ECOLE
POLYTECHNIQUE
DE BRUXELLES

ULB

UNIVERSITÉ LIBRE DE BRUXELLES

Etude de la topologie de réseaux d'acteurs extraits à partir de romans célèbre

Ligne du sous-titre du mémoire

Mémoire présenté en vue de l'obtention du diplôme
d'Ingénieur Civil en informatique à finalité spécialisée

[Antoine Heymans]

Promoteur
Professeur [Hugues Bersini]

Service
[Iridia]

Année académique
2019 - 2020

Contents

Chapter 1

Introduction

Social network analysis is an interdisciplinary discipline originally developed under the influence of sociology and mathematics (Scott, 1988). It consist of using graph and network theory to represent relations among actors as a network in order to analyze them. Originally the “actor” used in the network are people but in some field use it to represent abstract entities such as organizations related between them by financial exchange (Thilagam, 2010). This makes Social Network Analysis useful in various field like geography (Ter Wal and Boschma, 2009) or organizations management (Ribeiro et al., 2017). It has been used in psychology (Westaby et al., 2014) as it ”provides a powerful set of tools for describing and modeling the relational context in which behavior takes place, as well as the relational dimensions of that behavior”(Butts, 2008). It has also concrete applications outside of the academic world: Burcher and Whelan (2017) provides several interviews with operational agent of Criminal Intelligence Analysis using Social Network Analysis and suggests more collaboration between researcher and practitioners using it. Social network analysis have also be designated as a way to enforce laws and disrupt gang by the The International Association of Chiefs of Police (Crocker, nd).

Research in computer science are developing semantically-oriented techniques to analyze fiction. Elson et al. (2010) had presented a method to extract social network from literary which allows to apply Social Network Analysis techniques on it. Quang Dieu and Jung (2015) presented a method to extract a social network from movies. Automated method that analyze text of fiction also produce metadata that help to analyze them. For instance Waumans et al. (2015) produce the social network novels but also count the mean distance between two dialogs in the novel. This measure produce meaningful result when it is used to classify novels. Social network can be seen as a metadata with the particularity of being present in any fiction no matter the support of this fiction. Other metadata are only measurable in movies, novels...

This master thesis consists firstly in the development of a software that extracts social networks and other metadata from novels and movie scripts. Secondly it consists in the analysis of the topology of the extracted networks. The software has been originally developed during an other master thesis (Nicodème, 2015) and further advancement have been achieved in Waumans et al. (2015). It was only working on novels. For this reason I will focus on the modification that I have made and the evolution of the state of the art since this period. The global way the software works will also be explained more briefly.

Chapter 2

Social Network

Before extracting or analyzing any social networks, let's introduce the notion of social network and the basic concept related to network theory. Social network are defined as "a network of social connections and personal relationships between people" (Tabassum et al., 2018). Example are given on table 2.1. In social network analysis, relationships between people are represented using network in order to analyze them with mathematical tools provided by network and graph theory. "It helps in understanding the dependencies between social entities in the data, characterizing their behaviors and their effect on the network as a whole and over time." (Tabassum et al., 2018) We should note that in computer science, the word network and the word graph have the same meaning. The choice of these words usually depends on the application. For instance "social networks" are never called "social graphs" but it would represent the same concept. The word "graph" is more used in mathematics while the word "network" is most common in engineering (Estrada, 2013). In this work, the word network will be used most of the time to avoid confusion.

2.1 Network Theory: definition

Network theory is the study of networks. A network or a graph are defined mathematically as a pair of set $G = (N, E)$ such that $E \subseteq [N]^2$. Concretely it means that element from the second set should be tuple of 2 element from the first set. Graph are usually represented with points symbolizing element from the first set linked by lines representing element from the second set. You can see an example of this representation on figure X. In network theory, element from the first set are called "nodes" and element from the second set are called "links" or "edges". If for any pair $XY \in N$, the link $(X, Y) = (Y, X)$ then the graph is called undirected. This is the most common type of graph. There exist also directed graphs whose edges are directed from a sender to a receiver. In this work we will focus on undirected graphs. (Diestel, 2005)

In many applications, values are associated to nodes or link of the network. Often, a numerical value is associated to links and called weight. Those values are considered to be part of the network

Examples	Applications
Friendship networks	College/school students Organizations or web(Facebook, MySpace, etc.)
Follower networks	Twitter, LinkedIn, Pinterest, etc.
Preference similarity	networks Pinterest, Instagram, Twitter, etc.
Interaction networks	Phone calls, Messages, Emails, Whatsapp, Snapchat, etc.
Co-authorship networks	Dblp, Science direct, Wikibooks, other scientific databases, etc.
User-user citation networks	Dblp, Science direct, Wikibooks, other scientific databases, etc.
Spread networks	Epidemics, Information, Rumors, etc.
Co-actor networks	IMDB, etc.

Table 2.1: Example of social networks (Tabassum et al., 2018)

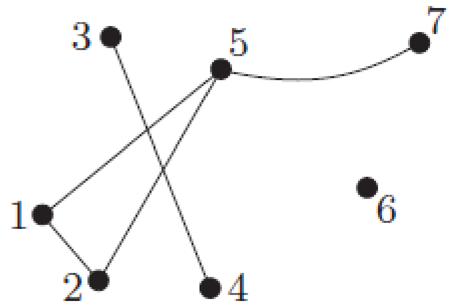

Figure 2.1: Network of Nodes = 1, ..., 7 and links = $\{(1,2),(1,5),(2,5),(3,4),(5,7)\}$ (Diestel, 2005)

even if it is not in the mathematical definition.

2.2 Network key notion

Some key notion are needed to understand this measure:

- **Neighborhood** : Two nodes are defined as *neighbors* if they are linked on a graph. The nodes $u, v \in G$ are neighbors if the edge $(u, v) \in G$. The *neighborhood* N_i , of a node i is the set of nodes that are neighbors with the node i.

- **Degree**: The degree, k_i of a node i is defined as the size of its neighborhood, which corresponds to the number of edges that are incident to the node i.

- **Path**: A path is a sequence of nodes such that for all pair of consecutive nodes i, j: the edge $ij \in E$ which means that the nodes are linked by an edge. The *Shorter Path* between two nodes is a path binding the 2 nodes such that there is no shorter path binding them. The notion of distance of a path can be defined as the sum of the weight along this path or as the number of nodes bounded by the path. The length of a shorter path is named *Geodesic Distance*. It can not be measured on disconnected nodes.

- **Connection**: Two nodes of a graph are connected if there exists a path in the graph that bind them. The nodes don't have to share an edge. To avoid any confusion, the nodes that share a same edge are referred here as linked. A graph is connected if all nodes from the graph are connected.

- **Clique**: A k-clique is a set of fully connected k nodes, a set of nodes K such that there is an edge $xy \in E \forall x, y \in K$.

- **Eccentricity**: The eccentricity,ϵ_i , of a node i is the maximum size of the shortest path that bind the node with an other node of the graph. It can be seen as the distance between the node and the further node from the graph. It can only be measured on connected graph.

(Diestel, 2005)

2.3 Network: Measure

There is multiple measures characterizing networks. I will give here the measures that are commonly used in social network analysis. Further explanations about the measures used in social network analysis are available at Tabassum et al. (2018).

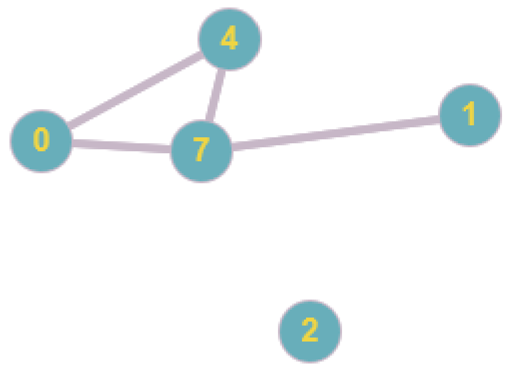

Figure 2.2: Graph of transitivity 1/3. They are 3 potential triangle: 0-4-7, 0-7-1 and 1-4-7. Only 0-4-7 is a triangle.

2.3.1 Mean degree

The mean degree is the mean value among the degree of all nodes. It can be computed more easily with the formula $\mu = \frac{|N|}{|E|}$ where $|N|$ is the total number of nodes and $|E|$ the total number of edges. It represents how much nodes of a network tends to share links or how much the network is connected.

2.3.2 Clustering coefficients

$$C_i = \frac{2 \cdot |e_{jk}|}{k_i \cdot (k_i - 1)} : i, k \in N_i, e_{jk} \in E \tag{2.1}$$

Different clustering coefficient may be computed.
The local clustering coefficient ,C_i, of a node i, is given by the formula 2.1 where E is the set of edges and N_i is the neighborhood of i. It represent how much the node is part of a connected neighborhood or how much the node is close to being part of a clique. In a group of friends, this measure is usually close to 1 as most of the friends know each other. In a network that represent love relationships, most of the methods will have a clustering coefficient close to 0 as romantic partners of an individual don't tend to share such a relations between each others.

As there is multiple global measure of graph clustering named "global clustering coefficient", we will give here the most popular. The first one have been given by Watts and Strogatz (1998). It's computed by taking the mean local clustering coefficient among all nodes. It is also named the mean local clustering coefficient.

The second global clustering coefficient is taken by counting the number of triangle in the graph divided by the number of "potential triangle". "Potential triangle" are set of 3 vertices such that one vertex is linked to the 2 others vertices. An illustration of this is given on figure 2.2. This measure is also called transitivity and will be named in that way in this master thesis to avoid any confusion.

Both of this measure are related to the probability of clique formation and transitivity. Network with high clustering coefficient tend to have cluster of nodes connected between them. Pair of nodes connected by a third node tend to be also linked by an edge. This is why the second measure is called "transitivity".

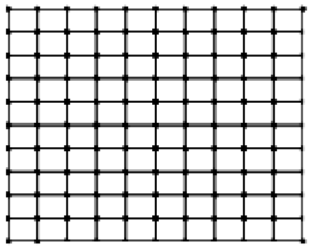

Figure 2.3: Square Lattice. (Svenson, 2006)

2.3.3 Radius and Diameter

The diameter from a graph is the greater eccentricity of its nodes. It can be used to know how far could potentially be actors in a network. The radius of a network is the smaller eccentricity of all its nodes. A small eccentricity indicates that a particular node is near from all the other nodes. However both of those measures are very sensitive to outliers and not very robust. As the eccentricity, it can not be measured on disconnected network.

2.3.4 Average Path length

The average path length (Albert and Barabási, 2002) or average geodesic distance is the mean distance of the shorter path between each pair of node from a connected network. It measures the distance between 2 random actors of the network. This measure can be compared with the radius and the diameter which gives close information but unlike them it is considered as robust.

2.4 Networks characterization

There exists a lot of networks. Of course a lot of them have human origin such as the internet network, electrical networks or social networks. But the notion of network can also be used to observe natural process as the set of vein of a body or the relation between metabolic and protein. Some networks seems build in a structured way, some of them seems random. Some networks have been humanly designed, some of them seems to evolutate from themselves. To characterize network we use topology and models.

Some networks have a very regular and easily-recognizable topology such as lattices, stars, circle. An example of it is available on figure 2.3. We also saw fully connected graphs on previous question. However most of the networks don't have a such regular topology. especially social networks. The structure of social network is not decided in advance, it's the product of a continuous growing due to local interactions among individuals. We can oppose regular graphs to random graphs, graphs that are the result to a random growth. Of course, some graphs are also the result of a process that includes randomness but can not be modeled using random graphs.

