Assignment 3.

Ontology Plus Context and Modeling

MSDS 453: Natural Language Processing

Husein Adenwala

Northwestern University

1. Introduction & Problem Statement

The purpose of this assignment is to evaluate qualitative clustering analysis and Ontology/Graphs/knowledge graphs-based context modeling and to create a sentiment and genre classification model of movie reviews. The manual clustering and ontology-based context modeling will be done on the 10 reviews for the movie Frozen 2 and I will use deep learning model, specifically Long short-term memory (LSTM) recurrent neural network architecture model for sentiment and genre classification on all 249 reviews.

The purpose of clustering and ontology-based clustering is to evaluate how ontology/knowledge graphs can provide context-driven understanding of the words in the corpus and can incorporate equivalent classes and identity entity Co-Referencing in each.

In this research, I will identify key works in the 10 Frozen 2 reviews that will be used for qualitative clustering. I traversed the 10 reviews to identify the Entity relationship and used that to create an ontology in PowerPoint and in Protégé ontology software. I experimented with various vocabulary setups and hyperparameters for LTSM model (unidirectional and bidirectional). Results from my experiment suggest that bidirectional LSTM models perform better than unidirectional LSTM models; however, all the models performed poorly for sentiment classification.

2. Data preparation, exploration, visualization

The data consists of 249 movie reviews of 25 movies that was stored and uploaded in a csv file. The tables and plots below show the movies' names, the number of movie reviews and movie genres.

All 25 movies have 10 reviews except the movie Martian and Red Notice, which have 9 and 20 reviews respectively. The number of positive and negative reviews are equal for all movies except Martian. Also, it is important to note that the count for each movie genre is unbalanced; the Drama genre has only 9 reviews.

Movie Title	Review Type (pos or neg)	
Arrival	Negative	5
	Positive	5
Casino_Royale	Negative	5
	Positive	5
Cruella	Negative	5
	Positive	5
Encanto	Negative	5
	Positive	5
Frozen_2	Negative	5
	Positive	5
Groundhog_Day	Negative	5
	Positive	5
Guardians_of_the_Galaxy	Negative	5
	Positive	5
Hereditary	Negative	5
	Positive	5
Interstellar	Negative	5
	Positive	5
KingOfStatenIsland	Negative	5
	Positive	5
Lamb	Negative	5
	Positive	5
Mission_Impossible_Fallout	Negative	5
	Positive	5
Pacific_Rim	Negative	5
	Positive	5
Pirates_of_the_Caribbean_Curse_of_the_Black_Pearl	Negative	5
	Positive	5
Poltergeist	Negative	5
	Positive	5
Red_Notice	Negative	10
	Positive	10
Speed_Racer	Negative	5
	Positive	5
Spider_Man_3	Negative	5
	Positive	5
Sudden_Impact	Negative	5
	Positive	5
The_Martian	Negative	5
	Positive	4
The_Ring	Negative	5
	Positive	5
The_Conjuring_The_Devil_Made_Me_Do_It	Negative	5
	Positive	5
The_Matrix_Resurrecton	Negative	5
	Positive	5
Us_2019	Negative	5
	Positive	5

Genre of Movie	Number of Reviews
Action	70
Comedy	60
Drama	9
Horror	60
Sci_Fi	50
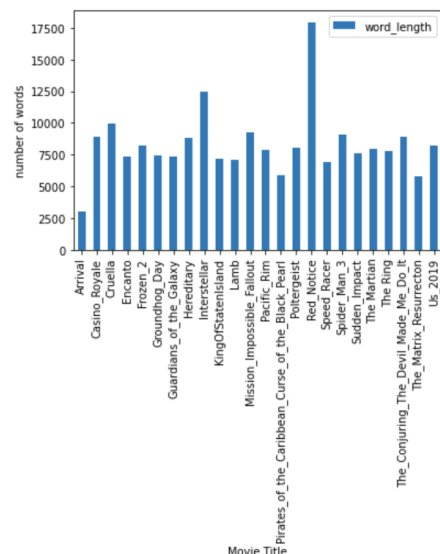
Data Processing

I experimented with lemmitization and stop work removal for creating a graph in python; however, the cleaned data adversely affected subject verb extraction. For the purpose of Graph analysis, I did minimal data cleaning.

Total number of words= 206952 : Total number of unique words= 19353

For LSTM models, I used tensorflow vectorization function, which maps text features to integer sequences.

After Stop word removal and lemmatization: Total number of words= 122245 : Total number of unique words= 17691.

Data was split into train, validation and test, with 199 reviews for train, 25 review for validation and 25 reviews for test.

