Assignment 3: Ontology Plus Context and Modeling

Vivian Xia

Northwestern University

MSDS453: Natural Language Processing

Syamala Srinivasan

February 21, 2022

Introduction & Problem Statement

The representation of text is the most important part of natural language processing. The issue is that often the text is not represented with enough meaning, which results in a lack of poor results in applications such as clustering and classification. The use of ontologies can help model the concepts and ideas behind the words and sentences in a structured way so that the relationships between the concepts can be observed. The goal is to build an ontology based on the topic of ten documents of *Guardians of the Galaxy* movie reviews. This ontology will help identify the classes or concepts and the instances of those. The relationships between different classes and instances can be defined to better understand the documents.

The objective of building an ontology will be accomplished by creating it manually, using the ontology editor Protégé, and using Python algorithms. This editor is a tool that makes creating ontologies and viewing relationships much easier. An ontology can also be created using the library spacy to create Python algorithms. These algorithms extract entity pairs and relations in order to create a knowledge graph showing the relationship between the entities. The knowledge graphs are another method used to observe the relationships between words to better understand the context behind the documents.

The entire corpus of 250 movie review documents, with half being positive reviews and the other half being negative reviews, is then used to create a deep learning model that can classify the documents into its sentiment class. A long short term memory model will be built and used to capture the sequential correlations, which will be stored in the memory to be used when iterating through the next time step. The goal of this model is to use the text to discriminate between the two sentiment classes.

Data

The packages os is used to define the path to the corpus of documents. The package time is used measure how long it takes each deep training model to process. The package packaging is used to check the verify the tensorflow and keras version to make sure that the models can run. The libraries pandas and numpy are used for formatting the data. The library spacy is used to extract entity pairs and relations to build the knowledge graphs. The library network is used to create the graphs. Matplotlib is used to visualize graphs including the directed graph and performance metric plots for the models. The tqdm package shows the progress of a process using a progress bar. Tensorflow, keras, and sklearn are used to build the model. The nltk library are used to clean the text of stop words for the deep learning models.

The corpus is made up of 250 documents with each document being either a positive or negative movie review. There are sets of 10 documents, 5 positive and 5 negative, that correspond to one of the twenty-five movies that make up the corpus. However, in this corpus, instead of twenty-five unique movies, there are twenty-four. There are two sets of movie reviews for the same movie, so one movie, *Red Notice,* has a set of 20 documents instead of 10.

Preprocessing of the ten documents is performed on the ten *Guardian of the Galaxy* documents for knowledge graph experiments. The subset of the data is loaded in from the corpus. The text is then split at the periods to break them up into sentences to add to the corpus for entity extraction. The first five sentences are observed. There are new line characters in the text, so those are removed by cleaning the text. The updated output of the five sentences shows that there are no new line characters and used for entity and relation extraction to create the knowledge graphs.

For the deep learning models, the entire corpus of 250 documents is used as inputs. The data is loaded and cleaned by getting tokens for the vocabulary. The special characters, digits

and numbers, stop word tokens are all removed. Lemmatization is also performed on the tokens to convert each token to its root word. The 2000 most common tokens are used to encode the text into token index sequences. Using one-hot encoding, the sentiment column of the dataframe is converted to 1's and 0's where 1 is positive and 0 is negative. Using the encoded texts and one-hot encoded labels, the data is then split into a training, validation, and test set with 168, 43, and 38 texts respectively. For the training dataset, there are 84 positive and 84 negative reviews, so there is an equal distribution of classes. The data is then used to model the classification of positive and negative sentiment.

Research Design and Modeling Method(s)

Part 1: Document Ontology and its Clusters

An ontology is a representation of knowledge in a specific domain. It defines basic concepts of a domain and its relations, allowing the identification of more broad or more detailed structures within a subject. The ontology provides a way to model the meaning behind each concept, so that the vectorizations can be restructured to a more meaningful representation, and in turn, achieve better results (Noy & McGuinness, n.d.).

The domain of this ontology is the representation of the 10 documents for *Guardians of the Galaxy*. To create the ontology, the application in Google Drive diagrams.net was used to represent the domain and its concepts. This application provides templates for different types of diagrams, in which a basic flowchart was selected. I first considered the classes I wanted to define in the domain. The ontology took several drafts as I considered the more abstract and specific classes and instances that should and should not be defined. And from the ontology, qualitative clusters can be observed that describe the ten documents.

Part 2: Protégé to develop ontology

Developed by Stanford University, Protégé is an ontology editor tool that builds ontology visualizations. The final draft ontology made in Part 1 will be built using Protégé. This tool has essentially four aspects that needs to be defined. The classes and subclasses are the first to be defined. From there, the relationships between the classes are identified in object properties where the property is defined as well as its domains and ranges. The data properties are identified as well by defining the domain that the data property is applied to, and the range is the data type. Lastly, the individuals are defined by creating its name and selecting the corresponding type and property assertions. The ontology is then visualized by the software using OntoGraf.

Part 3: Knowledge Graph experiments

Knowledge Graphs for the *Guardians of the Galaxy* documents are created by using Python algorithms. From the text, the entities are extracted by looking for punctuation, compound words, modifiers, and subjects. Each token in a sentence is checked to see if it is one of those mentioned types to find entity pairs. Two specified entities are extracted if they have semantic relationships. These pairings can be observed from the sample output. The relations are then extracted by matching a sequence of words based on patterns. These relations will be used to build the edges in the graph. The output of the relations and their value counts are observed. The subject and object are then extracted using the entity pairs found earlier. The first entity in the pair is extracted as the source or subject. The second entity in the pair is extracted as the target or object of the sentence. The edge found from extracting relations is matched with its corresponding pairs.