2.4.1 Random networks

2 popular models have been given to represent the growth of random network. In the first one, given a number of nodes N and a probability p, a graph is constructed with N nodes and each potential edge is added with a probability p. In average the number of edge that appears is $N \cdot (N-1) \cdot p$ with $0 < p < 1$. The second model takes a number of nodes N and a number of edge M as input. The M edges are chosen randomly without replacement in the set of $N \cdot (N-1)$ potential edges. Those models are asymptotically the same. The first model gives a variable

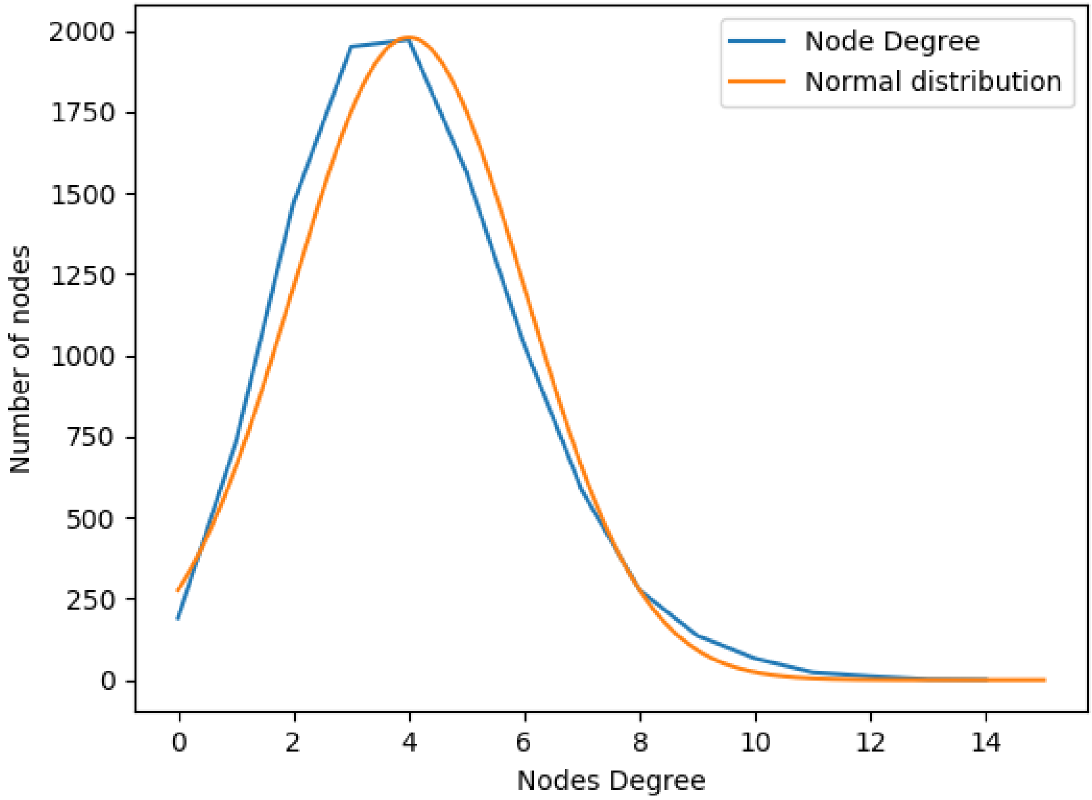

Figure 2.4: Random graph: degree distribution

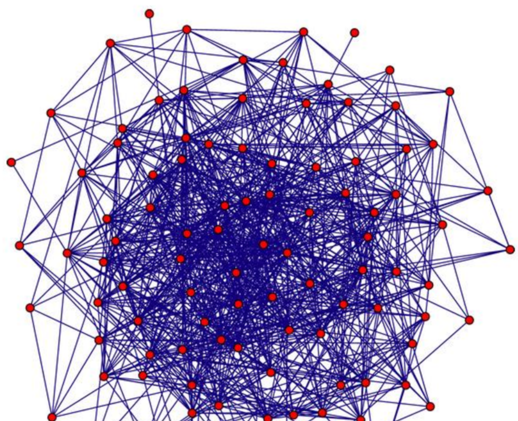

Figure 2.5: Example of random graph(Antoniou and Tsompa, 2008)

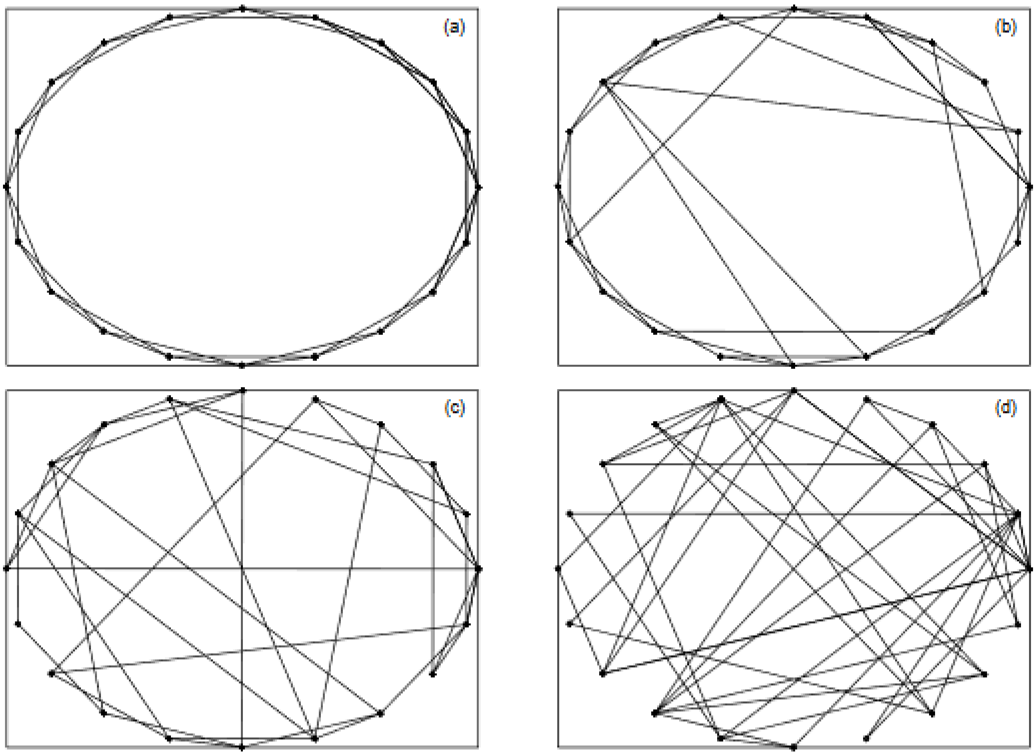

Figure 2.6: Construction of a small-world network (Barrat and Weigt, 2000)

number of edges but for big values of N, the networks generated with those models should share a common topology. This topology is mainly characterized by a number of edges approximately constant for a given couple (N, p) and by a distribution of degree. Figure 2.4 shows the plot of the degree distribution of a random graph obtained with the second model. The graph has 10000 nodes and 20000 edges, so the mean degree of its node is $\frac{2 \cdot E}{N} = 4$. the plot includes a normal distribution centered on 4. The degree distribution of random graphs follows normal distributions. An illustration of random graph is also given on 2.5 The mean path length of random networks is small and logarithmically with N. Due to random connection, nodes are likely to share connections with very distant nodes. Due to this random connections, their clustering coefficients are very small.

2.4.2 Small-world networks

Small world networks are an other type of networks that presents elements of randomness. Initially the goal of this classification was to represent networks that are highly clustered as latices but have an irregular topology and have a very short average path length between its nodes. A lot of social networks responds to this properties, for example network of friends are highly clustered: people sharing common friends are likely to be friends. Of course they are not regular as their evolve randomly and their average path length can be surprisingly small: Milgram (1967) have shown that the median number of intermediary needed to connect two randomly chosen people is six. Random networks don't have those properties as links are randomly added, node have the same probability to share an edge with each other node of the network. Some regular networks such as lattices are also highly clustered but they have a very long mean path length: as each vertex is only connected to its nearest neighbors, a path needs to go through a lot of cluster of neighbors before joining an opposite node. (Svenson, 2006)

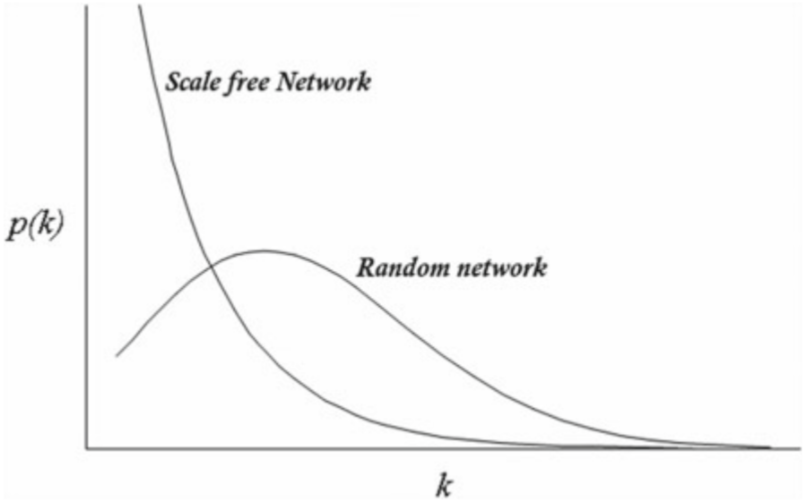

Figure 2.7: Comparison between the degree distribution of a random network- small world network and a scale free network (Kumar et al., 2018)

A first model for small-worlds network have been given by Watts and Strogatz (1998). The networks of N nodes, $N \cdot 2 \cdot k$ edges are constructed by the following algorithm taking a probability p as input:

1. A lattice of N site is drawn, each drawn being connected to its 2k neighbors.
2. For each nodes, each edge is removed with a probability p.
3. For each edge removed on a node, a new edge is added linking the node with an other node randomly chosen. The step 2 and 3 may be executed at once.

After this construction, a network is obtained with a high clustering thanks to the initial lattice structure. But the edges added in the last step allow to connect groups of neighbors that are fare for each other and reduce the mean path length. The construction of the network is illustrated on figure 2.6. An initial graph is shown on image (a) and the following image shows the network after the removal and additions of some links. The networks constructed with this model are characterized by being connected and having an almost constant degree for each node. Others models have been given for small-world network by following the same intuition of combining lattice with randomness and long range link. An common properties of those networks is that the mean-path length grows logarithmically with N and that their degree distribution is similar to random graphs.(Barrat and Weigt, 2000; Svenson, 2006)

2.4.3 Scale-Free networks

Scale free networks are a third type of network that present elements of randomness. It has been introduced by Barabasi (1999a). They are usually defined as networks having degree distribution following power laws: $P(k) \approx c \cdot k^{-y}$. An exemple of such distributions is given of figure 2.7. It means that a very small number of nodes present a very huge connectivity. This property is present in different networks such as the world-wide-web (Jeong and Barabasi, 1999) or the web of human sexual contacts (Liljeros et al., 2001). The constructions of scale-free network follows two principle. Firstly they *grow* and during this growth new node are added and linked with existing nodes. Secondly the creation of new nodes is done with *preferential attachment*: nodes have always more chance to be connected to nodes that already have a lot of edges.The growth and the preferential attachment are responsible of the creation of nodes having a lot of connections that

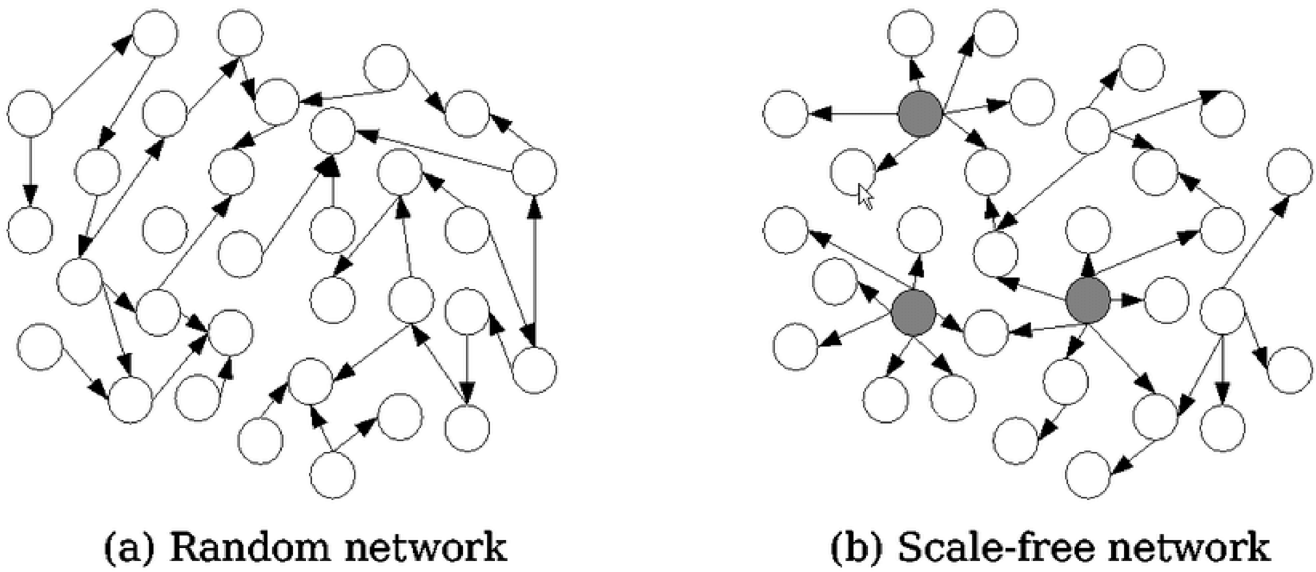

Figure 2.8: Comparison between (a) a random network and (b) a scale-free network (Svenson, 2006)

are called *hubs*. Hubs have a key role in such networks. For instance a big part of the circulation in the internet network go through those hubs and the removal of several hubs would lead to a major increasing of some path-length and could also lead to some disconnections between set of nodes in the network (Li et al., 2005).(Barabasi et al., 2003)

The most popular method for constructing scale-free networks have been given by Barabasi (1999b). The algorithm depends on two parameters: a number of node N, an initial number of node N_0 and a number $m \leq N_0$.