The table below shows the count of each of the top 100 words after lemmitization and stopword removal:
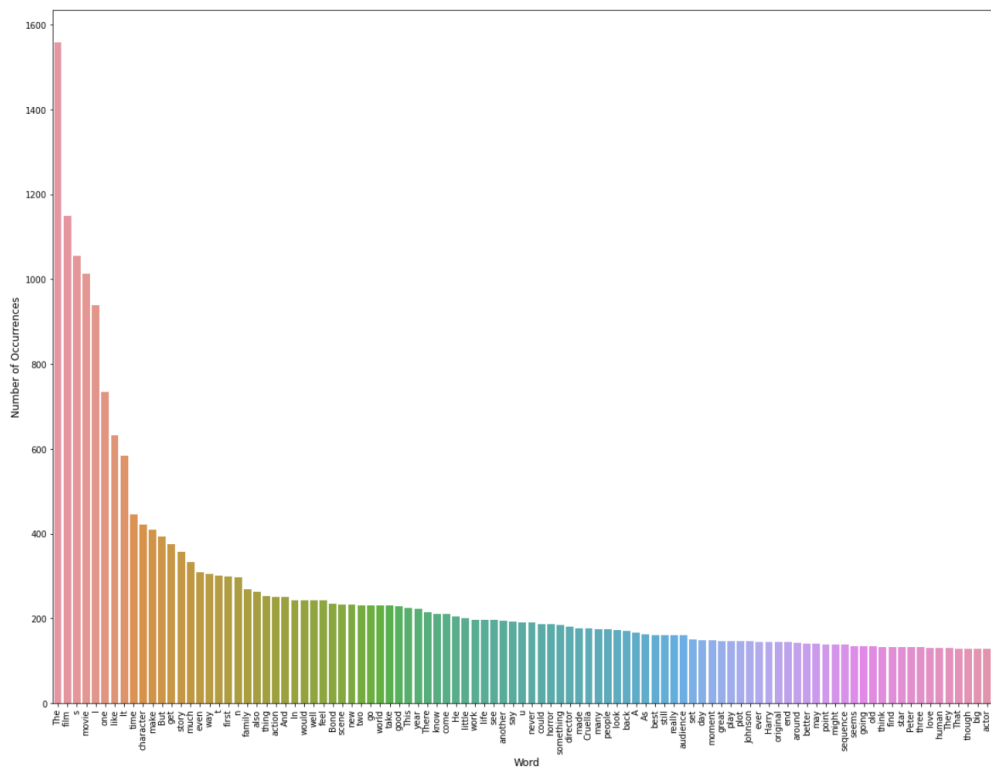
Word count per document

3. Research Design and Modeling Method(s)

Qualitative clustering and Ontology

For clustering, I manually extracted key terms that I found important and prevalent in my movie review documents (Frozen 2). I found cast names, movie names and character names to be important and adjectives, verbs and catch phrases that are commonly used to describe movies to be prevalent.

For ontology, I extracted Name entity relationships from the 10 documents; however, I was focused on proper Nouns and relationships

I then created the ontology in Microsoft power point and in Protégé software which has s plug-in architecture that can build both simple and complex ontology-based applications.

Graph

Creating a graph or ontology in python involved first using Spacy library which provides a pretrained model that has language understanding and can detect parts of speech and to some extent identity equivalent classes. For this research, I downloaded the "en_core_web_sm" spacy model which is trained on articles and blogs. I used it to categorize subjects, words and objects and saved it in a data frame. Subsequently, I used the Networks graph library that I used to create, solve, and visualize graph networks.

LSTM

For this research, Long Short-Term Memory (LSTM) models are analyzed and compared with varying architectures.

Long Short-Term Memory (LSTM) networks that are a type of recurrent neural network capable of handling long-term dependencies in sequence prediction problems.

I experimented with several vocabulary sizes and layers of LSTM models and regularization techniques.

All models have an embedding layer to create embedding vectors. The embedding layers get numeric data from the encoder that is created using the Text vectorization function. All models use the ReLU activation function in the dense and convolution layers, which is one of the popular activation functions in neural networks because it is computationally efficient and fixes the problem of vanishing gradient. I used validation accuracy as the metric to evaluate the performance. Finally, I used Sparse categorical cross entropy and Binary categorical entropy to measure the loss between labels and predictions and SoftMax and sigmoid activation function for the output layer.

Model research was done using Python programming language and TensorFlow. TensorFlow is a free and open-source software library for machine learning and artificial intelligence and TensorFlow can be implemented in python by Keras. Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow which is extremely user-friendly.

I used the sci-kit learn library which has functions to create a confusion matrix and report on precision, accuracy and F1 score.

Additionally, I used NumPy and Pandas library for data for EDA and finally, I also used seaborn package and matplotlib library to visualize the data and results.

4. Results

Part 1 Manual Clustering and Ontology :

I performed qualitative clustering of documents by traversing the document and selecting key words; however, they need contextual reference to make sense. Below are some key words and based on my interpretation, I found three clusters:

Doc1 Rating: positive	Doc2 Rating: positive	Doc3 Rating: positive	Doc4 Rating: positive	Doc5 Rating: positive	Doc6 Rating: Negative	Doc7 Rating: Negative	Doc8 Rating: Negative	Doc9 Rating: Negative	Doc10 Rating: Negative
authenticity	goodwill	exploited	Standout moments	Creativity	disappointing	familiar	subpar	Astonishing beauty	Stunning visuals
Ambitious	sweet spot	Comedy	Should be DVD offering	Inspiring visuals	Compulsory heterosexuality	Worthy Successor	lackluster	Animation Fantasy	Lack of originality
Worthy successor	besotted	Generation Taste	Not original	warmth	Convoluted	Not original	disappointed	Captivates audience	mediocre
Cluster 1 Positive	Cluster 1 Positive	Cluster 5 Genre Comedy	Cluster 2 Negative	Cluster 1 Positive	Cluster 2 Negative	Cluster 2 Negative	Cluster 2 Negative	Cluster 4 Genre Animation F	Cluster 3 Average

