ECOLE
POLYTECHNIQUE
DE BRUXELLES

ULB

UNIVERSITÉ LIBRE DE BRUXELLES

Première ligne de titre du mémoire Deuxième ligne de titre du mémoire

Ligne du sous-titre du mémoire

Mémoire présenté en vue de l'obtention du diplôme
d'Ingénieur Civil [...] à finalité [...]

[Prénom Nom]

Directeur
Professeur [Prénom Nom]

Co-Promoteur
Professeur [Prénom Nom]

Superviseur
[Prénom Nom]

Service
[Nom du service]

Année académique
20xx - 20xx

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 links among individuals as a network in order to analyze them.

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.

This master thesis consists firstly in the development of a software that extracts social networks 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 (Waumans et al., 2015) and 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.

Chapter 2

Extraction of the network

2.1 Character identification

“Character identification consists in detecting which characters appear in the considered narrative, and when exactly they appear in this narrative” (Labatut and Bost, 2019). The first step is very easy with scripts as the characters speaking in each dialog are manually annotated. However novels don’t contain such annotations and the set of characters has to be extracted from the text. A method has been developed to respond to this problem. In both cases, the second step is easier and consists only in binding each occurrence of character mention to the corresponding character.

2.1.1 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). Here we will focus on automatic method that can be used on many different text. 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.

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 the current state of the art contains method 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

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 tagger 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.
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.

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-names a binding between proper names are divided between cluster, each of them being associated to a single character and cluster 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.
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. 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.

Proposed method of character extraction

The character identification process follow the same steps as the original research but some of these steps have been modified. These steps are:

1. The detection of potential listeners and speakers in all sentences of the text, using a POS-Tagger.

2. The extraction of a list of characters from the text.

3. The assignation of speakers and listeners in each dialog sentence.

The way those steps are performed in the original and new algorithm is described in the next sections.

The main modification is that in the original algorithm was based of single words characters names associated to chunks to identify this character in the text while the presented algorithm use characters names composed of multiple words. Those names will be called ”multi-tokens” names here despite the fact that some of them are composed by a single word.

”Multi-tokens” names generations:
Those names are generated from chunks, all proper nouns following each other are considered as a ”multiple tokens” name. Chunks that are made of multiple sequence of proper nouns could lead to the generation of multiple names.
Ex:

chunk	**names extracted**
Ron and Hermionne	Ron, Hermionne
you Mr. H. Potter	Mr. H. Potter
yer brother Charlie	Charlie

Detection of potential speaker and listener

All the sentence of the text are separated in 2 categories: the sentence that are part of dialogues and sentences that are part of context description between dialogues. In dialogues, the object and subject of each sentence are designated as potential listener and speaker. In contexts, only the subject is extracted as potential speaker of the context which will be further used to detect speakers of the following dialog.
The extraction is made using the pattern POS-tagger. Initially the entity extracted were chunks

but the program has been modified to use multi-tokens words.

Extraction of character

Original paper: The extraction of characters is made using a 2 pass algorithm.

1. *First pass*: Reading of all headwords of chunks to detect proper nouns. For each chunk of each dialog context, if the main word is labelled as a proper noun, is composed of at least 3 letters and is not only uppercase. Proper nouns are added to the list of proper names, their number of appearance is recorded and the associated chunks are stored.

2. *Second pass*: Reading of all chunks to check if some proper names appear in the same chunk.

3. Pairing of proper names if the less common appear with the most common more than 1 times over 3.

4. Pairing of proper nouns sharing a common root.

5. Paired nouns are clustered and considered as aliases. The name that appear the most in each cluster will be used to represent all the cluster in the social network. The chunks stored by other names of the cluster are now linked with the most appearing name. They will be use to identify the character from a potential speaker.

Problem encountered by the algorithm Firstly the first pass detecting proper nouns causes many false positive. Some words are labeled as NP by the tagger without being capitalize, some words are capitalized and incorrectly labelled as NP because they are the first word of a sentence or refer to a proper word which is not a name. A non proper name incorrectly labelled a single time over multiple appearance will be considered as a proper name.
Example of words incorrectly labeled as proper names :
'gruffly', 'Saturday','Transfiguration',"You'd","Uncle","Yeah", "Wizard", "London". Te high number of false positive increase a lot the number of characters and make the structure of the network non trustful. On the book "Harry Potter 1" detected characters have been manually labelled as true positive or false positive. Only 69 over 193 are labeled as true positive.

The fact that proper nouns are only composed of a single token makes the identification of characters in multiple way.
The pairing is needed to link firstnames and lastnames 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 book "Harry Potter 1".

1. **When both the firstname and the lastname of a character are used to identify him, they are not paired**: "Harry" and "Potter" are labelled as different character.

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

3. **Some characters are refered 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”.

New algorithm: The new algorithm use also 2 pass and is based on multiple-tokens names. It uses the first pass on the previous algorithm with additional conditions to detect proper nouns and then extract the multiple-tokens proper names during a second pass.

1. *First pass*: All chunks are read and the proper names are extracted from the chunk headword. To diminish the number of false positive, they should answer to multiple conditions to be extracted:
   - The POS-tagger label it as NNP but not as a location (tag NNP-Loc).
   - The word is capitalized but not entirely uppercase.
   - The word is not the first word of a sentence.
   - The word doesn't belong to a set of words that has been registered as non proper names. This set of words is the union of a set of english honorifics and a set of common english word provided by the library pattern.
2. *Second pass*: multi-tokens proper names are extracted from all the chunks headed by a proper noun. The number of appearance of each proper names is stored.
3. All proper names are analysed and split between a firstname, a lastname and an honorific. Their gender is infer from the lastname and the honorific when it is doable. They are classified following the composant that the name has. If a firstname has not been considered as a proper name but answer to the conditions, it is added to the list of proper names.
4. Names sharing the same firstname or lastname are linked together. 2 firstnames that appear together in a list of english nicknames or sharing a common root are considered as the same firstname. 2 type of linking are produced:
   The 1-linking that pairs immediately names having a lot in common. If one of the names gender was labelled as neutral, its gender is infer from the gender of the second name.
   The 2-linking makes 2 names potential pair. If the assigned gender of the names were not the same, its not modified.
   2 names labeled as male and female can not be paired.
5. Group of paired genderless names are paired with their gendered 2-value neighbor having the bigger number of appearance. Their gender is inferred from the gender of this neighbor.
6. Group of paired names without firstnames are paired with their most common neighbor having a firstname.
7. All the aliases are divided in cluster. A cluster is formed by a group of names such that it exist a 1-value path that link all of them. Each cluster designates a character. The most common name of each cluster is chosen as the headname and will be used in the network to designates the character. A new directed graph is build where each alias is pointing his headword.

Actor identification:

For each sentence of a conversation(dialog + context), the algorithm try to extract potential speakers and potential listeners. The subjects of the context are considered as potential speakers. Any proper names appearing in the context and being the subject of an other sentence are also taken. The subjects and objects extracted from the dialog are considered as potential listeners.

Alias coresolution:
Then an alias-table is constructed:
First pass: For each chunk of each dialog context, if the main word is labelled as a proper noun, is composed of at least 3 letters and is not only uppercase, this noun is added in the alias table.
Second pass: If 2 words from the alias table appear more than 2/3 of the time together, they are considered as alias.

Modification: To be considered as a potential character, nouns have to be:

- Be labelled as proper noun but not as an location.
- Begin with a capital letter.

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.

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.

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.

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.

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.

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.

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

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

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.

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.

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.

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

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.