- Initially, a fully-connected network of size N_0 is constructed.
- Iteratively a new node is added to the network. m edges are added to the node. An edge is connected to an existing node with a probability $p = \frac{k_i}{\sum_j k_j}$.
- The algorithm ends when $N - N_0$ nodes have been added.

An example of network obtained by this algorithm can be seen on figure 2.8 . As previously explained, the main characteristics of scale-free graph is their degree distribution in power law. Ostroumova Prokhorenkova and Samosvat (2014) also showed that the clustering coefficient tends to zero if $1 < y < 2$. However scale-free networks can have high clustering coefficients for bigger value of y. The mean path length of this model increases in $\frac{\log N}{\log \log N}$ with the number of nodes N.

2.4.4 Comparison between type of networks

As previously explained social networks are almost never regular due to their mechanism of growth and evolution. However the topology of different social network can have different properties: They may be close from random, small-world or scale-free networks. Also the definition of small world and scale free networks are such that a network can present both properties. Networks may be classified in topologies using all the measures that we explained in section 2.3 and visualization tool. Figure 2.9 shows network of each type for visual comparison.

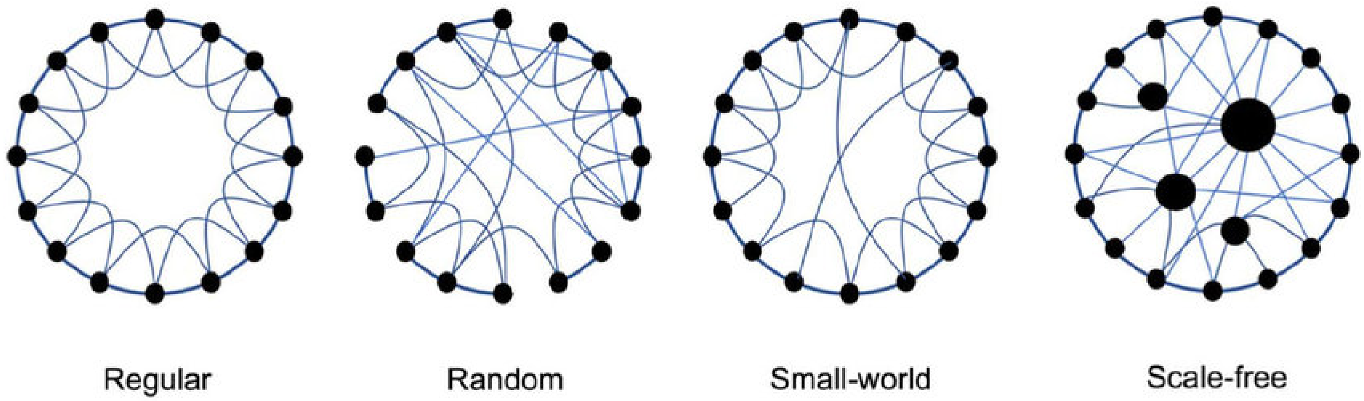

Figure 2.9: Comparison between different type of networks (Selvarajoo, 2019)

Chapter 3

Social network construction

In this part of this Master Thesis, I will briefly explain what are the main step executed by the software to extract social network and meta data from novels or script. The modification of the program that I've made will be more detailed. The bigger changes concerns the extraction of characters from novels. Also the original software was only analyzing novels, some adjustments have been made to extend it to movie scripts. I don't the state of the part for part who don't have been deeply modified. Every information about it are available at Waumans et al. (2015) and Nicodème (2015).

3.1 Assumption and conception of the social network

There is multiple way of constructing a social network from a text. Especially because there is multiple way of analyzing the structure of the narrative. Waumans et al. (2015) choose to see a narrative as a succession of event. The text is composed of a series of **conversation** that involve a set of characters. Each conversation can be subdivided into a succession of mini-event that are called dialog. In a **dialog** a person chosen in the set of character involved in the conversation will address a message to a the other characters from the set, the **audience**. The person that send the message is called the **speaker**. Also all sentences that are out of dialogs are called **context** and linked with the following each conversation or the one that is currently happening. Sentences of context can be located between dialogs from a same conversation. Characters that appear together in a conversation are linked in the social network with a weight that correspond to the number of conversation they are both in.

The narrative is also divided in **scene** that could contain multiple dialog. The scene represent additional cut where all the previous context should be thrown up. In novels scene are used to represent new chapter, in movie script scene are directly annotated.

3.2 Text analyzed

Most of the novels analyzed in this work have been provided by Waumans et al. (2015). They has been selected to represent a variety of popular series and some classic novels. They choose to select series to observe if social network from novels being part of a serie are sharing common properties. I've added to it the novel "Pride and Prejudice" to compare my result of character extraction with the results of Vala et al. (2015). Various script of movies have been added to compare social network developed in literary fiction and movies. There is both original scenario and adaption from books in order to measure difference between those type of movies. Adaptation of analyzed book have been added when they were available.

Novel	Script
Pride and Prejudice	
Harry Potter 1→7	Harry Potter 1-2-3-4-6-7
A song of Ice and Fire 1→5	Black Panther
The Lunar Chronicles 1→3	Thor Ragnarok
His Dark Materials 1→ 3	Blackkklansman
The Mortal Instruments 1→6	Boyhood
The Liveship Traders 1→3	Halloween
The Wheel of Time 1→14	Joker
Rain Wild Chronicles 1→4	Jurassic Parc 1→3
	Lord of the ring 1→3
	Shrek 1 and 3
	The devil wears prada
	Titanic
	Alien 1→3

Table 3.1

3.3 Implementation

Most of the NLP function used on this software come from the library pattern (De Smedt and Daelemans, 2012) only available on python 2. As new tools have been developed in python-3 library, the choice have been made to pass the code to python 3. In order to keep the structure of the code developed with the tool pattern, a python 3 branch of the library have been used. The branch is still in development and every function are not runnable but I manually fixed the needed function use it on the software. My fork of the library is available at *https://github.com/antheymans/pattern*.

To use new function that doesn't exist on the pattern library, the choice has been made to use the Spacy library (Honnibal and Montani, 2017). This library develloped by the MIT has the advantage of being well-documented. This is completed by the use of NLTK (Bird et al., 2009) which posses a wide variety of data and corpus for NLP. A lot of state of the art article use the standford CoreNLP package (Stanford NLP Group, nd) wich also haves all the state of the art tool. However this package is written in java. There exist a python adaptation of it but it is is poorly documented and doesn't include all the tool of the java version (Qi et al., 2018).

3.4 Formatting of input

In order to be used as input of the program, a narrative should be on the form of a text and transformed in order to follow a set of rules. The program have been made to exploit information available in books and movie scripts but a narrative following any other format can be used in the program if it follows the rules.

- The text should be stored in a *.txt* file with an *utf-8* encoding.
- All information located in the document should be part of the narrative. The title can be kept but any preface or thanks from the author will be considered as being part of the narrative.

- All dialogs should be delimited by double-quote. Double quote can not be used for other purpose. All double quote present in the text should be replaced with simple quote.
- If consecutive dialogs belong to the same conversation and have the same speaker, they should be concatenated. There is an exception to this rule is there is a context between those dialogs.
- A new scene should be represented by an empty line. No other empty lines are authorized.
- If the speaker of a dialog is annotated, it should be indicated in uppercase on the previous line. The name of the speaker should be preceded by "_" and followed by ":".
  For instance:
  _TINTIN:
  "Milou, go outside!"

The formatting is made manually using regex. A script have been created to transform automatically scripts having the most common structure but it often requires minor manual adjustment.

3.5 Preprocessing of the text

Initially all the sentences of the text are extracted, parsed and divided into chunks. Annotated speaker are extracted and linked with the related dialogs. Sentence belonging to a same dialog are grouped. The location of scene breaks is also stored in order to later cut the conversations at this points. The sentiment associated to each sentence is also computed using 2 scores: a measure of polarity that say how much the sentence is negative or positive and a measure of subjectivity that say how much the sentence tends to give personal opinion.

3.6 Construction of conversations

To group dialog in conversation, the program should be able to decide if 2 consecutive dalogs from a scene belong to a same conversation. The difficult part is that there is often sentences of context between dialogs of a conversation. The method develloped by Nicodème (2015) use the size of the spacing between dialogs to perform this task. It build conversations as a a list of dialogs having an associated list of non-dialog sentences. The second list is called the context of the conversation. The program begins by counting the number of spacings separating each dialogs. The spacing between dialogs belonging to different scene is not taken into account. From the number of occurrence of each value of spacing, a threshold value will be chosen. Dialogs separated by a spacing above this threshold value are not part of the same conversation. The threshold value is chosen using the following formula:

$$\begin{aligned} treshold = max\{spacing | frequency(spacing) > 100 \\ \vee (frequency(spacing) >= 10 \\ \wedge frequency(spacing) >= frequency(spacing + 1) \cdot 2 \\ \wedge frequency(j) > frequency(spacing) \forall j \in [1, spacing])\} \end{aligned} \tag{3.1}$$

The formula has been empirically chosen by Nicodème (2015).

A conversation end with the last dialog of the scene or with a dialog being separated by its successor by a spacing bigger than the threshold value. All sentences of context located in a conversation or between the current conversation and the previous one are grouped. This group is the context of the conversation.

3.7 Character identification

Character extraction is a major task of social network extraction. "Character identification consists in detecting which characters appear in the considered narrative, and when exactly they appear in this narrative" (Labatut and Bost, 2019). 2 sub-task can be considered as separated from each other:

1. The extraction of a list of characters.
2. The detection in the text of character appearance.

The first sub-task will be deeper analyzed on this work, as a new detection system have been developed to improve the result from Waumans et al. (2015) on novels. On this part, scripts and novels need different processing as scripts have annotated speaker for most of their dialog.

3.7.1 Character extraction in Novels

According to Labatut and Bost (2019), character in novels may appear on the form of a proper noun, a pronoun or an anaphoric noun phrase. Proper nouns that are composed of a single word are called proper names. In this work, character are only detected when they are on the form of a proper noun. This choice is motivated by the fact that characters will be later connected when they are mentioned in the same conversation. The cost of this simplification is the lost of smaller characters that appears only under the form of an anaphora. I considered that in most cases, a character taking part in a conversation is mentioned at least once under the form of a proper noun. In our situation, the first step of *character identification* on a written support is the extraction of names that represents the characters and the linking of aliases (names that refers to the same character). Once all extracted names are bound with a character, each occurrence of a name in the story signal the appearance of the associated character. As the task of extracting a set of proper nouns and binding aliases is error-prone, some authors decided to do it manually (Agarwal et al., 2013). He et al. (2013) proposes to build automatically a list of character from wikipedia but this method has the disadvantage to focus only on main characters and to be dependent on external information unavailable for some stories. Here we will focus on automatic method because they can be used on many different text with a minimal pre-processing.

The disadvantage of proper names over proper nouns

A major modification of the program concern this first step of the character identification process. The original process was only considering that a character could be represented by a proper name and was linking names that appears together more than 1 times over 3. But using only proper names over all proper nouns is a simplification that makes the program loose a lot of information. The use of proper nouns makes the binding of nouns more complex but also allow to use more information's to perform this binding. The current state of the art contains methods to perform this task.

chunk	names extracted using proper nouns	names extracted using prop
Ron and Hermionne	Ron AND Hermionne	Ron
you Mr. H. Potter	Mr. H. Potter	Potter
dear Harry Potter	Harry Potter	Harry OR Potter
James Potter	James Potter	James OR Potter
yer brother Charlie	Charlie	Charlie

The fact that proper names are only composed of a single token makes a perfect binding of names designating the same character impossible. The pairing is needed to link first names and last names of characters but in some cases multiple proper nouns that should not be linked appear in the same chunk. Here are some common mistakes in character extraction with example from the method of Waumans et al. (2015) used with the book "Harry Potter 1".