Ontology for the 10 documents captures the relationship between the key words in the document; however, it does not pick an average sentiment, it picks comedy and animation. There is some overlap between the ontology and the clustering pattern.
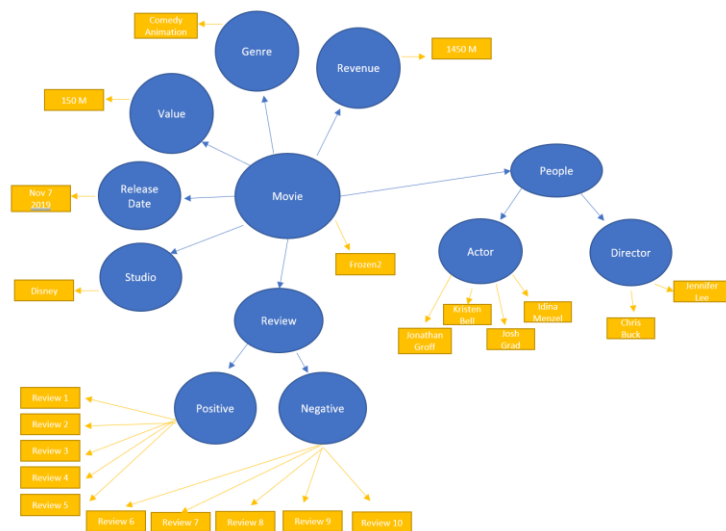
Part 3 Using Protégé to create Ontology :

The Protégé ontology tool allows the user to create graphs based on Class, subclass and instance hierarchy. It enables better abstraction of classes and instances; for example, rating is class and positive and negative are instances of the classes.

The ontology created by protege for the Movie Frozen 2 has the following hierarchy:
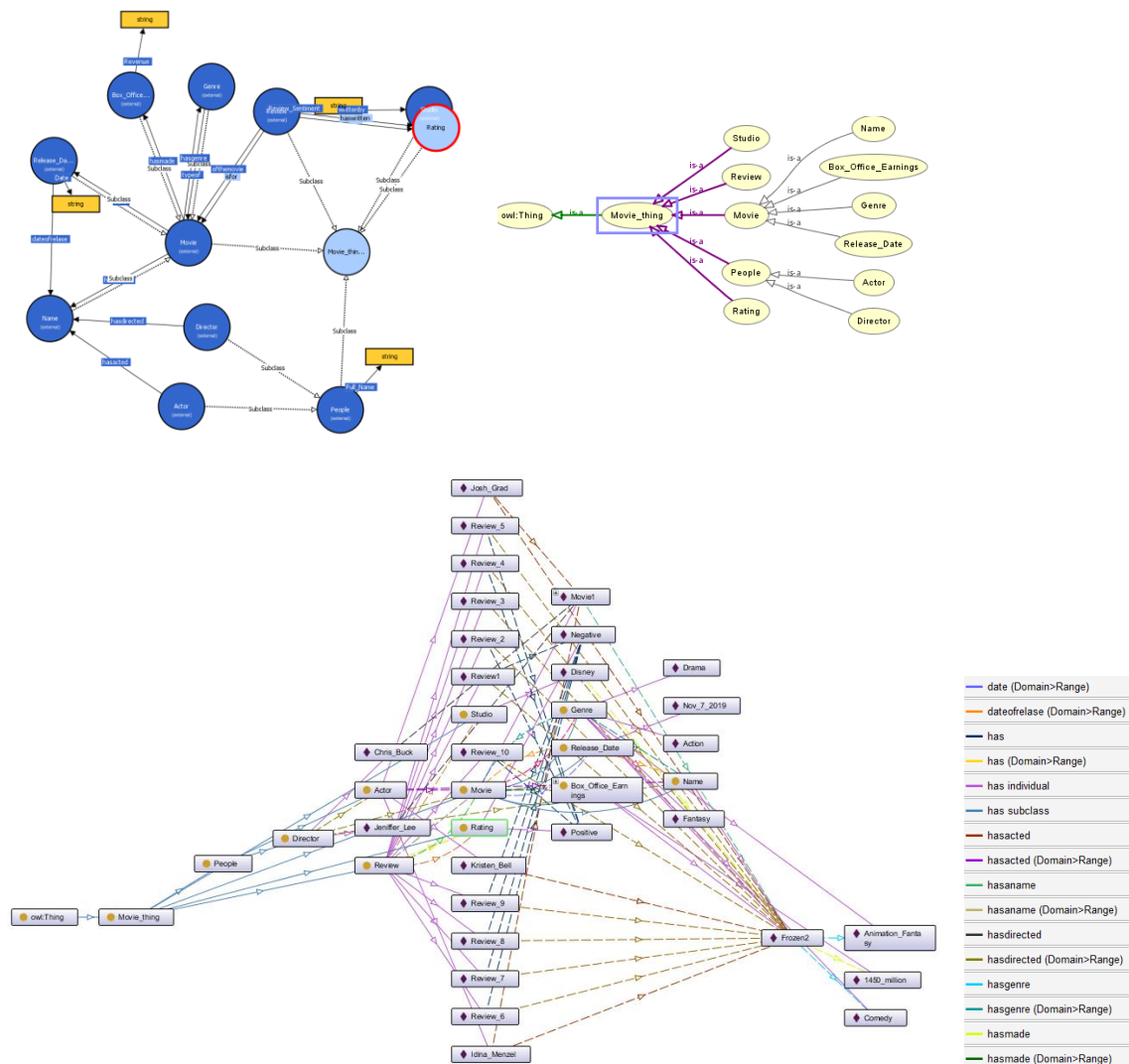

