Allison, Sarah, Ryan Heuser, Matthew Jockers, Franco Moretti, and Michael Witmore. 2011. Quantitative Formalism: an Experiment. Pamphlet 1, Stanford Literary Lab.

Andrews, Nicholas O. and Edward A. Fox. 2007. Recent Developments in Document Clustering. Tech. rep., Department of Computer Science, Virginia Tech.

Aristotle. 2007. *Poetics*. The Internet Classics Archive - Atomic and Massachusetts Institute of Technology.

Baayen, Harald, Hans van Halteren, and Fiona Tweedie. 1996. Outside the cave of shadows: using syntactic annotation to enhance authorship attribution. *Literary and Linguistic Computing* 11:121–131.

Bakhtin, Mikhail. 1981a. Epic and Novel: Towards a Methodology for the Study of the Novel. In J. M. Holquist, ed., *The dialogic imagination: four essays*. Unversity of Texas Press.

Bakhtin, Mikhail. 1981b. Forms of Time and of the Chronotope in the Novel: Notes Toward a Historical Poetics. In J. M. Holquist, ed., *The dialogic imagination: four essays*. Unversity of Texas Press.

Baldick, Chris. 2008. *The Oxford dictionary of literary terms*. Oxford Paperbacks. Oxford University Press.

Bamman, David, Ted Underwood, and Noah A. Smith. 2014. A Bayesian Mixed Effects Model of Literary Character. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Voluma 1: Long Papers)*, pages 370–379. Baltimore, Maryland: Association for Computational Linguistics.

Basili, Roberto, Paolo Marocco, and Daniele Milizia. 2008. Semantically rich spaces for document clustering. In *Proceedings of the 19th International Conference on Database and Expert Systems Application*, DEXA Workshops, pages 43–47. IEEE Computer Society.

Bekkerman, Ron, Hema Raghavan, James Allan, and Koji Eguchi. 2007. Interactive Clustering of Text Collections According to a User-Specified Criterion. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 684–689.

Bemong, Nele and Pieter Borghart. 2010. State of the Art. In N. Bemong, P. Borghart, M. D. Dobbeleer, K. Demoen, K. D. Temmerman, and B. Keunen, eds., *Bakhtin's Theory of the Literary Chronotope: Reflections, Applications, Perspectives*, chap. 1, pages 3–16. Academia Press.

Bloom, Harold. 2003. Introduction. In M. de Cervantes (translation by E. Grossman), *Don Quixote*. HarperCollins Publishers.

Celikyilmaz, Asli, Dilek Hakkani-Tur, Hua He, Greg Kondrak, and Denilson Barbosa. 2010. The actor-topic model for extracting social networks in literary narrative. In *NIPS Workshop: Machine Learning for Social Computing*.

Elsner, Micha. 2012. Character-based Kernels for Novelistic Plot Structure. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012)*. Avignon, France.

Elson, David K., Nicholas Dames, and Kathleen R. McKeown. 2010. Extracting Social Networks from Literary Fiction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*.

Elson, David K. and Kathleen R. McKeown. 2010. Automatic Attribution of Quoted Speech in Literary Narrative. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*.

Emerson, Caryl. 1986. *Boris Godunov: Transpositions of a Russian Theme*. Indiana-Michigan Series in Russian and East European Studies. Indiana University Press.

Finkel, Jenny Rose, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics*, pages 363–370.

Fowler, Alastair. 1982. *Kinds of Literature: An Introduction to the Theory of Genres and Modes*. Oxford: Clarendon Press.

Freytag, Gustav. 1863. *Die Technik des Dramas*. S. Hirzel.

Gupta, Suhit, Hila Becker, Gail Kaiser, and Salvatore Stolfo. 2005. A Genre-based Clustering Approach to Content Extraction. Tech. rep., Department of Computer Science, Columbia University.

Hall, Mark, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA Data Mining Software: An update. *SIGKDD Explorations* Volume 11(1):10–18.

Holmes, David I. and Richard S. Forsyth. 1995. The Federalist Revisited: New Directions in Authorship Attribution. *Literary and Linguistic Computing* 10:111–127.

James, Henry. 1908. Preface to volume 7 of the New York edition (containing: The tragic muse). `http://www.henryjames.org.uk/prefaces/text07.htm`.

Jayannavar, Prashant Arun, Apoorv Agarwal, Melody Ju, and Owen Rambow. 2015. Validating Literary Theories Using Automatic Social Network Extraction. In *Proceedings of NAACL-HLT Fourth Workshop on Computational Linguistics for Literature*, pages 32–41.

Jockers, Matthew L. 2013. *Macroanalysis: Digital Methods & Literary History*. University of Illinois Press.

