Movie Reviews: Ontologies and Knowledge Graphs

W. Andy Holst

School of Professional Studies, Northwestern University

MSDS 453: Natural Language Processing

Dr. Syamala Srinivasan

August 8, 2021

Movie Reviews: Ontologies and Knowledge Graphs

This final paper in Natural Language Processing (NLP) focuses on creating ontologies and identifying "context" via Knowledge Graphs in order to evoke relationships which may exist between entities in the dataset of 200 movie reviews from our NLP course. The work focuses specifically on the creation of Ontologies using Stanford's Protege program, and on the creation of Knowledge Graphs by using "SpaCy" (see https://spacy.io/), a free, open-source library for NLP in Python.

Unfortunately, the results that are obtained from experiments conducted during the course of my research are quite disappointing. The implementation of knowledge graphs, while shown to be successful when deployed at extremely large scales such as Google's Knowledge Graph project, is challenged by the difficulties and subtleties of the written and spoken language that is contained with corpora (Dery, 2016). So, while a few interesting results are presented in this research, the overall result is that using knowledge graphs on corpora with many different styles and approaches (such as appear to be present in our 200 movie reviews).

Data

The data utilized in this research consist of the text only of 200 movie reviews that have been collected and compiled by students in this NLP course. Each movie review (the "text") is at least 500 words long.The data preparation, exploratory data analysis and visualization steps undertaken to prepare the dataset for subsequent research, analysis, and modeling were as follows:

- A Google Colaboratory Notebook (Python 3.7.11) was created.
- Python Packages imported as necessary for pre-processing and modeling, including: SpaCy (3.1.1), Pandas (1.1.5), Matplotlib (3.2.2), Re (2.2.1), and NetworkX (2.5.1).

- Much of the exploratory data analysis and visuals produced in this research paper involved plotting Knowledge Graphs using the NetworkX function called `from_pandas_edgelist()` which produces a graph from a Pandas DataFrame that contains an edge list.

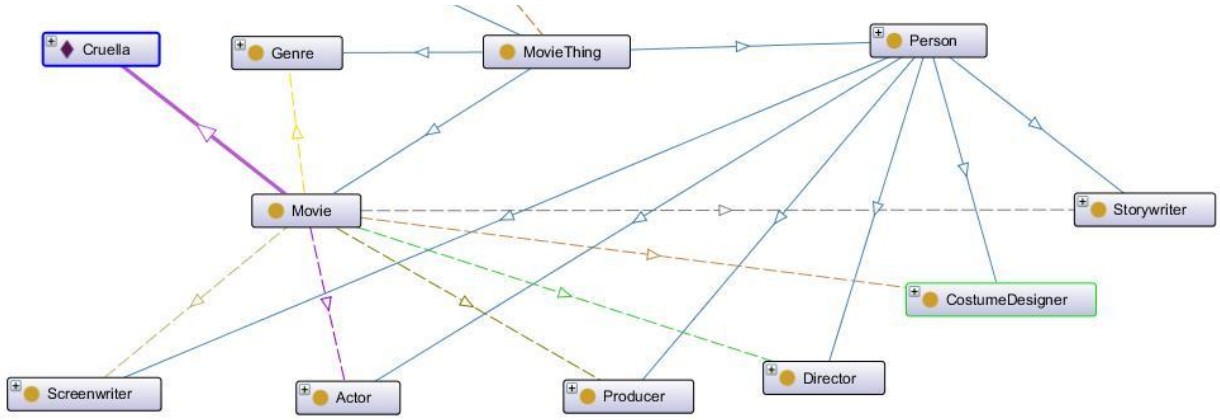
First Analysis: Ontologies of a small subset of the corpora

Ontology: Research Design and Modeling Method(s)