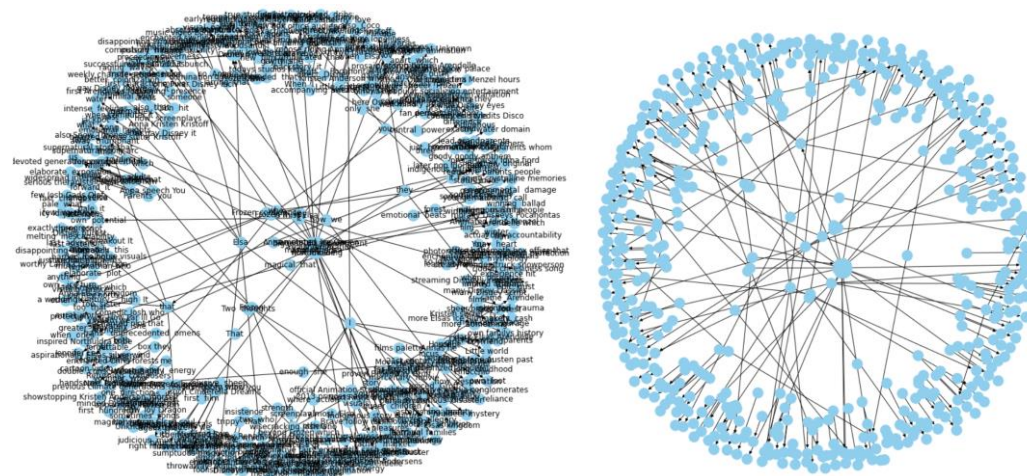This can be visualized in the graphs below:

Part 4 Graph using python:

I used the Spacy library in python to extract Subject, Verb and Object to create a Graph/Ontology for the 10 movies Review for Frozen 2.

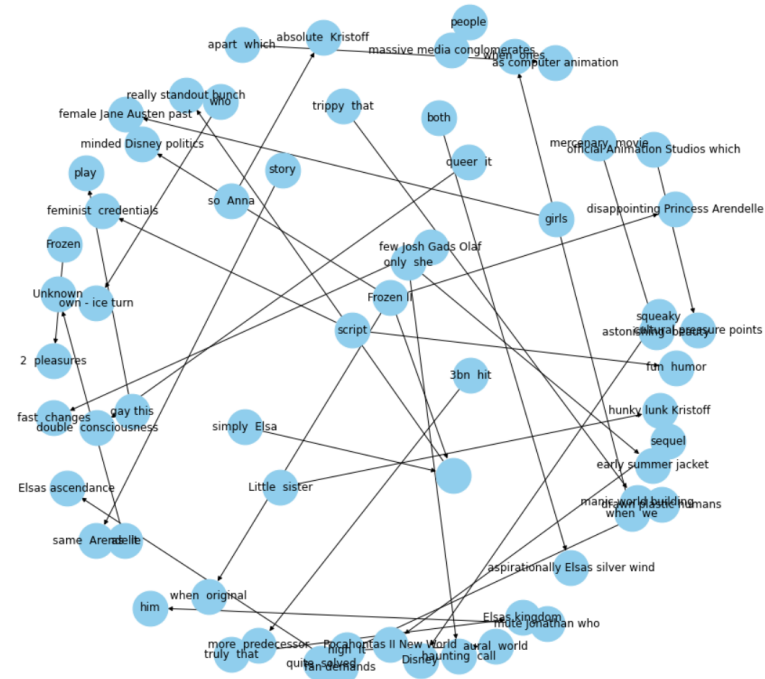All Entity relationships are shown below with and without labels:

Ontology Plus Context and Modeling

Isolatin on most common 5 relationships

```
is          35
are         11
has          8
was          6
```
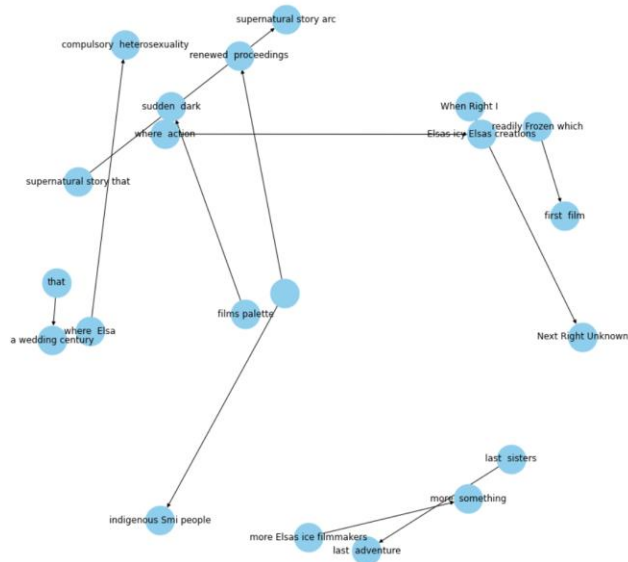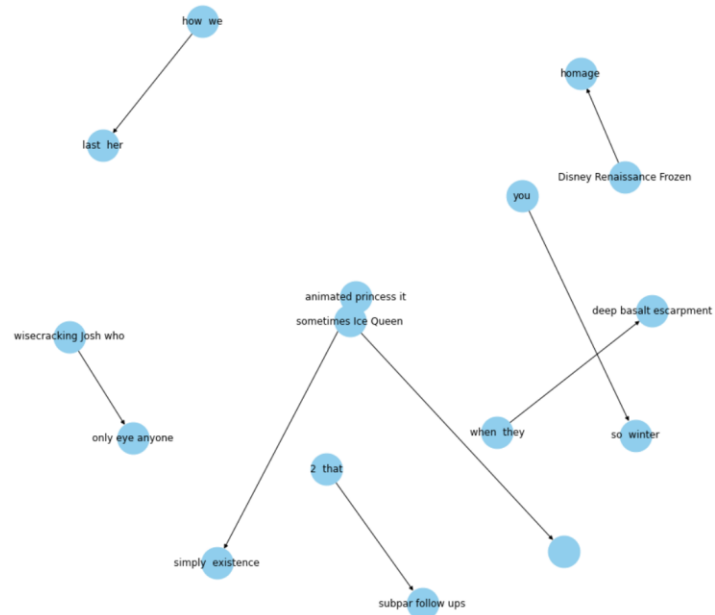
Isolatiing on relationship/verb "IS"

Ontology Plus Context and Modeling

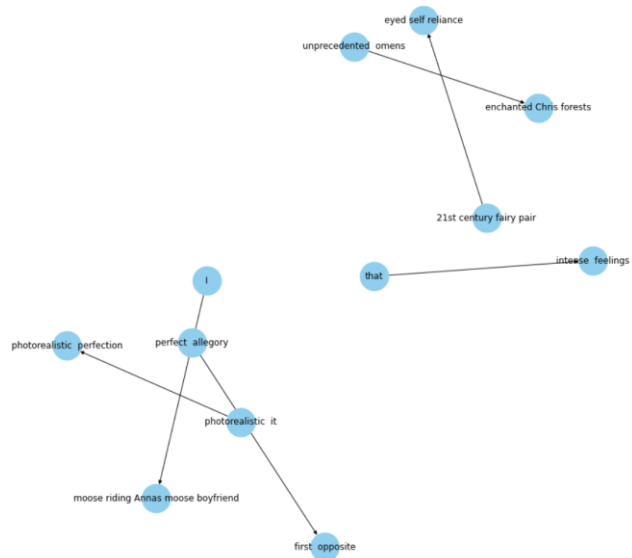**Isolatiing on relationship/verb** "ARE"
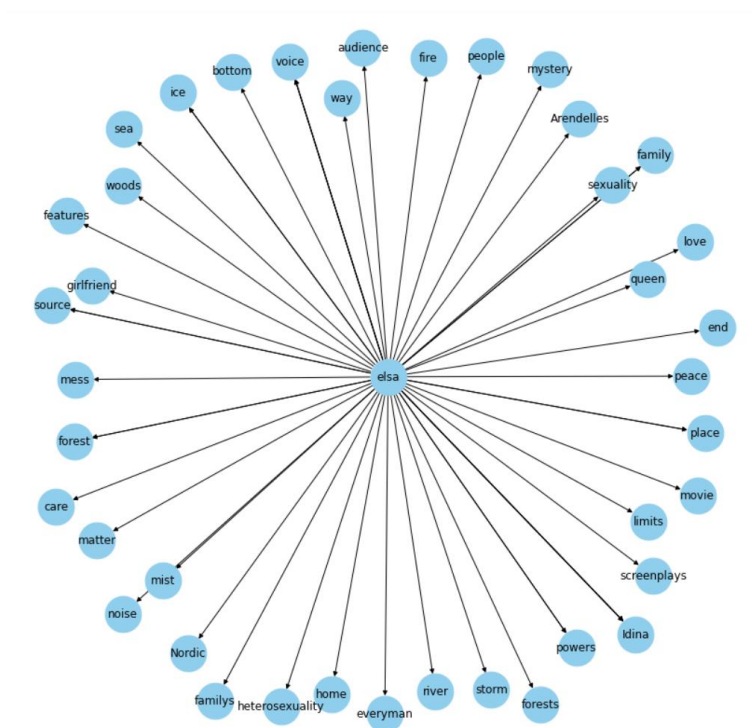

Isolatiing on relationship/verb "HAS"

Isolatiing on relationship/verb "WAS"

Ontology Plus Context and Modeling

Extracting Verb and Object for when Subject is Elsa. Elsa is the main character in the Movie

Isolatiing on relationship/verb "Voiced"

Ontology Plus Context and Modeling

Actress Idina menzel is the actress for ELSA in the movie

Part 5 LSTM Genre and Sentiment Classification:

The following 12 Experiments were done for classifying movie genre and sentiment.