1. **When both the first name and the last name of a character are used to identify a character, they are not paired**: "Harry" and "Potter" are labeled as different character.

2. **When multiple characters share the same first name or last name, if the linking is made they will all be clustered as a single character**: the names referring the 8 characters of the "Weasley" family are mixed in 4 cluster. Each cluster is composed of aliases referring to 2, 3 or 4 characters.

3. **Some characters are referred with proper nouns that should not be used for pairing**: "Mr" is paired with "Dursley" and "Weasley" because the text often refer to "Mr Dursley" or "Mr Weasley". It causes the clustering of unrelated words. The same problem appear with other title like "professor".

4. **Some chunks contain multiple characters**: The algorithm link to a single character the chunks "Ron and Hermionne", "Mr and Mrs Weasley" or "Fred and George".

Name Entity Recognition

The task of labeling group of words as Entity is called Name Entity Recognition (NER). Those entity includes *person*, *location* and *organisation* (Gudivada and Arbabifard, 2018). This is the most common way to extract characters names from a novel. The best NER methods are supervised or semi-supervised and trained with annotated datasets of news, text from social media or biomedical data (Yadav and Bethard, 2018). This decreases their performance on novels, especially novels from fantasy or the older ones (Dekker et al., 2019). Unsupervised methods typically rely on rules and domain-based knowledge, it makes them completely domain dependent. POS-tagger may be considered as naive program of NER. The original software (Waumans et al., 2015) was extracting all words labeled as proper names by a POS-tagger with the major issue of considering only single-word names while in Elson and McKeown (2010), proper names are tagged using a POS-tagger and contiguous proper names are considered as a proper noun. Vala et al. (2015) also extracts subjects of verbs present in a dataset of verbs strongly associated with “person” entity. This techniques allows to detect anaphoric nouns and not only proper nouns. Coll Ardanuy and Sporleder (2014) store the number of times that words are classified at location or person, so words that are likely to be location are removed from the location list.

Pattern the NLP library of python that is used in the software doesn't have any NER module. Spacy, another NLP-library of python have an available module that performs NER using ‘Conditionnal Random Fields’, a statistical model that uses supervised learning. Pre-trained model are available but the model could also be trained manually with an annotated dataset. The most common tool for NER is the Stanford Named Entity Recognizer (NER) wich is only available on java. For reason of practicability, it has not been considered in this work.

Unification of Character Occurrences

Unification of Character Occurrences is the task of unifying all mentions of a same character in a narrative (Labatut and Bost, 2019). When only proper nouns are considered, this task can be simplified into Alias Resolution: the linking of all names that refers to a same characters and the making a differentiation between names that refers to different characters (Scott and Carrington, 2011).
The binding of names can be done using a measure of string similarity, set of rules or using meta-information of strings such as an inferred gender. In multi-stage methods, proper nouns are are divided between cluster, each of them being associated to a single character. The clusters are merged following a sequence of conditions. Multiple methods have been developed to solve this task with their own specificity without that one method stand out from others.
The original software (Waumans et al., 2015) was linking proper names that appears together 1/3 of the times to link first name with last name with the major inconvenience of binding different characters sharing a same first name or last name and leaving unbound name and lastname of characters that are more frequently called by only one of those words. However most of the state of the art methods focuss on proper nouns: Elsner (2012) discards all proper nouns that appears less than 5 times, then try to bind multi-words nouns between them before binding them with single words nouns. Coll Ardanuy and Sporleder (2014) also classify names into multiple classes and apply different rules on those classes. Vala et al. (2015) propose to bind nouns that share words except in some cases, such as nouns sharing a last name but having a different first name. Davis et al. (2003) proposed a method to bind entity representing art object by generating variations of proper nouns and linking them with names corresponding to those variations, this method have been applied on characters detection in novels by Elson and McKeown (2010); Vala et al. (2015). In Elsner (2012); Coll Ardanuy and Sporleder (2014); Elson and McKeown (2010), a gender is inferred from each proper nouns using a list of masculine and feminine first names and gendered titles to avoid to merge cluster of names having different genders.Coll Ardanuy and Sporleder (2014) a sliding window is also used on the text to detect pronoun near nouns and use it to infer the noun gender. It ends by removing infrequent characters, characters whose the total number of apparition of the cluster of names is smaller that a threshold. This should reduce the number of "false positive" characters (cluster of names that are not related with a character), at the cost of minor character. Elsner (2012) draws the conclusion that all methods are error-prone and that the difficulty to obtain annotated data on this task makes any comparison between the different methods very difficult. Even with a dataset containing all characters present in a narrative, the presence of multiple aliases would makes the evaluation of the character extraction very difficult.

Some methods also use co-reference resolution tools to link characters mentions between each others (Vala et al., 2015). Coreference resolution tools are program that automatically clusters mention in text that refers to the same entity. They do it using neural networks and are especially used to link names with pronouns or anaphoric noun phrase (Wiseman et al., 2016; Martschat and Strube, 2015) . This is useless to the software here as those form of references are not used in this work.

3.7.2 Proposed method of character extraction in novels

As explained in the previous section, the method here extract proper nouns, the extracted nouns are composed of multiple words including honorifics. Also only proper nouns are extracted, anaphoric noun phrase are ignored. The method consist of a multi-pass algorithm that passes 2 times over chunks to extract a list of nouns, classify them and then pass 5 times over the set of nouns to bind aliases.

Proposed system of Names Entity Recognition

To find a NER method for characters extraction in novels, we relied on some hypothesis. Names from fiction have the particularity to be invented in order to be easily recognizable by the reader. We can rely on the hypothesis that in most novels, there is no namesake person. An exception to this hypothesis is the characters "Barty Croupton" in Harry Potter, but the younger is called "Barty Croupton Junior" to avoid any confusion from the reader. Also we consider that characters from novels should be labeled as person, so we are only looking at the NER system capacity of distinguish person from other entities. We will later accept some entities as "speaking object" as characters. But it has been considered that most of them have specific nouns that are likely to make a NER system to label them as character and the remaining ones should be minor in the narrative. So if we miss one of them, it's not considered as a major issues

As explained in the state of the art, a python library, Spacy, provide a tool for the task of name entity recognition. However this tool has not been trained specifically with novels nor for the task of extracting characters names from a text. It doesn't detect honorifics and works as a "black box" which gives sometimes incorrect result. For instance in the book "Harry Potter 1" the tool detect has characters: 'Harry bellowed': 1, 'Harry anxiously': 1, 'knew yeh didn',' baker', 'yer meddlin','Happy Birthday Harry'. It seems that interjection such as "yer" or "yeh" are interpreted as nouns and a lot of verbs or adjectives are seen as part of nouns like with "Harry bellowed". Also the tool doesn't make a difference between proper nouns and anaphoric noun such as with "baker". This is an issue as anaphoric noun phrase loose their meaning out of their context and considering them as character names could lead to major confusion. For instance the anaphoric noun phase "the father" could represent a lot of characters according to the context. For al those reasons it has been chosen to avoid to work with this library.

A Name Entity Recognition method has been designed to extract characters names from novels. This technique uses a POS-Tagger to extract proper names from the text and several collection of proper names commonly used in English to isolate characters names. Then proper nouns are extracted from chunks containing proper names. The collection of words contain a list of countries, nationalities, honorifics (classified following the related gender), stop-words, profanities, words related to time and words related to the academic domain. In this method, a proper name is considered as *valid* if it begins with a capital letter but is not composed with capital letters only. The assumption has been made that proper names contains more than one letter.
First of all the POS-tagger is passed on the head word of all chunks to extract proper names. The POS-tagger used is the one developed by the python library pattern which classify proper names whether they are related to a location or not. If a word is not labeled as a location, the algorithm considers the word as a proper name if it doesn't appear in the lists of English proper names and is not the first word of a sentence. First words of sentences are not kept as they are capitalized and so they are more likely to be wrongly classified as proper names. The assumption is made that proper names will appear several time and that at least one of those appearance will not be at the begin of a sentence. Proper names and proper names related to locations are saved with the number of their appearance.
Secondly the algorithm check that words labeled as location don't appear in the list of proper names. In such cases, the word will be removed from the list of proper names if it appeared more often as a location than as a name.
The final step of the Names Entity Recognition method consists of extracting proper nouns from proper names. All chunks containing a proper name are isolated. Then all sequence of consecutive valid proper names are extracted and considered as proper nouns. For instance from the chunk "Mr Harry Potter and Ron" 2 proper names are extracted: "Mr Harry Potter" and "Ron". The proper nouns are registered with the number of their appearance.

In this method, many restrictive conditions are putted on proper names extraction. But the way names are extracted implies that if a name is extracted one time, he will be considered each time as a proper name. For this reason the addition of new names should be very cautious. The hypothesis is made here that a frequent character's name will appear at least once in a situation that allows to univoquely identify him as a character's name. On the original program, only few conditions are observed and result shows that most of the extracted characters don't refer to actual character of the novel. This observation will be explained more formally in the section 3.7.2

Proper nouns classification

All the extracted proper nouns are classified following their gender and their form to make easier the alias resolution. The classification of genders is also used to produce gender-related information about the topology of the network.
Firstly all the names are parsed using the python tool *HumanName* from the library NameParser. It separates words contained in a noun in a list of honorifics, first name and last name.

To infer the gender, the algorithm will used the list of honorifics classified by gender and a list of gender related first name. The gender of nouns containing a first name from the list of first name or an honorific from the list of labeled honorifics will be assigned. If multiple words in the noun are related to different genders, the majority vote is used. After this phase all nouns are labeled as *masculine*, *feminine* or neutral.

Then a category will be assigned to each words following their forms:

1. The first category contains nouns that have at least an honorific, a first name and a last name.
2. The second one contains nouns with a first name and a last name but no honorific.
3. The third category contains nouns with an honorific and a first name but no last name.
4. This category contains nouns with an honorific and a last name but no first name.
5. The fifth category contains the remaining nouns.
6. If the first name of a noun is not in the list of proper noun and doesn't respect the criteria to be labeled as proper noun, the noun is considered as an error and is removed.

Alias resolution

The alias resolution process consist of a multi-pass algorithm that firstly creates primary link and secondary link between all nouns, then transforms secondary link into primary link in some cases. After this linking phase, cluster are build to represent character. Each cluster consists of nouns connected between them by a path of primary link. The most used noun of the cluster is called the head of the cluster and is used to represent the associated character in the social network. In the following explanations, I will speak about "compatible genders" between 2 nouns, it means that the gender assigned to those nouns doesn't prevent them to point the same character. Concretely they have the same gender or one of them is neutral. To be connected, nouns have to have compatible genders. Once two nouns are linked by a primary link, if one of them was gender neutral, its gender will be inferred from its neighbor. In such situations the neutral word will delete all its connections with words of the opposite gender. In the following explanation I will also speack about "primary-connection" or secondary"neighbors"to speak about names connected using a path of primary link and names neighbors from each other by a secondary link.

During the first pass, all first name used by nouns of class ¡3 are taken one by one. Diminutive and nicknames associated to a first name are looked in the set of proper names. Then all proper nouns that possess that first name, a diminutive or a nickname are grouped. In some cases the first name, the diminutive or the nickname has been labeled as last name in other nouns, those nouns are also added. For each pair of nouns from this group that have compatible genders, the nouns are linked

- using a primary link if:
  - they have the same last name and their first name is the original first name, a diminutive or a related nickname. This also works with nouns that don't have last name as they last name is considered to be the empty string. For instance: "D Dursley" and "Dudley Dursley".
  - if both of the names have an empty last name or first name. In this case the algorithm consider that they share a same first name/last name even if it is possibly labeled differently in each noun. For instance:"Mr Flitwick" and "Professor Flitwick".

  Once 2 nouns are connected with a primary link, if one of them was neutral, he will receive the gender of the other word.

- using an in primary link if:
- their first name is the original first name, a diminutive or a related nickname and only one of them have an empty last name.
- if one of them have an empty first name. Secondary link represent a potential connection between names which could be unrelated. For instance "Mr Durlsey" could be related to "Dudley Dursley" or to "Vernon Dursley", so he will be linked using a secondary link to both of them. "Harry Potter" and "James Potter" should not be linked as it appear clearly that this is 2 different character sharing a last name.

In a second phase, all last name are taken one by one. Last name that has also been used in previous pass as a first name are not considered because they are likely to be incorrectly labeled. All nouns sharing the last name or a diminutive are grouped. For each pair in this group, if they have compatible genders, they are linked using:

- a primary link if both of their firstname are empty.
- an in primary link in other cases.