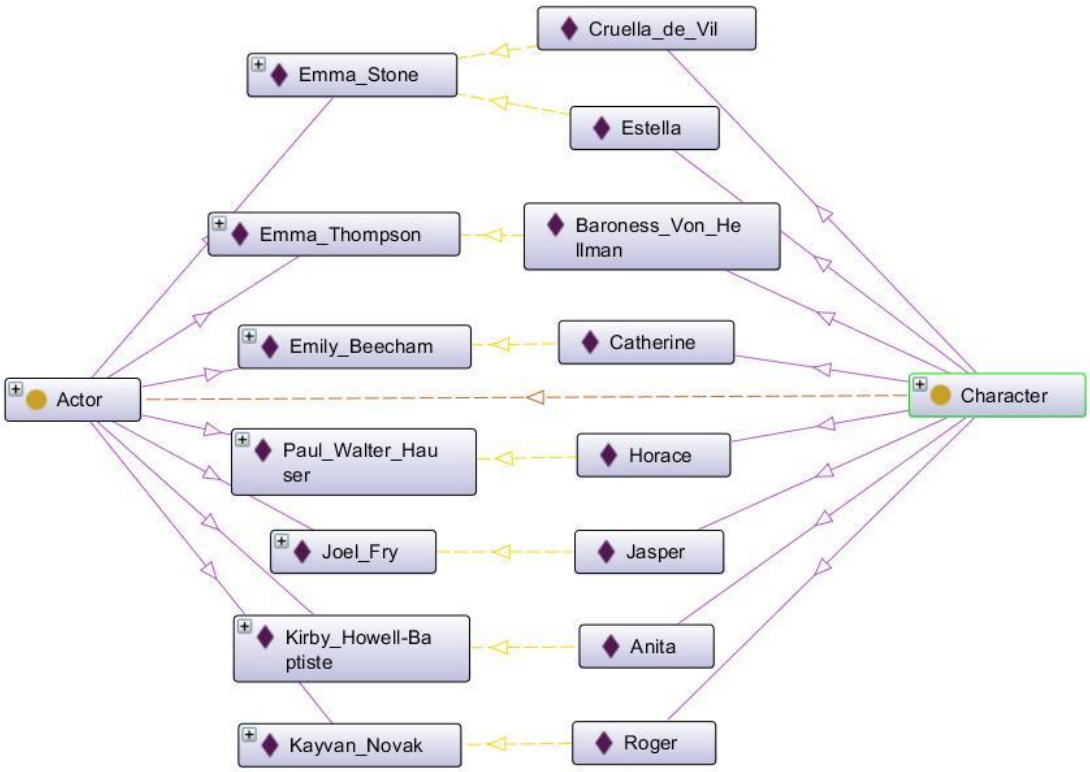
During week 6 of our NLP course, we created a basic ontology for our individual movie reviews (a subset of 10 reviews, based upon the movie “Cruella”) using the free open-source ontology editor created by Stanford, called Protege (https://protege.stanford.edu/) .

Ontology: Results, Analysis and Interpretation

What I found to be very useful during the use of Protege was how overriding themes emerged from the reviews. When taking a bottom-up approach to the ontology, I found two basic themes to emerge: The first theme is how "People" involved in making of a movie are all interconnected. By way of example Cruella's "Director" is "Craig Gillespie" the authors of reviews sometimes help readers to learn about Craig Gillespie, who is an Australian and is best known for his cynical portrayal of the famous Tonya Harding scandal in “I, Tonya”. So, one movie review author makes a recommendation about Cruella in hopes that you enjoy Gillespie's work (and perhaps his use of cynicism). So with this in mind, I really focused on building my ontology around the people within the films: Actors (and Characters), Directors, Screenwriters, Storywriters, Costume Designers, etc. Below is a visual of how the people of the movie are related:



The second theme that emerged for me while re-reading each of my reviews is the "telling of the story". When reading each review, it struck me that the authors really presented their narratives in a story format much the same way that the movies themselves evoke a story and mood, introducing characters (and which actor) and placing them in the most memorable scenes in the movie to evoke emotions which the readers will remember and hopefully be convinced to go watch the movie. This narrative style really makes it quite easy to put together the Character-->isPlayedBy-->Actor relationships in the ontology. One way of visualizing how the story emerges from each review is by centering the ontology on just this "isPlayedBy" object property. The visual below shows the different characters in Cruella and how they are associated with the Actors:



Second Analysis: Knowledge Graphs using spaCy

Knowledge Graphs: Research Design and Modeling Method(s)