Name	Classification	Vocabulary size	Type	Layers	optimizer	Regularization	Train Acc	Validation Acc	Test Acc	Test loss	Time
Model 1	Genre	1000	LSTM	1 bidirectional 64 LSTM layer , 1 Dense layer 64	Adam		97	75	63	0.625	2 m 33 s
Model 2	Genre	1000	LSTM	2 bidirectional 64 , 32 LSTM layer , 1 Dense layer 64	Adam	Dropout	94	88	83	0.63	2 m 16 s
Model 3	Genre	1000	LSTM	1 unidirectional 64 LSTM layer , 1 Dense layer 64	Adam	Dropout	55	45	46	1.4	1 m 33 s
Model 4	Genre	5000	LSTM	1 bidirectional 64 LSTM layer , 1 Dense layer 64	Adam		96	83	79	0.62	3 m 33 s
Model 5	Genre	5000	LSTM	2 bidirectional 64 , 32 LSTM layer , 1 Dense layer 64	Adam	Dropout	96	66	66	1.2	10 m
Model 6	Genre	5000	LSTM	1 unidirectional 64 LSTM layer , 1 Dense layer 64	Adam		76	66	33	1.4	3 m 22 s
Model 7	Sentiment	1000	LSTM	1 bidirectional 64 LSTM layer , 1 Dense layer 64	Adam		100	50	45	2.26	4 m 30 s
Model 8	Sentiment	1000	LSTM	2 bidirectional 64 , 32 LSTM layer , 1 Dense layer 64	Adam	dropout, L2	100	66	58	1.8	10 m
Model 9	Sentiment	1000	LSTM	1 unidirectional 64 LSTM layer , 1 Dense layer 64	Adam		90	50	41	0.83	1 m 42 s
Model 10	Sentiment	5000	LSTM	1 bidirectional 64 LSTM layer , 1 Dense layer 64	Adam		100	54	29	2.4	4 m 9 s
Model 11	Sentiment	5000	LSTM	2 bidirectional 64 , 32 LSTM layer , 1 Dense layer 64	Adam	dropout L2	100	44	41	2.12	6 m 9 s
Model 12	Sentiment	5000	LSTM	1 unidirectional 64 LSTM layer , 1 Dense layer 64	Adam		100	62	50	1.13	2 m 46 s

T-SNE plot of models Dense layer for movie genre classification
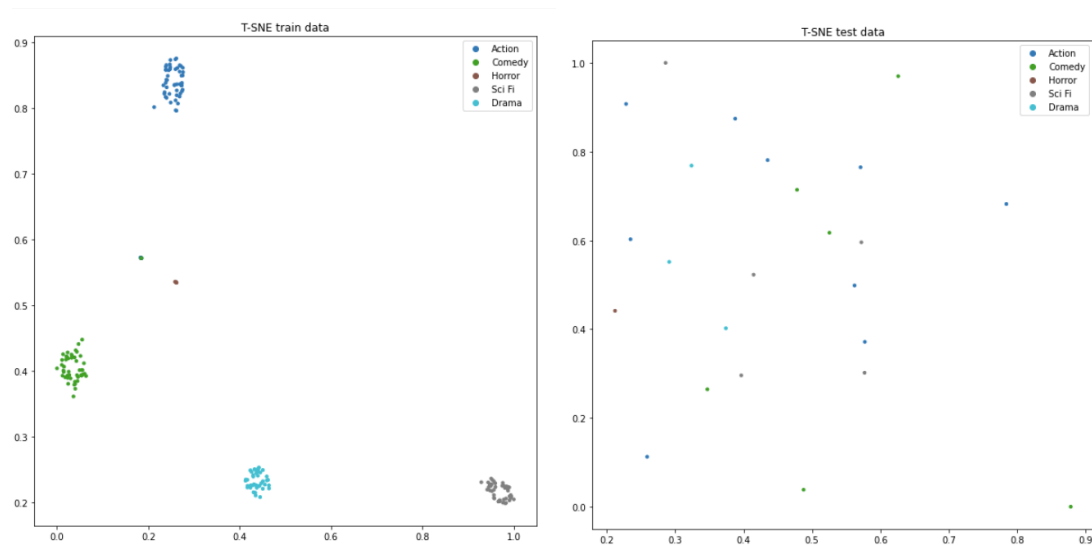
5. Analysis and Interpretation

Key Findings:

Clustering and Ontology.

Qualitative clustering was done based on key words. I identified 3 clusters (positive, negative and comedy/animation). The Ontology structure allowed for more extensive name entity relationship(NER) mapping, especially in Protégé. The Ontology created by me reflects my bias of choosing the NER. I already knew which reviews were positive or negative, which made it difficult to be unbiased. I feel without prior knowledge, I would not have been able to identify the correct sentiment for all the reviews as there was a lot of ambiguity in the reviewers' language.

Graph.

I experimented with preprocessing data to programmatically create a graph. Unfortunately, none of the methods worked. The pretrained spacy model could not accurately perform Name Entity recognition and could not identity all the equivalent classes and understand entity co-referencing. Isolating the top 5 relationships, we can see that most of the relationships don't make sense. The best of the worst relationship was "was". I also experimented with creating an SVO relationship by isolating on the subject. I got decent results when extracting relationships for the subject "ELSA". However, all the graphs created programmatically were lacking and are subpar compared to the one generated qualitatively.