Once these phase have been completed, we will observe secondary link and separate those who are due to a true connection between 2 names and the links that should be removed. We will takes the nouns that contains the less information and try to link them with the nouns that are more likely to be their neighbors. The chosen neighbors is the most common noun among the neighbors. This technique will result in some errors but their numbers should be limited.
The third pass focuses on neutral nouns of category 1 and 2. We will try to find them a gendered alias among their potential partners. Firstly each of them will be grouped with all nouns connected to him using primary links. Then a second group will be formed with all they secondary-neighbors. Iteratively the most common neighbors will be removed from the second group and linked to the first group using a primary connection. If the noun is gendered, the gender will be assigned to all word from the first group and the process will end. If the noun is also neutral, all nouns primary

ly connected to it will be added to the first group and again the most common noun from the second group will be chosen to be linked with the first group.
The fourth and fifth pass focus on words from the 3-4-5 categories that are not connected with nouns from the 1-2 categories. Those words consists of a single first name or last name and should be linked to the most common character having this name. Again in the forth pass they are grouped with the primary-connected nouns. The most common noun among their secondary neighbors of 1-2 categories is added to the group and primary ly linked with it. But nouns that are not linked to any 1-2 nouns will remain unbound. In the fifth pass, all secondary links between those names will be iteratively transformed into primary link. Link between nouns that becomes incompatible due to their gender during the process, are removed.
In the last part of the algorithm, cluster are made with words primary-connected and the most common noun of each cluster is chosen as head of the cluster.

Performance

As explain in the state of the art, there is no straightforward way to measure the performance of the character extraction techniques. Firstly the performance of the NER and the alias resolution system are not independent: For instance a NER method that add redundant adjectives to nouns will only decrease the performance of the overall social network if the alias resolution system is not able to link all those nouns. Secondly most method from the state of the art don't have open-access implementations. Thirdly there exist almost no annotated data that would allow to automatically score different method. The goal of the algorithm developed here is to find different aliases referring to each character. But there exist multiple way of doing it that are not strictly better from each others. The name "Weasley" for instance could be associated accurately to multiple characters in "Harry Potter" while associating it with the noun "Malefoy" is clearly a mistake.

For those reasons, the method developed in this master thesis will only be compared with the previous method used in the software (Waumans et al., 2015) and with a method of the state of the art, Vala et al. (2015), whose authors provide the result of their character extraction for the novel "Pride and Prejudice". I will manually count the number of clusters headword that refer to an actual character to count the number of false positive of character detection. I will also compare the set of main characters detected by our program and the set of main characters detected by Vala et al. (2015) and manually check which errors appears. This comparison is not possible with the original software as the program didn't classified the cluster of characters names by number of appearance.

To count the number of false character, I've decided to consider only the most common name of a list of aliases. So a character is considered as false is the most common name is false even if other aliases are valid to designate a character. The name should be part of the canonical name of a character or designate it unambiguously. In the novel "Harry Potter" appears a lot of magic object or animals. Animals and speaking object are considered as character as they are potentially part of a conversation. For instance the word "Hat" (capitalized) is considered as designating univoquely a character as it can only refers to a speaking object "the Magic Hat" but the words "Mommy" or "Chaser" could refer to a lot of characters according to context, they are considered as an error. Word related to name, like "Potter" are still considered as valid even if they are not univoque. This measure don't give indication of how well alias are bound bind the algorithm but it allow to measure if the algorithm is able to isolate names referring to character in the text.

The result shows that the proposed algorithm is far better than Waumans et al. (2015) at the task of recognizing characters entities in a text of novel, for the 3 tested novels, the rate of "true

Table 3.2: Number of detected character that refers to actual characters(true positive).

book	Porposed method	Waumans et al. (2015)	Vala et al. (2015)
harry potter 1	65% (70/108)	32% (61/193)	//
The Lunar Cycle	67% (24/36)	26% (25/96)	//
Pride and prejudice	89% (44/56)	46% (53/116)	85%(61/72)

chatacters" is each time at least two times bigger. We also see that the result variate a lot from one novel to another. This could be explained by the fact that some novels as "Harry Potter" develop a large universe and invent a lot of proper names to describe it. Those unknown proper words are particularly difficult to dealt with for the extractor. A program using more syntactic information could may be decrease the importance of this issue. The presented algorithm obtain result very near from Vala et al. (2015) however the type of errors that appears on the novel ''Pride and Prejudice" differs. In Vala et al. (2015) most of the errors comes from anaphoric noun phrase such as *servant*, *housekeeper*, *owner* or *assistant*. In our program those element should not be extracted as they are not univoquely linked to a character and depend of their context but we could imagine that an other program that constructs a social network in a context-dependent way could use those information. Most of the errors that appears in our own program are nouns of location wrongly considered as character: *Netherfield Park*, *Pemberley House*, *Brighton*, *North*. This different of topologies between errors of the 2 considered method could be due to the fact that we develop our own "Name Entity Recognition" program adapted to characters recognition that is mainly focus on the form of words wich allows to separate anaphoric noun phrase from proper nouns but can not use syntactic information to separate name of characters from uncommon location. The "location" tag of the TAG-Poser is used as a list of countries but it only allows to detect the most common location. On the other hand Coll Ardanuy and Sporleder (2014) used a common Name Entity Recognition tool that should be able to label nouns as person our location. For future works, it would be interesting to combine rules used by this method with a standard NER tool that would separate characters names from locations.

The second test that we will make here will be to compare the main clusters of nouns that represent the characters for the presented method and Vala et al. (2015). This test has the disadvantage of being manual and qualitative more and not automated and quantitative. But not metrics exist to measure the ability for an alias resolution system to connect the related nouns between them.

In the character extraction of Pride and Prejudice, a first difficulty that appears is the extraction of the Bennet family. In this family there is 5 daughters named in the text and a mother only referred as Mrs. Bennet. In the presented method, the character of Mrs.Bennet have been mixed with "Elizabeth Bennet", the most common character among the sisters. About that last characters, both method have been able to link it with her nicknames "Eliza" and "Lizzy" . In the Bennet family both method isolate correctly the sisters "Jane"' and "Lydia" but Vala et al. (2015) didn't record the alias "Lydia Bennet". Our method have a second error, the character " Mr. Fitzwilliam Darcy " is separated in 2 characters: "Mr Darcy" and "Mr Fitzwilliam" that don't appear on the figure as it is less used. But the character "Charlotte Lucas" is correctly extracted while Vala et al. (2015) mix it with an other character'Mary King'. They also mix the character "William Goulding" with the character "William Lucas". We see that they have considered 2 anaphoric noun phrase as character: "servant" and "housekeeper" while our method have wrongly considered 2 location as character "Permberley House" and "Netherfield Park". Our method bind family names such as "'The Gardiners" with the most common character of the family while the other method avoid to use them.

False Positive	Appearance	Gender	Aliases
0	920	-1	’Elizabeth, Lizzy, Eliza, Miss Elizabeth, Miss Elizabeth Bennet, Bennets, Mrs. Bennet, Mrs. Bennet ’–My Miss Bennets, Miss Bennet, Elizabeth Bennet, Miss Eliza Bennet, All Elizabeth, Miss Eliza, Miss Lizzy
0	359	1	’Mr. Darcy, Mr. Darcy:–but, Mr. Darcy–that Mr. Darcy, ”, Darcy
0	273	-1	’Jane, Miss Jane Bennet
0	212	1	’Mr. Bingley, Bingley
0	188	-1	’Lady Catherine, Kitty, Right Honourable Lady Catherine, Catherine
0	180	1	’Wickham, Mr. Wickham, Mr. Wickham–when, George Wickham, George
0	158	-1	’Lydia, Miss Lydia Bennet, Miss Lydia,
0	148	1	’Mr. Collins, The Collinses, Collinses,
0	120	-1	’Charlotte, Lady Lucas, Miss Lucas, Miss Lucases, Charlotte Lucas
1	114	1	’Netherfield, Netherfield Park, Rosings Park, Park, Rosings, The Netherfield
0	106	-1	’Miss Bingley, Mrs. Bingley, Caroline Bingley,Bingleys, Caroline
0	94	-1	’Longbourn, Mrs. Long, Long,The Longbourn
0	86	1	’Mr. Bennet, The Bennets, Bennet
0	57	-1	’Mrs. Gardiner
0	56	1	’Sir William, Sir William Lucas, Lucas,Lucases, The Lucases
0	56	-1	’Mary, Maria, Maria Lucas,
1	54	1	’Pemberley, Pemberley House, Longbourn House,
0	54	-1	’Miss Darcy, Mrs. Darcy, Georgiana Darcy,Georgiana
0	53	0	’Meryton, All Meryton
0	40	1	’Mr. Gardiner, Gardiner, Gardiners, The Gardiners

Table 3.3: 20 main character detected in “Pride and Prejudice” by the presented method. The first column tell if the character refers to a true character from the novel and was used in the previous measurement. The second column contain the total number of appearance of names related to the character and the third one contain a 1 if the infered gender is masculine, a 0 if it is neutral and a -1 if it is feminine. The last column contain all aliases of the character.

False positive	Appearance	Aliases
0	752	Eliza, Elizabeth, Lizzy, Miss Eliza, Miss Eliza Bennet, Miss Elizabeth, Miss Elizabeth Bennet, Miss Lizzy
0	313	Bennet, Mr. Bennet
0	309	Colonel Fitzwilliam, Fitzwilliam, Mr. Darcy, Mr. Fitzwilliam Darcy
0	291	Jane, Miss Jane Bennet
0	217	Catherine, Honourable Lady Catherine, Kitty, Lady Catherine, Lady Catherine de Bourgh, Miss de Bourgh
0	150	Mr. Collins
0	115	Mr. Bingley
0	107	Lady Lucas, Maria, Maria Lucas, Mary, Mary King, Miss King, Miss Lucas
0	73	Mr. Wickham
0	58	Mrs. Gardiner
0	45	Sir William, Sir William Lucas, William Goulding
0	39	Colonel Forster, Forster
0	36	Gardiner, Mr. Gardiner
0	34	Lydia
0	29	Mrs. Collins
0	21	Mrs. Hurst
0	20	Mrs. Phillips
1	18	servant
0	16	Caroline
1	16	housekeeper

Table 3.4: 20 main characters detected in “Pride and Prejudice” by Vala et al. (2015). The first column tell if the character refers to a true character from the novel and was used in the previous measurement. The second column contain the total number of appearance of names related to the character. The last column contain all aliases of the character.

Speaker	Issue
˙BEAUREGARD - KLAN NARRATOR (O.S.)	The name is given with an anaphora
˙JEROME TURNER (V.O.)(CONT'D)	additional information are given with the annotation
˙RON (CONT'D) / ˙RON STALLWORTH	This is 2 different annotation that refers to the same character
˙RON STALLWORTH, FLIP AND JIMMY	An annotation can refer to multiple character
˙CSPD OFFICER BRICKHOUSE	Character can be represented by their function and not only by their name
˙AMERICAN TERRORISTS	Groups are also considered as speaker in some situation

Table 3.5: Speaker annotated in the script of the movie Blackkklansman

The observation of the characters extracted by both method shows some tendencies. Our method tend to extract more aliases but also extract some incorrect ones. It also bind set of aliases more cautiously with result in dividing a character in two set of aliases but avoid to mix between them completely different characters. The only case of mixed characters in our method comes from characters of a same family that differs only by a title, both associated to the same gender. As previously shown, the proposed method ignore anaphoric noun phrase but is likely to consider locations as characters. This analysis don't allow to consider one of the method to be really better than the other. From this we can consider that our method is approximately at the level of the state of the art.

A major default that should be pointed of the proposed method is the use of a words dataset for honorific detection. Honorifics are only properly labeled if they are present in the word list. Therefore, novels using set of foreign honorifics or inventing their own set of honorifics (which is common in fantasy or science fiction) could completely miss them. Honorifics would be considered as first name and it could decrease the performance of the alias-resolution process. Also the gender of characters would be assigned in less cases.

Further improvement

The method developed here for character extraction in novels gives satisfying result but could have several improvements. Firstly a NER tool should be used in order to use syntactical information to separate location from characters names. Secondly some rules could be added to link aliases between them for instance in the case of characters having multiple last names. Then a way of using anaphoric noun phrase in the system could be used in order to detect unnamed characters but this addition should be handle carefully as the addition of anaphoric noun phrase could lead to the extraction of many nouns that are not univoquely linked to a character. Thirdly, it would be interesting to extend the set of recognized honorifics or to implement a way to detect new honorifics in a novel.