Layton, Robert, Paul Watters, and Richard Dazeley. 2011. Automated unsupervised authorship analysis using evidence accumulation clustering. *Natural Language Engineering* 19:95–120.

Ledger, Gerard and Thomas Merriam. 1994. Shakespeare, Fletcher, and the two noble kinsmen. *Literary and Linguistic Computing* 9:235–248.

Madsen, Deborah L. 1994. *Rereading Allegory: A narrative approach to genre*. Palgrave Macmillan.

Margonari, Massimiliano. 2011. An Unsupervised Text Classification Method Implemented in Scilab. Tech. rep., Open Source Engineering.

Moretti, Franco. 2011. Network Theory, Plot Analysis. Pamphlet 2, Stanford Literary Lab.

Newman, Mark E. J. and Michelle Girvan. 2003. Finding and evaluating community structure in networks. *Physical Review E* 69:1–16.

Pavlyshenko, Bohdan. 2012. The Clustering of Author's Texts of English fiction in the vector space of semantic fields. *The Computing Research Repository* abs/1212.1478.

Poudat, Céline and Guillaume Cleuziou. 2003. Genre and Domain Processing in an Information Retrieval Perspective. In *Proceedings of the International Conference on Web Engineering (ICWE)*, pages 399–402.

Propp, Vladimir I. A. 1968. *Morphology of the folktale*. American Folklore Society Bibliographical and Special Series. University of Texas Press.

Propp, Vladimir I. A. 1984. *Theory and History of Folklore*, vol. 5 of *Theory and history of literature*. Manchester University Press.

Rydberg-Cox, Jeff. 2011. Social Networks and the Language of Greek tragedy. *Journal of the Chicago Colloquium on Digital Humanities and Computer Science* 1:1–11.

Sack, Graham. 2012. Character Networks for Narrative Generation. In *Intelligent Narrative Technologies: Papers from the 2012 AIIDE Workshop, AAAI Technical Report WS-12-14*, pages 38–43.

Shahnaz, Farial, Michael W. Berry, V. Paul Pauca, and Robert J. Plemmons. 2006. Document Clustering Using Nonnegative Matrix Factorization. *Information Processing and Management* 42(2):373–386.

Spang, Kurt. 1993. *Géneros literarios*. Teoría de la literatura y literatura comparada. Madrid, Spain: Editorial Síntesis.

Steinbach, Michael, George Karypis, and Vipin Kumar. 2000. A Comparison of Document Clustering Techniques. Tech. rep., Department of Computer Science and Egineering, University of Minnesota.

Suen, Caroline, Laney Kuenzel, and Sebastian Gil. 2013. Extraction and Analysis of Character Interaction Networks From Plays and Movies. Digital Humanities Conference abstracts.

Wang, Xufei, Jiliang Tang, and Huan Liu. 2011. Document Clustering via Matrix Representation. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*, ICDM '11, pages 804–813. IEEE Computer Society.

Willett, Peter. 1988. Recent Trends in Hierarchic Document Clustering: A Critical Review. *Information Processing & Management* 24:577–597.

Woloch, Alex. 2003. *The One vs. the Many: Minor Characters and the Space of the Protagonist in the Novel*. Princeton, New Jersey: Princeton University Press.

Zhang, Bin. 2013. *Learning Features for Text Classification*. Ph.D. thesis, University of Washington.

Appendix

Description of the features

Graph metrics

Feature	Explanation
graphDensityW/oProtAndIsol	Float value. Density of the graph without the protagonist and the isolate nodes.
graphDensityW/oIsolates	Float value. Density of the graph without the isolate nodes.
graphDensityProtagonist1:3	Boolean value. True if difference of densities of the graph with and without the protagonist is significant (higher than 0.05).
graphDensityProtagonist2:3	Boolean value. True if difference of densities of the graph with and without the protagonist is non-significant (lower than 0.01).
densityDifference:3	Float value. Difference of density with and without protagonist.
graphDensityProtagonist:1	Boolean value. True if difference of densities of the graph with and without the protagonist is non-significant.
densityDifference:1	Float value. True if difference of densities of the graph with and without the protagonist is lower than 0.01 (non-significant difference).
proportionOfIsolates	Float value. Proportion of nodes that are isolate.
sizeOfGraph	Float value. Number of nodes of the graph in relation to the graph with the most nodes.
edgeSizeOfGraph	Float value. Number of edges of the graph in relation to the biggest graph with the most edges.
averageClustering	Float value. Network average clustering coefficient.
averageClusteringWithoutMain	Float value. Network average clustering coefficient in the graph without the protagonist.
diameter	Float value. Diameter of the graph.
radius	Float value. Radius of the graph.
proportionEccentrics	Float value. Proportion of eccentic nodes.
proportionCentrals	Float value. Proportion of central nodes.