LSTM Models:

I experimented with single layers and multi-LSTM bidirectional and unidirectional layer models. Overall, the bidirectional model performed better than unidirectional models. Having 2 LSTM layers performed better than a one-layer LSTM model. However, the improvement was not very significant. Vocabulary size had a bigger impact on performance. I found vocabulary size of 1000 performed better than a vocabulary size of 5000. A larger vocabulary size added noise to the data which made the model memorize the train data and meant that it could not generalize to the test data.

The best LSTM model for Genre classification received a test accuracy of 83%. These results, however, cannot be accurately evaluated because the test dataset was small, and the genre class was unbalanced.

The model is overfitting the train data which suggests that it is memorizing the data. The T-SNE plot shows that the model cannot discriminate clusters in test data.

Ontology Plus Context and Modeling

Name	Classification	Vocabulary size	Type	Layers	optimizer	Regularization	Train Acc	Validation Acc	Test Acc	Test loss	Time
Model 2	Genre	1000	LSTM	2 bidirectional 64 , 32 LSTM layer , 1 Dense layer 64	Adam	Dropout	94	88	83	0.63	2 m 16 s
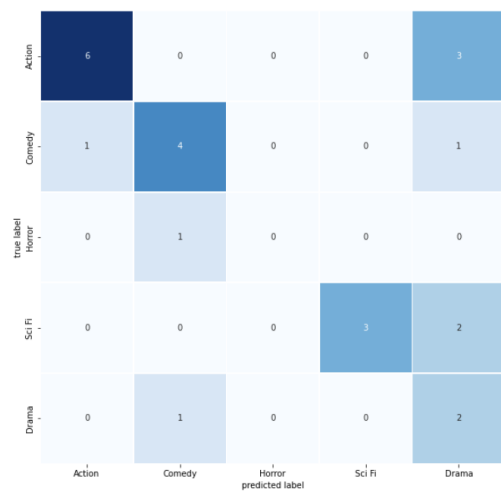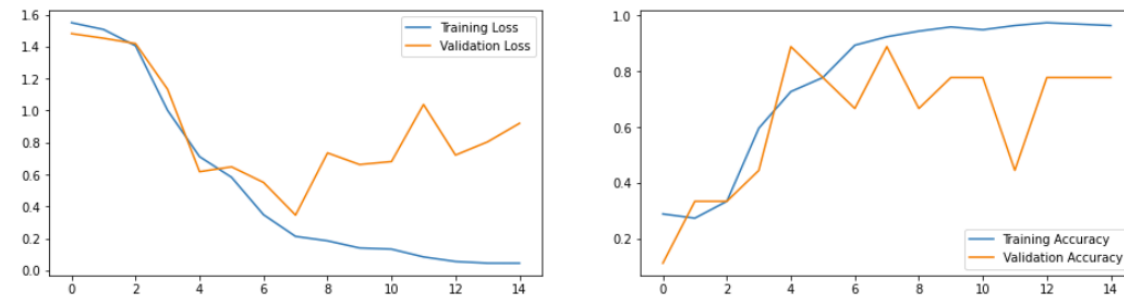

```
Classification Report
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         8
           1       1.00      0.83      0.91         6
           2       0.00      0.00      0.00         1
           3       0.75      0.50      0.60         6
           4       0.43      1.00      0.60         3

    accuracy                           0.79        24
   macro avg       0.64      0.67      0.62        24
weighted avg       0.82      0.79      0.79        24
```

The best LSTM model for sentiment classification received a test accuracy of 58%. The predictions are not better than a coin toss. For test data, the model predicts all reviews to be negative.

The model is overfitting the train data which suggests that it is memorizing the data and cannot generalize on test data.

Name	Classification	Vocabulary size	Type	Layers	optimizer	Regularization	Train Acc	Validation Acc	Test Acc	Test loss	Time
Model 8	Sentiment	1000	LSTM	2 bidirectional 64 , 32 LSTM layer , 1 Dense layer 64	Adam	dropout, L2	100	66	58	1.8	10 m
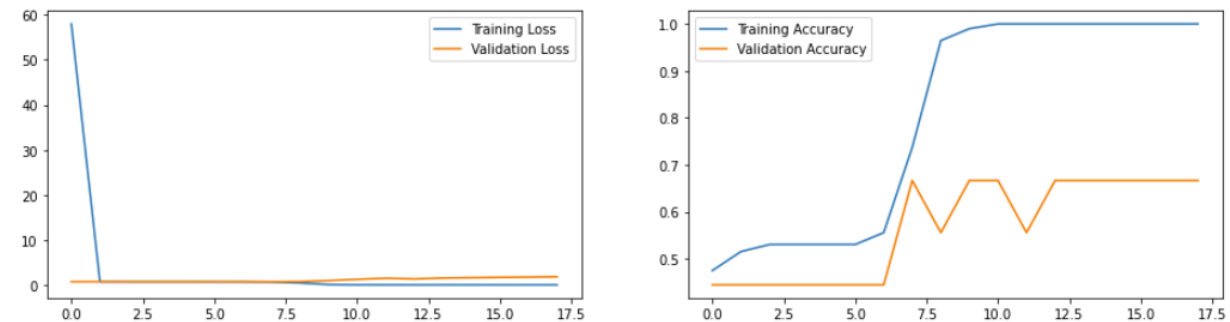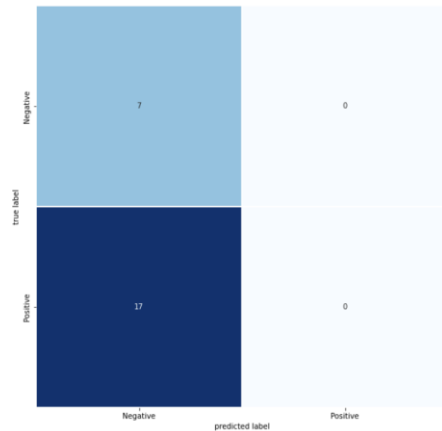
6. Conclusion

Qualitative clustering and ontology have some overlap and the clusters created are seen in the ontology graph. However, clustering and ontology have different objectives and are not completely related.