3.7.3 Character extraction in scripts

As previously explained, in all scripts, the speaker is directly annotated for most of the dialogs. The list of all annotated speaker is a good approximation of the character list. However it causes a list of issue that I will detail here under. Example annotation are given on table 3.5 to illustrate issues caused by using those annotations.

- The observation of scripts shows that those annotation mostly use canonical characters names (with the first and last name for the character). But sometimes character are only referred with their first or last name. So a minimal alias resolution system is needed.

- As the name of characters used for annotation are mostly canonical name, the list will miss aliases of the characters. Especially nicknames and familiar names used by characters that are very intimate with each others.
- If a character don't talk during a conversation and is only mentioned by other characters using unknown aliases, he will not be detected as taking part of the conversation. The network may miss some relations.
- If a character character is part of the audience in some conversation but never talk, he will not be added in the social network. We may loose some minor character.
- Information could be difficult to extract from annotations. Multiple characters or extra information could be written and everything is uppercase which makes difficult to extract names. A system filtering the information is needed.
- As script are not perfectly formatted, the automatic labelisation of speaker and dialogs using regex may lead to errors. I've manually corrected some of those but I can't look all the text for them. This causes the incorrect extraction of words as characters.

In order to use annotations while diminishing the number of related errors, a method has been developed. The method is inspired from the previous method for novels with many simplifications. The method is develloped here under:

1. In pre-processing: When a parenthesis is detected, we assume that the end of the sentence is an additional information which is not kept. All words are capitalized. The sentence is spitted in multiple speaker when "And", "/" or ",“ are detected. Useless characters such as ";" or "." are removed.
2. At the begin of character extraction, all speakers are registered as proper nouns.
3. Each of the proper names that composed the speakers are also registered as proper nouns and linked with the corresponding speaker using a secondary link.
4. Speaker are also splitted when a "-" is detected. "-" are used in scripts to gives multiple name to a same speaker, for instance: "˙BEAUREGARD - KLAN NARRATOR “. All names generated from the split are considered as proper nouns and link with the corresponding speaker using a primary link.
5. The genders of all nouns are infered using the technique explained in section 3.7.2.
6. For all extracted nouns that have been at least one time extracted after a split (they have been part of a bigger speaker in one occurrence), the neighbor having the most appearance is chosen. If the neighbor have a compatible gender, he is linked with a primary linked and their gender is equalized following the technique of section 3.7.2.
7. Cluster are made of nouns connected by primary links. The most common of the nouns is the head of the cluster. Each cluster represent a character and its aliases.

The method allow to increase the number of detected aliases, link aliases between them and infer gender from names. It also allow to filter useless information from speaker and split multiple speakers when their appears. As speakers are not always properly chosen in pre-processing, we may still have some errors. There is also still the problem with non-speaking character. However those characters are minor and we consider that it is not an important issue. On the movie "Blackkklansman", no errors are detected. All the extracted characters are really speakers from

Cluster mentions	Gender	Main Name	Other names			
312	1	Ron Stallworth	Stallworth	Ron		
158	0	Flip				
120	1	Felix				
78	1	Patrice				
58	0	Chief Bridges	Bridges	Chief		
57	1	Walter	Walter Breachway	Breachway		
57	1	Devin Davis	Davis	Devin		
29	1	Connie				
29	1	Walker				
25	0	Ivanhoe				
23	0	Sgt Trapp	Trapp	Sgt		
20	0	Landers				
20	0	Kwame Ture	Ture	Kwame		
19	1	Beauregard - Klan Narrator	Klan Narrator	Narrator	Klan	Beauregard
19	1	Jimmy				
13	1	Jerome Turner	Turner	Jerome		
11	1	Mr Turrentine	Turrentine	Mr		
10	0	Agent Y	Agent			
5	0	Officer Mulaney	Mulaney	Officer		
5	0	Pleasant Man	Man	Pleasant		
5	1	Hakeem				
3	0	Wheaton				
3	1	Butch				
3	1	Jesse				
3	0	Cspd Officer Brickhouse	Brickhouse	Cspd		
3	0	Cspd Officer Myers	Myers			
2	0	Black Mass	Mass	Black		
1	0	Officer Cincer	Cincer			
1	0	Pre - Recorded Message	Recorded Message	Message	Recorded	Pre
1	1	Ron Stallworth				
1	0	All				
1	0	Voice				
1	0	Klansmen				
1	-1	Odetta				
1	1	Josh				
1	0	American Terrorists	Terrorists	American		

Table 3.6: Extracted characters for the script of the movie Blackkklansman. The first column contain the total number of times each alias of the character have been recorded as speaker. The second one contain the gender of the character. Then comes all aliases of the name.

the story and their aliases are correctly bound. However some of the speakers are not human being: “Pre - Recorded Message” refers to a sentence given by the messagery of a phone, “All” refers to a sentence said by a crowd. I remind that in section 3.7.2, I’ve defined speaking object as potential character. The list of extracted characters of “blackklansman” is given on figure 3.6.

An other solution would have been to use the character extractor developed for novels on script. The table 3.2 shows that this system is doing many errors. Using this system on script could lead to even more errors as script have a less regular format than novels. This difference could be explained by the fact that scripts are not commercial product, they are not salable so error in the format are not corrected while the script is understandable for the users. In particular we found more uppercase words that makes the extractions of proper nouns more difficult. However the use of techniques associated to novels on scripts could improve the result for further advancement when character extractor will decrease this number of false positive, as techniques associated to novels use more information than just the annotation.

3.7.4 Identification of character appearance

This step consists in binding each occurrence of character mention to the corresponding character and to identify speakers and listeners in each dialog. The algorithm have been mainly designed by Nicodème (2015) but some adaptations have been made. The previous algorithm was using proper names and chunks associated to this proper names. The new version use proper nouns only.

Detection of potential speaker and listener

The first phase of this task is the identification of potential speaker and listener in dialog. This don’t require the use of the character list previously extracted. It can also be considered as a part of the conversation construction.

The program already parsed all sentences and separate dialogs from context. The parsing include the tagging of words by a POS-tagger. All group of words labeled as subject by the POS-tagger are considered as potential speaker. Words labeled as object of the sentence are potential listener. Those group of words are mainly pronouns and anaphoric noun phrase, that are useless for the social network construction. For this reason, proper nouns will be extracted from those group of words. If no proper noun can be extracted, the subject or object is not kept. The extraction follows the step explained section 3.7.2. The initial software was keeping chunks but the system has been modified to be compatible with the new way of extracting characters.

Filtering of potential speaker and listener

For each conversation, each character detected as speaker or listener need to be associated to a character extracted in the text. I remind that potential speaker and listener are proper nouns. To look for a character, the following step are followed:

1. If the noun correspond to an alias, they are linked.
2. If it is not directly related to an alias, the nouns is separated into its different words. We link the noun with each alias that correspond to one of those words.

3. At the end of the previous steps, the noun is linked with 0, 1 or multiple aliases. The characters corresponding to each aliases will be looked up and we consider that each of them made an appearance at this sentence.

Identification of character in a conversation

The program loads separately each conversation and look for characters appearance into it. Character are always separated into speaker and listener.

- Firstly all dialogs of the conversation are loaded and the program extract characters from the potential speaker and listener and filter them using the procedure given in section 3.7.4.
- If no speaker is found, the program will look for sentences of the context located around the dialog and try to extract names from it. In some situations the speaker is mentioned just before or after the dialog, such that in the following sentence of Harry Potter 1: "Oh, yes," said Mr. Dursley, his heart sinking horribly. "Yes, I quite agree.". In this sentence the speaker, "Mr Dursley", is mentioned just between the line of dialogs. If no speaker are found, the program will also look in the context associated to the previous and next dialogs.
- After this step, the number of appearance of each speaker in the conversation is counted. If a dialog have multiple speaker, only the speaker that have the most appearance in the conversation is kept.
- In the last step, if the speaker of some dialogs stay unknown, we will try to infer the speaker from data of other dialogs.
  1. If there is only one speaker in the conversation, we assume this is a monologue and all dialogs are assigned to this speaker.
  2. If there is 2 speakers, we assume that 2 characters are talking to each others. Firstly, the program will look for the speaker of the last dialog and assigned the other speaker to the dialog. If the conversation is beginning or the last dialog don't have an identified speaker, the program will look for the next dialog. Again, if a speaker is found, the other speaker is assigned to the dialog.
  3. If there is more than 2 speakers, the program will look for the speakers of the previous and next dialogs. Then it will assign to the dialog the most common speaker from the conversation among all detected speakers except the speakers previously identified.
- Again, the number of appearance of each speaker in the conversation is counted. If a dialog have multiple speaker, only the speaker that have the most appearance in the conversation is kept.
- Each sentence of context is linked with the speaker of the nearest dialog.
- If a dialog have no identified listener, the program consider that all characters appearing in the conversation are listener, except the speaker of the dialog.

This method has the disadvantage of giving more appearance of most common speaker in a conversation. It also tend to add in conversation character that are mentioned without being part of it. For example if in Harry Potter, Harry told to Ron "Don't say that to Hermionne", she could wrongly be considered as the audience of the sentence. However in such case, the social network that will use this information would still be meaningful, as Harry speaking about Hermionne to Ron, implies a relationship between those 3 characters.

Filtering of useless character

All characters that don't have been identified a single time among all conversations are removed from the list of character. They will not be part of the social network.

3.8 Construction of the network

The social network is constructed using the identification of characters in dialogs in section 3.7.4. The implementations is made using the python library *networkx* (NetworkX developer team, 2014). Two types of network are constructed:

1. context networks: social network that are relative to a single conversation and the associated context.

2. incremental networks: networks which are the result of the incremental addition of context networks. They represent the link between characters from a set of conversation.

3.8.1 context networks

Context network are a particular type of network developed by Waumans et al. (2015). They have the particularity of representing only a conversation, so, all characters of those networks tend to be linked between them. They have been designed only to make possible the construction of incremental network.

They are builded by executing the following steps:

1. A conversation and the related context are isolated.

2. For all dialogs having an identified speaker and audience, a link is added between the speaker and each of the members of the audience. The link is weighted following the number of dialog that links the couple of characters. This weight initialized at 1. There is also a second weight representing the sentiment between the characters. It is initialized with the value of the polarity detected in the line of dialog.

3. If there is already a link between a speaker and a listener, the first weight is incremented. The polarity of the dialog is added to the sentiment-weight.

4. At the end of the network construction, the sentiment weight of each link is normalized by dividing it by the first weight of the link.

3.8.2 incremental networks

The name "incremental network" come from Waumans et al. (2015) in opposition from the context networks. They are constructed by incrementally adding context network in a chronological order. There is as much incremental networks as there is conversations in a novel or a script. The objective of constructing such graphs is to observe how the social network evolve with the story. The final incremental networks represent all the relations in the narrative. It can be considered as a standard social network extracted from fiction that have been studied in many article such as Agarwal et al. (2013); Dekker et al. (2019); Elson et al. (2010); Quang Dieu and Jung (2015); Labatut and Bost (2019); Dekker et al. (2018).

$$sentiment^{n+1}_{incremental} = \frac{weight - 1}{weight} \cdot Time - Factor * sentiment^{n}_{incremental} + sentiment_{context} \quad (3.2)$$

To build incremental networks, the context graphs are merges as follow: All new nodes of the new context graph are added to the incremental networks. All links are added by conserving the sentiment of the node and setting a weight at 1. This weight represent the number of conversation where the link appear and should not be confused with the weight of the context network. If a link was already existent in the incremental networks, the weight is increased by one. The sentiment score is modified by using equation 3.2 with $Time - Factor = 1$. The goal of this equation is to slightly decrease importance of previous sentiment score and see how the sentiment between characters evolve during the narrative. At the end of the addition of links, all node that represent character that don't appear in enough conversation are removed. The goal of this removal is to throw all false characters and minor character that are not important for the social network. The threshold value have been kept at 1 to keep minor character as the new character extraction system have already removed a lot of false characters.

3.9 Data-extracted

Various data are extracted from the document and can be used for further analysis. This data include network, information about the characters and dialogs or meta-data used by the program. The networks include:

- The global network under the form of a *.gefx* and *.json* file. Those format make possible to manipulate the graph using external software like Gephi.
- *.png* representation of each context and incremental graphs.
- A csv file with the edges, weight and sentiment of all incremental and context graph produced.
- Two csv files containing the clustering coefficient of each graph.

The other data produced by the program are csv files containing:

- the extracted speaker and audience of each dialog of the text with the perceived sentiment and the index of the conversation.
- all aliases associated to each character, the number of appearance of all thoses aliases and the inferred gender.
- all the proper nouns extracted with their inferred gender, categories and number of appearance.
- all primary and secondary links between names used for alias co-resolution.
- the spacings between each pair of consecutive dialogs, the frequency of each spacing value and the final threshold value used on spacing to delimit dialogs.
- the index of the begin and end of each conversation.