The different experiments that were undertaken during research this week on Knowledge Graphs were primarily focused on the different pipeline options available in spaCY and ways of adjusting and filtering the results. SpaCy is designed specifically for production use and helps to build applications that process and “understand” large volumes of text (spaCy, n.d.). The spaCy library utilizes several different English language pre-processing pipelines, two of which are called `en_core_web_sm` and `en_core_web_lg` which seamlessly tokenize documents, and then tag parts and attributes of speech via one single function call `spacy.load()`. By splitting

documents into individual sentences and then tagging the different tokens, relationships begin to emerge from the text. These relationships are then visualized and analyzed by exploring combinations of knowledge graph Entity-Pair objects, which are comprised of three elements: “Relations / Edges”, “Source Nodes” and “Target Nodes”. Generally, an Entity-Pair within a Knowledge Graph creates visual connections between “Sources” and “Targets” via their “Relations” (Stanford, n.d.). So, for example in the figure below, there are two Entity-Pairs and the “Sources” are “Keira Knightley” and “Orlando Bloom”, who each “play” (the relation) the “Target” characters “Elizabeth” and “Will Turner”, respectively (both from Pirates of the Caribbean).

Pipeline Experiments: As a first experiment, I explored the differences between Knowledge Graphs produced using the pre-processing pipelines `en_core_web_sm` and `en_core_web_lg` which seamlessly tokenize documents, and then tag parts and attributes of speech. The “small” pipeline is 13.6 MB in size whereas the “large” pipeline 777 MB in size. *NOTE: For each experiment, I looked both at the entire corpus as well as singled out my own movie “Cruella” to aid in visualizations.*

Experiments to Find “Better” Relations: Next, realized that in certain Entity-Pairs produced “blank” results. For instance, the sentence `'I was an instant fan'`, erroneously returned

the following Entity-Pair: `['I', '']`. In an attempt to fix this, I learned that the pipeline was assigning the target “fan” to a part of speech called “attr”

```
I nsubj
was ROOT
an det
instant amod
fan attr
```

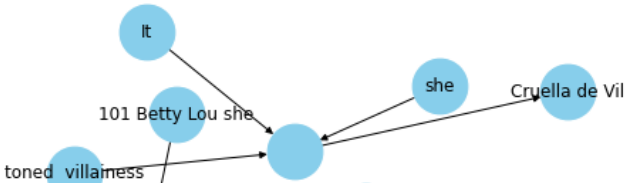
In this experiment, I attempted (unsuccessfully) to adjust the entity extraction algorithm `get_entities()` to account for the “attr” part-of-speech but unfortunately, I was unsuccessful.

Sentence Length Experiments: Next, I considered that perhaps complexity of sentences was affecting results. For instance, via some Google research I was able to ascertain and download the original dataset that our Knowledge Graph Python Code was used on originally. There was a recent Kaggle Tutorial that analyzed a dataset called `wiki_sentences_v2.csv` (Sanagapati, P., 2020). After conducting some Exploratory Data Analysis, I determined that the original dataset used in the tutorial had an average sentence length of 10 words whereas our movie review dataset has an average sentence length of 22 words. So, I decided to only analyze those sentences in our dataset that were 10 words or less.

Exploring Subsets of Knowledge Graph Plots: Next, since a Knowledge Graph of our entire corpus produced a result which was not legible, I decided to look at the entire Knowledge Graph of my own movie “Cruella”. The complete Knowledge Graph for “Cruella” had 268 Entity Pairs, so I organized the entity pairs alphabetically and then split them into four equally sized plots of 67 entity pairs each. This made the visualization of each Complete Knowledge Graph more readable for interpretation

Focused Knowledge Graphs on Specific Terms: A helpful way of visualizing relations is achieved by displaying only those entities which contain a specific Relation, Source Node or Target Node, essentially filtering for specific terms. For instance (as I noted previously above), I found that the relation/edge “plays” in some instances did correctly reveal the actor who played certain characters in a movie. Another example of filtering was visualizing the Source Node of the Movie Name itself (“Cruella”) and how it might reveal some clues.

Eliminating “Blank” Nodes: As mentioned above, I did notice quite a few “blank” nodes (see visual example below). So, I first tried (unsuccessfully) to correct these blanks, but I could not find a way to linguistically accomplish this. Instead, I explored filtering out blank results out from the visuals, hopefully this would improve readability (it did not help).



Knowledge Graphs: Results, Analysis and Interpretation

Pipeline Experiment Results/Analysis/Interpretation: In my exploration of different sizes of pipelines, I did not notice any appreciable differences, except for one. The only key difference that I noticed was by using the larger pipeline `en_core_web_lg` the total number of “blank” nodes was reduced some. For instance, in my “Source” nodes, the “blank” nodes were reduced from a count of 647 down to 566 (still a large number, but at least reduced!)