The pretrained spacy model that was used for programmatically creating graphs could not accurately perform Name Entity recognition and could not identify all the equivalent classes and understand entity co-referencing. To improve results in future, we would need to train a model on movie reviews to get a meaningful graph. The Manual method of creating a graph in power point or in protégé was significantly better in creating a knowledge graph.

Compared to power point, the Protégé graph provides the capability to add granular details in the ontology. It provides details on data types, relationship direction and joint and disjoint classes which cannot be easily incorporated by manual drawings.

Generally, the 2-layer bidirectional LSTM model with vocabulary size of 1000, performed better. This suggest that the longer sequence length require more LSTM layers to capture the contextual relationship in the data.

The LSTM model had some success in classifying Genre with approx. 80% accuracy for test data; however, the data set was small and not enough to properly evaluate the model results. All models performed poorly for sentiment analysis and the outcome was not better than a coin flip. The Confusion matrix for sentiment shows that the model predicts all reviews to be negative.

I found Vocabulary size of 1000 was the key to improving performance of all the models. A bigger vocabulary added noise in the data and because of it the model performed poorly on validation and test data.

The lack of data is an impediment for deep learning models, which require a large volume of data. The 249 reviews are not enough to train, test and validate a deep learning model. The Test data size was only 25 reviews which are not enough to evaluate confusion matrix and F1 scores for the 5-genre classification.

Appendix:

Appendix A
LSTM model summary example
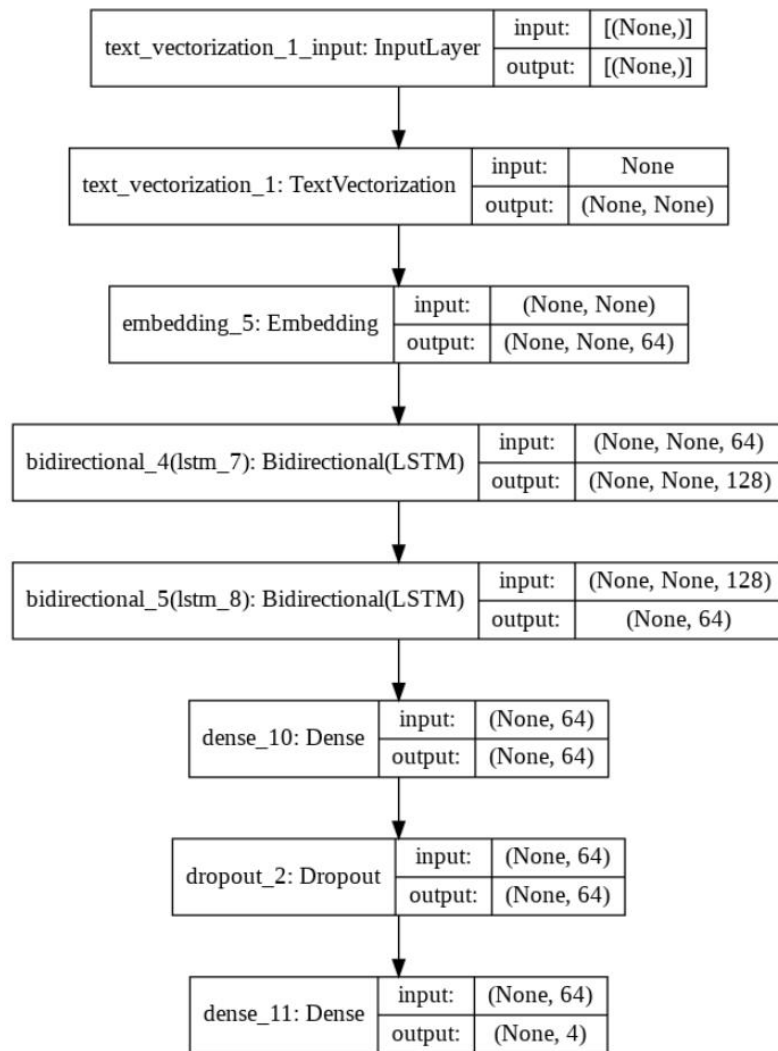
```
Model: "sequential_4"

_____
Layer (type)                 Output Shape              Param #
=================================================================
text_vectorization_1 (TextVe (None, None)              0

embedding_4 (Embedding)      (None, None, 64)          192000

bidirectional_3 (Bidirection (None, 128)               66048

dense_8 (Dense)              (None, 64)                8256

dense_9 (Dense)              (None, 4)                 260
=================================================================
Total params: 266,564
Trainable params: 266,564
Non-trainable params: 0
_____
```

Appendix B

Model with highest accuracy and f1 score