Chapter 4

Network analysis

In this second chapter of this work, the social network produced are analyzed. Some analysis have been produced by Waumans et al. (2015) and will be actualized using the correction of network extraction process. The analysis will also benefits to the addition of new metrics and the comparison between novels and scripts.

4.1 Connectivity

The size of each components of novels and scripts is on table 4.1. The number of components changes a lot between the texts. In all cases, there is a big components and other components are way smaller. The size of those secondary components is often 1 or 2. The bigger secondary components is located in TLT_1 and have a size of 6 while the main component have a size of 106. So the social network are composed of main set of connected characters while some characters are isolated or connected in small groups. Some of the isolated characters are due to errors in characters detection while others are due to dialog between minor characters or monologue of characters appearing in one scene only. From this we consider that the main component of a network is sufficient to observe the topology of the network. In next sections when only connected networks can be considered, the main component will be used. For instance the mean path length can be computed only on a connected network.

4.2 Distribution of degree

Waumans et al. (2015) has shown that characters social networks extracted from literature tend to have degree distribution following powerlaw. From this statement, the networks have been considered as scale-free. The modification in character extraction decreases a lot the number of characters so we should check that this observation hold.

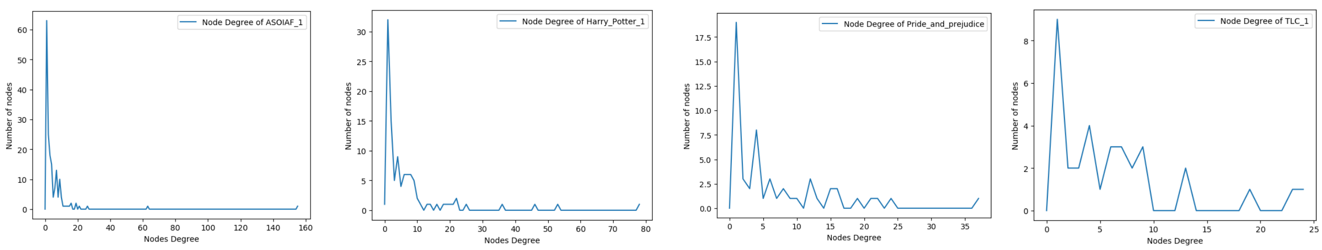

Figure 4.1: Degree distribution of social networks extracted from the first novel of the series “A song of ice and fire”, “Harry Potter”, “The Lunar Chronicles” and from the novel “Pride and prejudice”.

Title	Components Size
ASOIAF_1	[176]
ASOIAF_2	[268]
ASOIAF_3	[313, 1, 1, 2, 1]
ASOIAF_4	[263, 1]
ASOIAF_5	[374, 1, 1]
Harry_Potter_1	[105, 1]
Harry_Potter_2	[118, 1]
Harry_Potter_3	[123, 1]
Harry_Potter_4	[161, 1]
Harry_Potter_5	[269, 1]
Harry_Potter_6	[195]
Harry_Potter_7	[190, 1, 1, 1]
His_Dark_Materials_1	[106]
His_Dark_Materials_2	[72, 1, 1, 1, 1, 1, 1]
His_Dark_Materials_3	[79, 1]
Les_Miserables_1_4	[67, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1]
Pride_and_prejudice	[54]
TLC_1	[34]
TLC_2	[43]
TLC_3	[54]
TLT_1	[6, 106, 1, 1]
TLT_2	[96, 1, 1, 1, 1, 1]
TLT_3	[93, 1]
TMI_1	[93, 1]
TMI_2	[99]
TMI_3	[86, 1]
TMI_4	[95, 1]
TMI_5	[124, 1, 1]
TMI_6	[152, 1, 1, 1, 1]
TRWC_1	[58, 1, 1, 1, 1, 1]
TRWC_2	[51, 1, 1, 1, 1]
TRWC_3	[63, 1, 1]
TRWC_4	[68, 1, 1, 1]
TWOT_00	[108, 1, 1, 1, 1]
TWOT_01	[187, 1, 1, 1]
Alien_1_SCRIPT	[9]
Alien_2_SCRIPT	[35]
Alien_3_SCRIPT	[30, 1, 1, 1, 1, 1]
Blackkklansman_SCRIPT	[32, 1, 1, 1]
Black_Panther_SCRIPT	[102, 1]
Boyhood_SCRIPT	[69, 1]
Halloween_SCRIPT	[32, 3, 2, 1, 1, 2, 2]
Harry_potter_1_SCRIPT	[58]
Harry_potter_2_SCRIPT	[46, 1, 1]
Harry_potter_3_SCRIPT	[40, 1, 1, 1, 1, 2, 1]
Harry_potter_4_SCRIPT	[53, 1]
Harry_potter_6_SCRIPT	[47]
Harry_potter_7_SCRIPT	[71, 2, 2, 1, 1, 1, 1]
Joker_SCRIPT	[39, 1, 1, 1, 1, 1, 2]
Jurassic_Parc_1_SCRIPT	[32, 1]
Jurassic_Parc_2_SCRIPT	[2, 31]
Jurassic_Parc_3_SCRIPT	[24, 1, 1]
Lord_of_the_Rings1_SCRIPT	[29, 1, 1]
Lord_of_the_Rings2_SCRIPT	[41, 1, 1, 1, 1, 1]
Lord_of_the_Rings3_SCRIPT	[62, 1]
pride_and_prejudice_SCRIPT	[1, 58, 1]
Shrek_1_SCRIPT	[2, 1, 25, 1, 1, 1, 1, 1]
The_devil_wears_prada_SCRIPT	[33, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
Thor_Ragnarok_SCRIPT	[35]
Titanic_SCRIPT	[80, 1, 2, 1, 1, 1, 1, 1]

Table 4.1: Size of the components of each novel and script. Some title are under the form of an abbreviation, the corresponding title is in the appendix. The title of movies script are terminated with the mention *SCRIPT*.

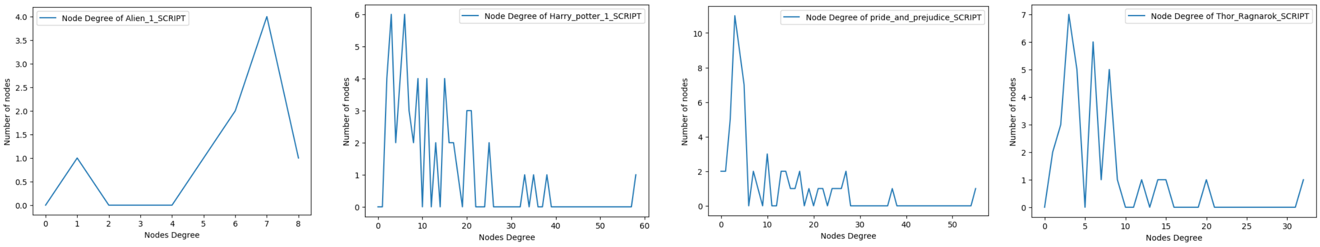

Figure 4.2: Degree distribution of social networks extracted from the script of the first movie of the saga“Harry Potter”, “Alien” and from the movies “Pride and prejudice”, “Thor: Ragnarok”.

On figure 4.1, the degree distributions from 4 novels shows a big amount of nodes with small degree and less nodes with more degrees. The first graph look like a heavy-tail distribution but others seems to have a more flattened curve. Also we see that the graphs have many irregularities. For this reason I have added all the distributions extracted from novels in a single distribution to fit it. It was already the way used by Waumans et al. (2015).
The distributions extracted from 4 movies are also shown on figure 4.2. The first example shows a distribution completely different from the others networks, this is due to the fact that movie is a “huis-clos”: all the narrative happen in a close environment with a limited amount of characters. In such movie, it’s intuitive that most characters are linked, this should also happened to books following this principle. However no books from our collections present this specificity, this could be a coincidence. An other hypothesis is that movie are more likely to present configurations with a smaller number of characters as this is easier to produce. Other books present a decreasing graph like novels but it seems polynomial. Like novels, the degree distribution have been summed in order to observe their general behavior.
The figure 4.3 shows the mean distribution on all novels with an exponential fit and a power-law fit. The power-law distribution seems to be a better approximation of the degree distribution. On the second sub-figure which is a zoom on the left of the graph, we can see that the exponential fit don’t grow as fast as the distribution while the power-law fit have an equivalent growth. On the third figure which is on a log-log scale, the distribution seems linear like a power-law distribution for small values of x. From this we can conclude that the degree distribution of novels follows a power-law like scale-free networks. The parameter of this power-law is -1.59962459.

The mean distribution on all scripts appears on figure 4.4. On the first figure, it’s difficult to say whether the exponential or power-law distribution are the best approximation. However on the next figures we see that the distribution is more linear on a log scale than on a log-log scale. The exponential seems to be the best approximation but is not a perfect fit. The distribution is may be the composed of an exponential and a power-law. It’s a major difference between novels and script.

4.3 Measures

Multiple measures have been performed on social networks: mean path length, mean degree and clustering coefficients (transitivity and global clustering coefficients). For each graph, a small-world network, a random network and a scale-free network have been generated with approximately the same number of nodes and edges. Measure have also been performed on those graph in order to allow comparison between different type of network and make easier the classification of characters social network.

The small-world networks are created using the method given in section, with the same number

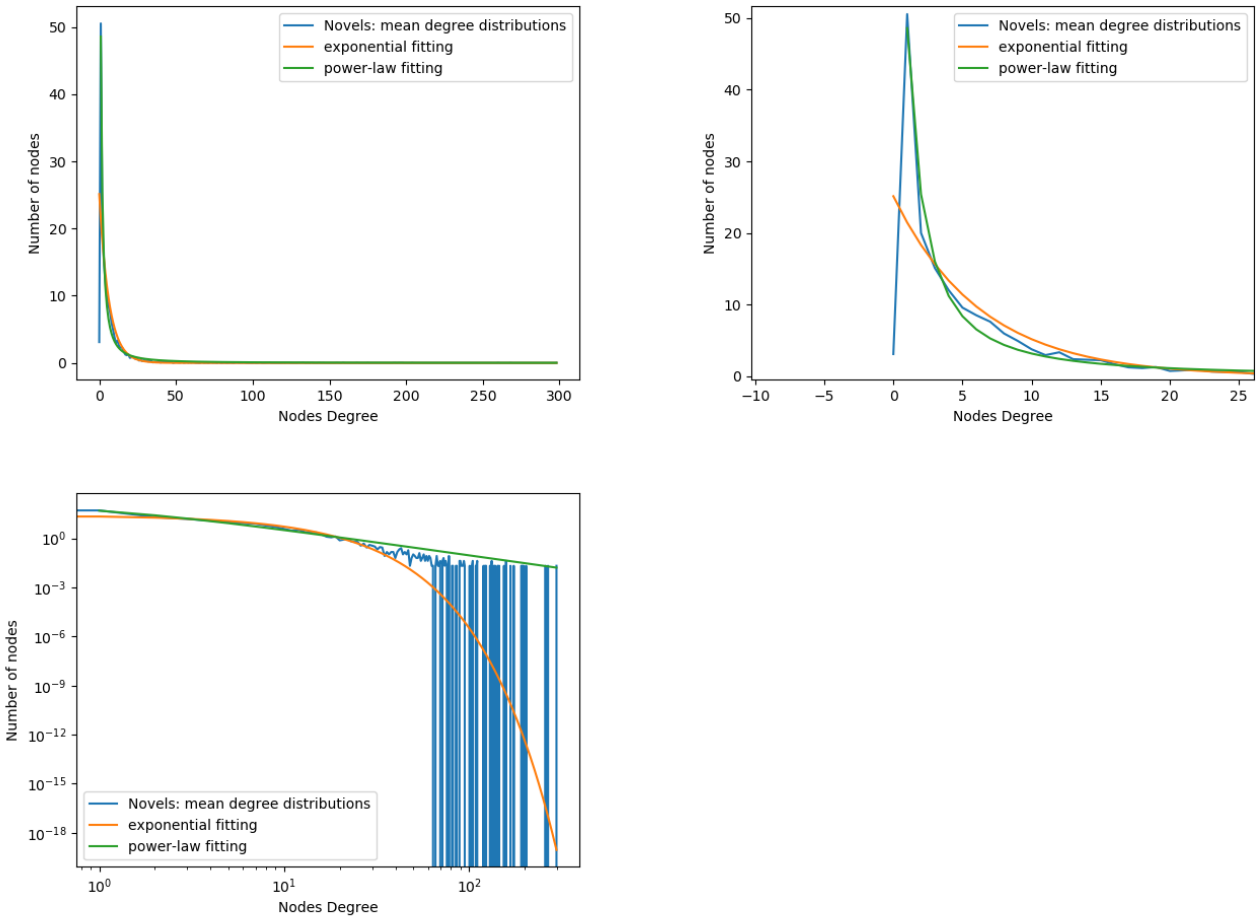

Figure 4.3: Mean degree distribution of all novels on linear and log-log scale. The power-law fit has for equation $f(x) = 147.23761823 \cdot x^{-1.59962459}$ and the exponential fit has for equation $f(x) = 25.13373576 \cdot -0.15795688^{x}$.