	647
it	152
It	144
that	115
he	104

	566
it	154
It	146
that	124
he	107

`en_core_web_sm` Source Nodes | `en_core_web_lgs` Source Nodes

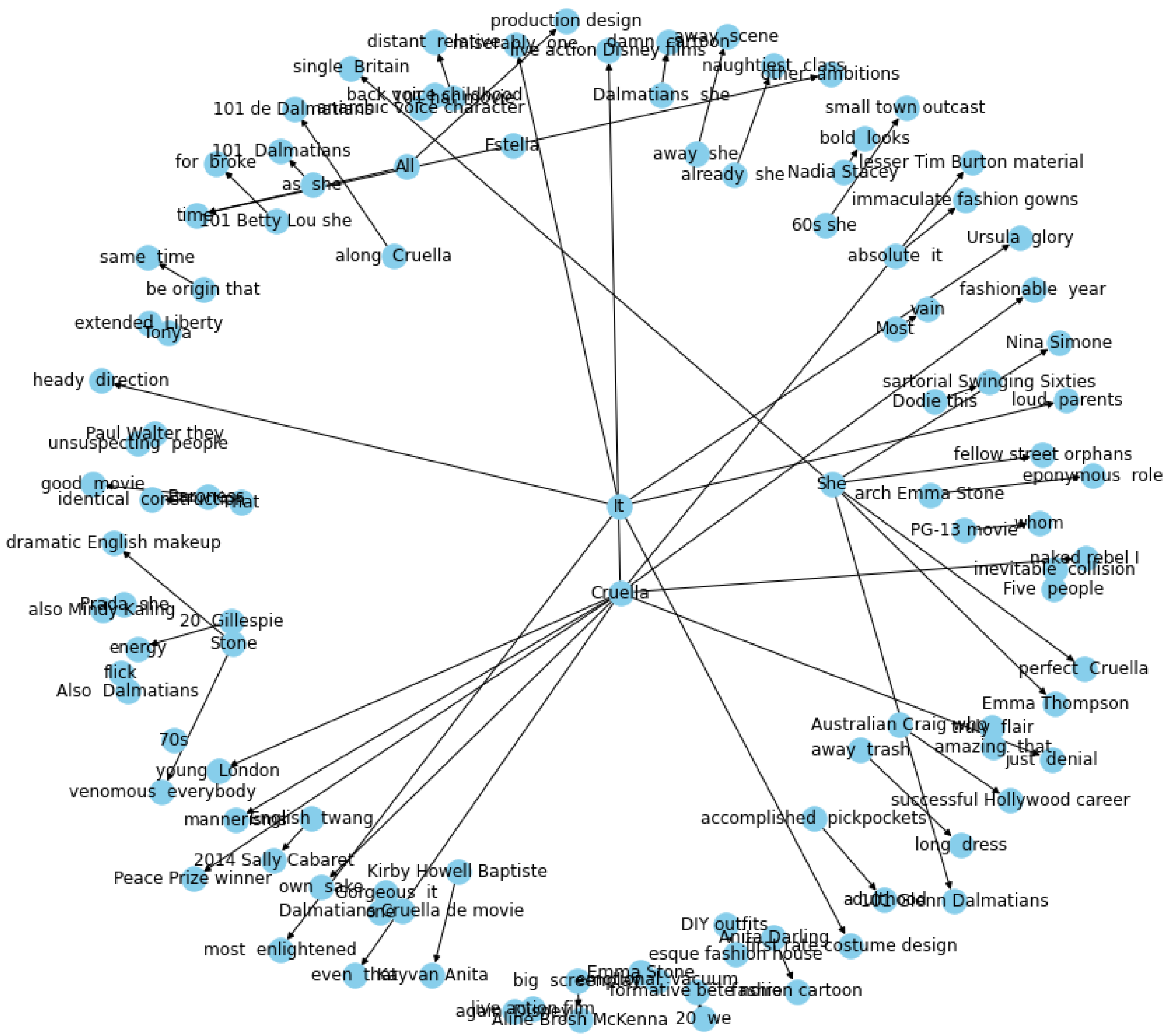
Results: larger pipeline reduced "blank" Nodes and Edges

"Better" Relations Results/Analysis/Interpretation: In my exploration of the "attr" part-of-speech, I was not successful in producing any improved results.