Protagonist and isolates

Feature	Explanation
mostSociableClusterCoeff1	Float value. Clustering coefficient of the node corresponding to the character with the highest degree is calculated over the complete graph.
mostSociableClusterCoeff2	Float value. Idem over the graph when isolate nodes are removed.
mostSociableClusterCoeff3	Float value. Idem over the graph when protagonist and isolate nodes are removed.
relativeWeightOfMain:3	Float value. Fraction between the weight of the protagonist (node with the highest weight) and the sum of the weights of the rest of the nodes in novels with 3^{rd} person point of view.
relativeWeightOfMain:1	Float value. Idem in novels with 1^{st} person point of view.
relativeWeightOf2ndMain	Float value. Fraction between the weight of the second protagonist (node with the second highest weight) and the sum of the weights of the rest of the nodes except for the first protagonist.
relativeWeightOf10Most	Float value. Fraction between the weight of the 10 main characters (10 nodes with the highest weights) without the protagonist and the sum of the weights of the rest of the nodes.
relativeWeightOf10Least	Float value. Fraction between the weight of the 10 most minor characters (10 nodes with the lowest weights) excluding isolate nodes and the sum of the weights of the rest of the nodes except for the first protagonist.
relativeWeightOfIsolates	Float value. Fraction between the cumulated weight of all the isolate nodes and the sum of weights of the rest of the nodes.
edgesOfMostSociable:1	Float value. Proportion of edges of the most sociable character (node with the highest degree) in 1^{st} person novels.
edgesOfMostSociable:3	Float value. Proportion of edges of the most sociable character (node with the highest degree) in 3^{rd} person novels.

Features using metadata

Feature	Explanation
1pNovel_protagonistInTitle	Boolean value. True if the protagonist (character with the highest weight attribute) is in title in novels with 1^{st} person point of view.
3pNovel_protagonistInTitle	Boolean value. True if the protagonist (character with the highest weight attribute) is in title in novels with 3^{rd} person point of view.
1pNovel_protagonistNarrator	Boolean value. True if protagonist (character with the highest weight attribute) is the narrator. Only in 1^{st} person novels.
pointOfView:1	Boolean value. Point of view is 1^{st} person.
pointOfView:3	Boolean value. Point of view is 3^{rd} person.
pointOfView:mixed	Boolean value. Point of view is mixed.
smallNovel	Float value. Size of the novel is at maximum one tenth of the biggest novel.
numberChapters	Float value. Proportion of chapters with respect to the novel with highest number of chapters.
narratorUnknown	Boolean value. Narrator is never introduced to the reader.

Gender of the nodes

Feature	Explanation
diffGenders2MainNodes	Boolean value. True if the two nodes with the highest weight have different genders.
allMale3MainNodes	Boolean value. True if the three nodes with the highest weight are all male.
relativeMale	Float value. Proportion of male characters out of all nodes where gender is known.
relativeFemale	Float value. Proportion of female characters out of all nodes where gender is known.

Dynamic features

Feature	Explanation
protagonistInAll:3	Boolean value. True if protagonist is in all chapters of the novel (point of view: 3^{rd}) .
protagonistInAll:1	Boolean value. True if protagonist is in all chapters of the novel (point of view: 1^{st}).
chaptersWithProt:3	Float value. Proportion of characters in which the protagonist appears (point of view: 3^{rd}).
chaptersWithProt:1	Float value. Proportion of characters in which the protagonist appears (point of view: 1^{st}).
propCharactersExposition	Float value. Proportion of characters in the exposition with respect to total number of characters of the novel.
propCharactersRisingAction	Float value. Proportion of characters in the rising action with respect to total number of characters of the novel.
propCharactersClimax	Float value. Proportion of characters in the climax with respect to total number of characters of the novel.
propCharactersFallingAction	Float value. Proportion of characters in the falling action with respect to total number of characters of the novel.
propCharactersDenouement	Float value. Proportion of characters in the dénouement with respect to total number of characters of the novel.
onlyExposition	Float value. Proportion of characters appearing only in the exposition.
onlyRisingAction	Float value. Proportion of characters appearing only in the rising action.
onlyClimax	Float value. Proportion of characters appearing only in the climax.
onlyFallingAction	Float value. Proportion of characters appearing only in the falling action.
onlyDenouement	Float value. Proportion of characters appearing only in the dé.
onlyOneStage	Float value. Proportion of characters appearing only in one of the five stages.