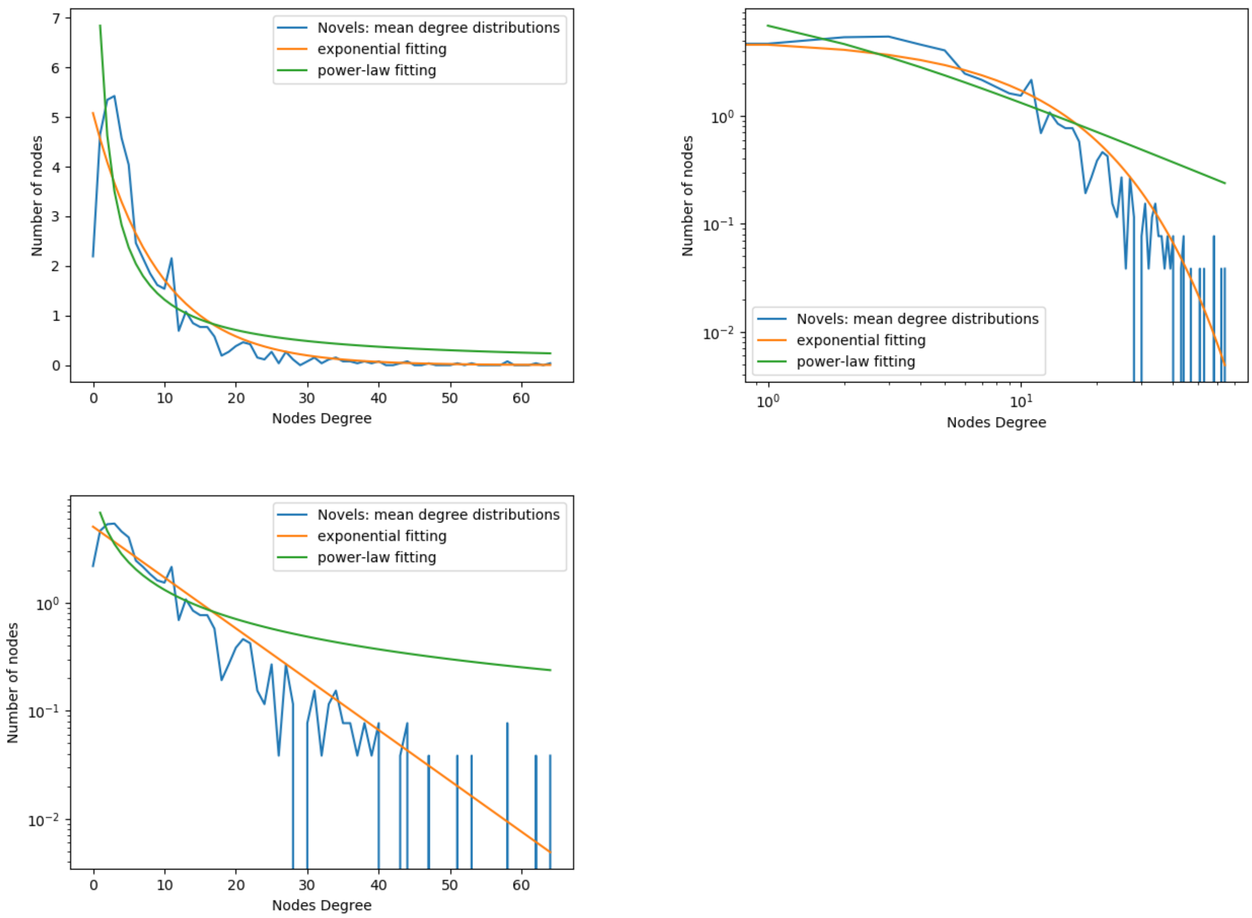

Figure 4.4: Mean degree distribution of all scripts on linear, log-log and log scale. The power-law fit has for equation $f(x) = 14.05704682 \cdot x^{-0.96477256}$ and the exponential fit has for equation $f(x) = 5.51683625 \cdot -0.11227941^{x}$.

of nodes and edges than the original graph and with a probability of 0.2 .
The scale free graph is created following the method of section, with the same number of node than the original graph. The m value used is computed using the parameters of the original network: $m = [NbrOfEdge/NbrOfNode]$. This gives a rough approximation of the number of edges of the network.

The random network is created using the method from section , with a number of edges following a random variable centered around the number of edges of the social network.
Lattice have not been generated as each type of lattice have approximately always the same values for clustering coefficient and mean degree. The mean path length of lattice growth with with $\sqrt{N}$ for square lattice.

4.3.1 Clustering coefficient

4.3.2 Mean Path Length

List of Figures

List of Tables

Bibliography

Agarwal, A., Kotalwar, A., and Rambow, O. (2013). Automatic extraction of social networks from literary text: A case study on alice in wonderland. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1202–1208, Nagoya, Japan. Asian Federation of Natural Language Processing.

Albert, R. and Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74:47–97.

Antoniou, I. and Tsompa, E. (2008). Statistical analysis of weighted networks. *Discrete Dynamics in Nature and Society*, 2008.

Barabasi, A.-L. (1999a). Albert, r.: Emergence of scaling in random networks. science 286, 509-512. *Science (New York, N.Y.)*, 286:509–12.

Barabasi, A.-L. (1999b). Emergence of scaling in random networks. 286:1–11.

Barabasi, A.-L., Dezso, Z., Regan, E., Yook, S.-H., and Oltvai, Z. (2003). Scale-free and hierarchical structures in complex networks. volume 661, page 1.

Barrat, A. and Weigt, M. (2000). On the properties of small-world network models. *The European Physical Journal B*, 13:547–560.

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python.* O'Reilly Media, Inc., 1st edition.

Burcher, M. and Whelan, C. (2017). Social network analysis as a tool for criminal intelligence: Understanding its potential from the perspectives of intelligence analysts. *Trends in Organized Crime*, 21:1–17.

Butts, C. (2008). Social network analysis: A methodological introduction. *Asian Journal of Social Psychology*, 11:13 – 41.

Coll Ardanuy, M. and Sporleder, C. (2014). Structure-based clustering of novels. In *Proceedings of the 3rd Workshop on Computational Linguistics for Literature (CLFL)*, pages 31–39, Gothenburg, Sweden. Association for Computational Linguistics.

Crocker, T. (n.d.). The power of social network analysis. `https://www.policechiefmagazine.org/power-social-network-analysis`. Retrieved 5 August 2020.

Davis, P. T., Elson, D. K., and Klavans, J. L. (2003). Methods for precise named entity matching in digital collections. In *2003 Joint Conference on Digital Libraries, 2003. Proceedings.*, pages 125–127.

De Smedt, T. and Daelemans, W. (2012). Pattern for python. *The Journal of Machine Learning Research*, 13:2063–2067.

Dekker, N., Kuhn, T., and van Erp, M. (2018). Evaluating social network extraction for classic and modern fiction literature. *PeerJ Prepr.*, 6:e27263.

Dekker, N., Kuhn, T., and van Erp, M. (2019). Evaluating named entity recognition tools for extracting social networks from novels. *PeerJ Computer Science*, 5:e189.

Diestel, R. (2005). *Graph Theory (Graduate Texts in Mathematics)*. Springer.

Elsner, M. (2012). Character-based kernels for novelistic plot structure. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 634–644, Avignon, France. Association for Computational Linguistics.

Elson, D., Dames, N., and McKeown, K. (2010). Extracting social networks from literary fiction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 138–147, Uppsala, Sweden. Association for Computational Linguistics.

Elson, D. K. and McKeown, K. R. (2010). Automatic attribution of quoted speech in literary narrative. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, page 1013–1019. AAAI Press.

Estrada, E. (2013). Graph and network theory in physics.

Gudivada, V. and Arbabifard, K. (2018). *Open-Source Libraries, Application Frameworks, and Workflow Systems for NLP*, pages 31–50.

He, H., Barbosa, D., and Kondrak, G. (2013). Identification of speakers in novels. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1312–1320, Sofia, Bulgaria. Association for Computational Linguistics.

Honnibal, M. and Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Jeong, H. and Barabasi, A.-L. (1999). Diameter of the world-wide web. *Nature*, 401:130–131.

Kumar, P., Verma, P., and Singh, A. (2018). *A Study of Epidemic Spreading and Rumor Spreading over Complex Networks*, pages 131–143.

Labatut, V. and Bost, X. (2019). Extraction and analysis of fictional character networks: A survey. *CoRR*, abs/1907.02704.

Li, L., Alderson, D., Doyle, J. C., and Willinger, W. (2005). Towards a theory of scale-free graphs: Definition, properties, and implications. *Internet Mathematics*, 2(4):431–523.

Liljeros, F., Edling, C. R., Amaral, L. A. N., Stanley, H. E., and Åberg, Y. (2001). The web of human sexual contacts. *Nature*, 411(6840):907–908.

Martschat, S. and Strube, M. (2015). Latent structures for coreference resolution. *Transactions of the Association for Computational Linguistics*, 3(0):405–418.

Milgram, S. (1967). The small-world problem. *Psychology Today*, 1(1).

NetworkX developer team (2014). Networkx.

Nicodème, T. (2015). Etude de la topologie de réseaux d'acteurs extraits à partir de romans célèbre. Master's thesis, Université Libre de Bruxelles.

Ostroumova Prokhorenkova, L. and Samosvat, E. (2014). Global clustering coefficient in scale-free networks.

Qi, P., Dozat, T., Zhang, Y., and Manning, C. D. (2018). Universal dependency parsing from scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 160–170, Brussels, Belgium. Association for Computational Linguistics.

Quang Dieu, T. and Jung, J. (2015). Cocharnet: Extracting social networks using character co-occurrence in movies. *Journal of Universal Computer Science*, 21:796–815.

Ribeiro, E., Macambira, M., and Neiva, E. (2017). *Social Network Analysis in Organizations as a Management Support Tool*, pages 243–265.

Scott, J. (1988). Social network analysis. *Sociology*, 22(1):109–127.

Scott, J. P. and Carrington, P. J. (2011). *The SAGE Handbook of Social Network Analysis*. Sage Publications Ltd.

Selvarajoo, K. (2019). Large scale-free network organization is likely key for biofilm phase transition. *Engineering Biology*, 3.

Stanford NLP Group (n.d.). Standford corenlp project page. `https://nlp.stanford.edu/software/`. Retrieved 30 July 2020.

Svenson, P. (2006). Complex networks and social network analysis in information fusion. pages 1 – 7.

Tabassum, S., Pereira, F., Fernandes, S., and Gama, J. (2018). Social network analysis: An overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8:e1256.

Ter Wal, A. and Boschma, R. (2009). Applying social network analysis in economic geography: Framing some key analytic issues. *The Annals of Regional Science*, 43:739–756.

Thilagam, P. S. (2010). Applications of social network analysis. In *Handbook of Social Network Technologies*.

Vala, H., Jurgens, D., Piper, A., and Ruths, D. (2015). Mr. bennet, his coachman, and the archbishop walk into a bar but only one of them gets recognized: On the difficulty of detecting characters in literary texts. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 769–774, Lisbon, Portugal. Association for Computational Linguistics.

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442.

Waumans, M. C., Nicodème, T., and Bersini, H. (2015). Topology analysis of social networks extracted from literature. *PLOS ONE*, 10(6):1–30.

Westaby, J., Pfaff, D., and Redding, N. (2014). Psychology and social networks a dynamic network theory perspective. *The American psychologist*, 69:269–84.

Wiseman, S., Rush, A. M., and Shieber, S. M. (2016). Learning global features for coreference resolution. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 994–1004, San Diego, California. Association for Computational Linguistics.

Yadav, V. and Bethard, S. (2018). A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2145–2158, Santa Fe, New Mexico, USA. Association for Computational Linguistics.