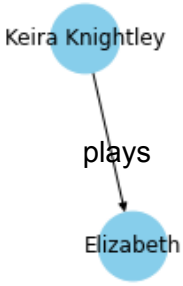
Sentence Length Experiments Results/Analysis/Interpretation: Testing out sentences of only 10 words or less did not appear to produce any better (or worse) results, so were inconclusive.

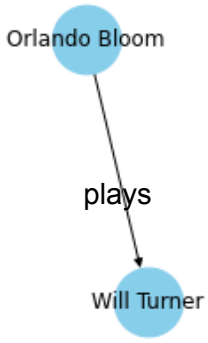
Subsets of Knowledge Graph Plots and Eliminating "Blank" Nodes

Results/Analysis/Interpretation: I did find that creating subsets of the Knowledge Graph Plots, when also combined with eliminating of the "blank" nodes, did help in some interpretation and readability. The plots on the next two pages show how it is possible in the movie "Cruella" to see how the Nodes "Cruella", "She" and "It" are clustered closely together and thus start referring to the same entity. These nodes then point out to notable terms such as "live action Disney films", "Tim Burton material", "costume design", "young London" and "heady direction". So there are definitely some themes and elements of sentiment emerging from this graph that focuses on the central terms of the movie.


production design
single Britain
101 de Dalmatians
101 Dalmatians
for broke
as she
All
Estella
101 Betty Lou she
along Cruella
same time
be origin that
heady direction
dramatic English makeup
energy
Also Dalmatians
Stone
70s
young London
venomous everybody
English twang
2014 Sally Cabaret
Peace Prize winner
Kirby Howell Baptiste
most enlightened
Dalmatians
away scene
Dalmatians she
away she
already she
small town outcast
bold looks
Nadia Stacey
lesser Tim Burton material
immaculate fashion gowns
60s she
absolute it
Ursula glory
fashionable year
vain
Most
Nina Simone
sartorial Swinging Sixties
loud parents
It
fellow street orphans
She
eponymous role
arch Emma Stone
PG-13 movie
whom
Cruella
inevitable collision
Five people
perfect Cruella
Emma Thompson
Australian Craig
flair
away trash
just denial
successful Hollywood career
accomplished pickpockets
long dress
DIY outfits
Anita Darling
costume design
esque fashion house
fashion cartoon
Aline Brosh McKenna
20 we

Focused Knowledge Graphs on Specific Terms: As mentioned previously, one helpful way of visualizing relations was achieved by displaying only those entities which contain a specific Relation, Source Node or Target Node, essentially filtering for specific terms. For instance, I found that the relation/edge “plays” in some instances did correctly reveal the actor who played certain characters in a movie.





Conclusions

Unfortunately, the results presented above from the experiments conducted during the course of my research were quite disappointing. The implementation of knowledge graphs, while shown to be successful when deployed at extremely large scales such as Google’s Knowledge Graph project, is challenged by the difficulties and subtleties of the written and spoken language that are contained with corpora (Dery, 2016). So, while a few interesting results are presented in this research, the overall result is that using knowledge graphs on corpora with many different styles and approaches is not an effective means of evoking key concepts from corpora.

Note: All of the Knowledge Graphs produced during this research are attached in the HTML file which shows all code to produce the results.

References

Dery, S. (2016, December 21). *Challenges of knowledge graphs*. Medium. https://medium.com/@sderymail/challenges-of-knowledge-graph-part-1-d9ffe9e35214.

Sanagapati, P. (2020, September 8). *Knowledge graph & NLP Tutorial-(BERT,spaCy,NLTK)*. Knowledge Graph & NLP Tutorial-(BERT,spaCy,NLTK). https://www.kaggle.com/pavansanagapati/knowledge-graph-nlp-tutorial-bert-spacy-nltk.

spaCy, (n.d.). *spaCy 101: Everything you need to Know · SPACY usage documentation*. spaCy 101: Everything you need to know. https://spacy.io/usage/spacy-101.

Stanford, (n.d.). *1. introduction - What is a Knowledge Graph*. What is a Knowledge Graph? https://web.stanford.edu/class/cs520/2020/notes/What_is_a_Knowledge_Graph.html.