A directed graph is constructed from the subjects and objects where the subject is shown that it is related to the object. Another graph is plotted that uses a subset of the entity pairings

and corresponding relations to minimize the noise. This graph uses the entity pairings that use "is" as their edge since "is" was found to be the most common relation extracted from the sentences. Different distances between the nodes show differently organized graphs. The words in the dataset are then made into lowercase to observe if there are more intersecting relationships between nodes. Another dataframe is created by filtering the lowercase dataframe to relations of only "is," "are," and "has." The directed graphs using these other dataframes are plotted.

Part 4: Deep Learning experiments (LSTM)

Text is sequence data in that order matters, so the sequence of the words in a sentence or text is meaningful information and give context to the later words in the text. The model architecture of recurrent neural networks, RNN, is designed to memorize the past patterns of words as it predicts the current cell state. Long short term memory, LSTM, is a type of RNN that has both input and output gates, like the basic RNN, but also an additional gate, the forget gate. These gates divide memory into long and short term memory to address the issue of lost long term memory and vanishing/exploding gradients in simple recurrent neural networks (Srinivasan, 2022).

LSTM is used to classify the sentiment of the documents into its respective class. This model uses a Sequential class with an embedding layer, LSTM layer, and two dense layers. The embedding layer turns the words into embedding vectors by using a shallow neural network to give the words semantic meaning. The mask in this layer skips the padded iterations. A bidirectional wrapper is used with the LSTM layer to run two LSTM layers on the input where one layer reads the input from left to right and the other layer reads it from right to left. The input is processed forwards and backwards to extract patterns from both sequences, which processing only forwards may not have detected. The LSTM layer uses a ReLu activation function. The first

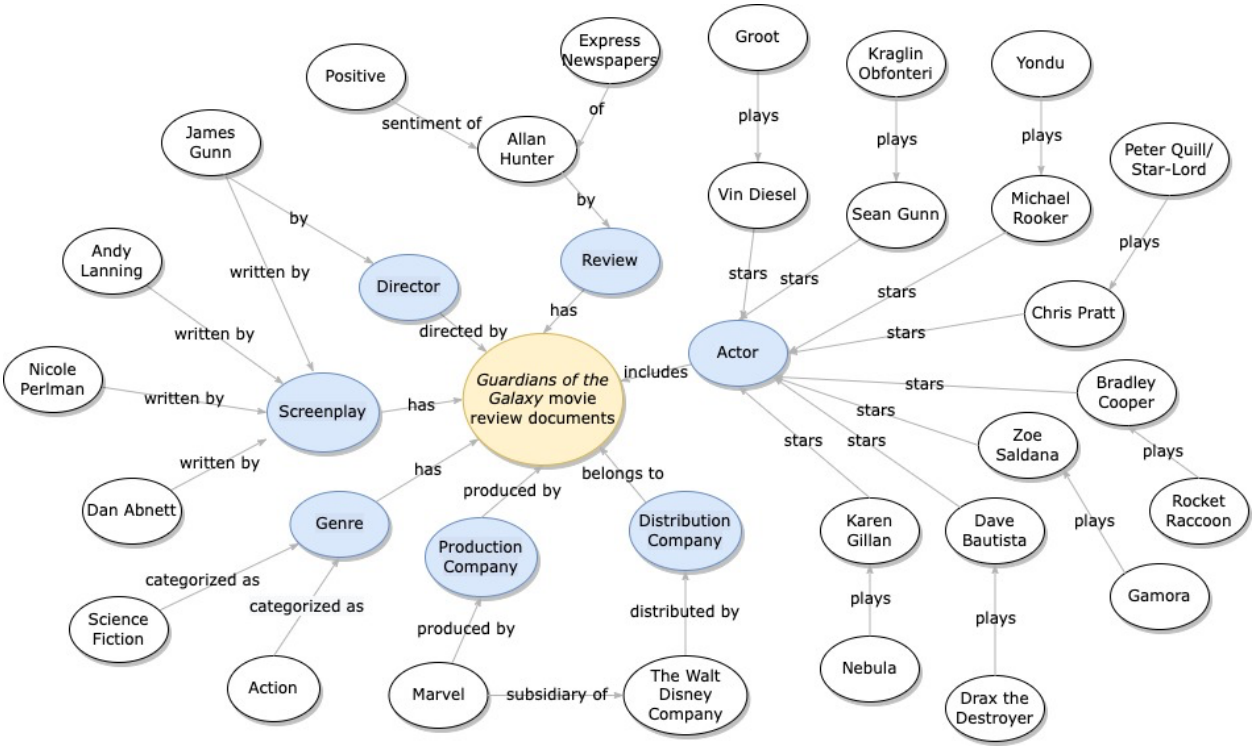
dense layer also uses the ReLu activation function to connect all the nodes from the LSTM layer to this layer's nodes. The next dense layer uses a SoftMax activation function to return values in the range of 0 to 1 as probabilities that each document belongs in which class. There are 2 nodes used to represent the two classes. The performance accuracy and loss of the model is then evaluated by a visualization. The vocabulary size of 2000 tokens will be held constant in the architecture. The same number of epochs and early stopping will also be used in every model.

Results

Part 1: Document Ontology and its Clusters

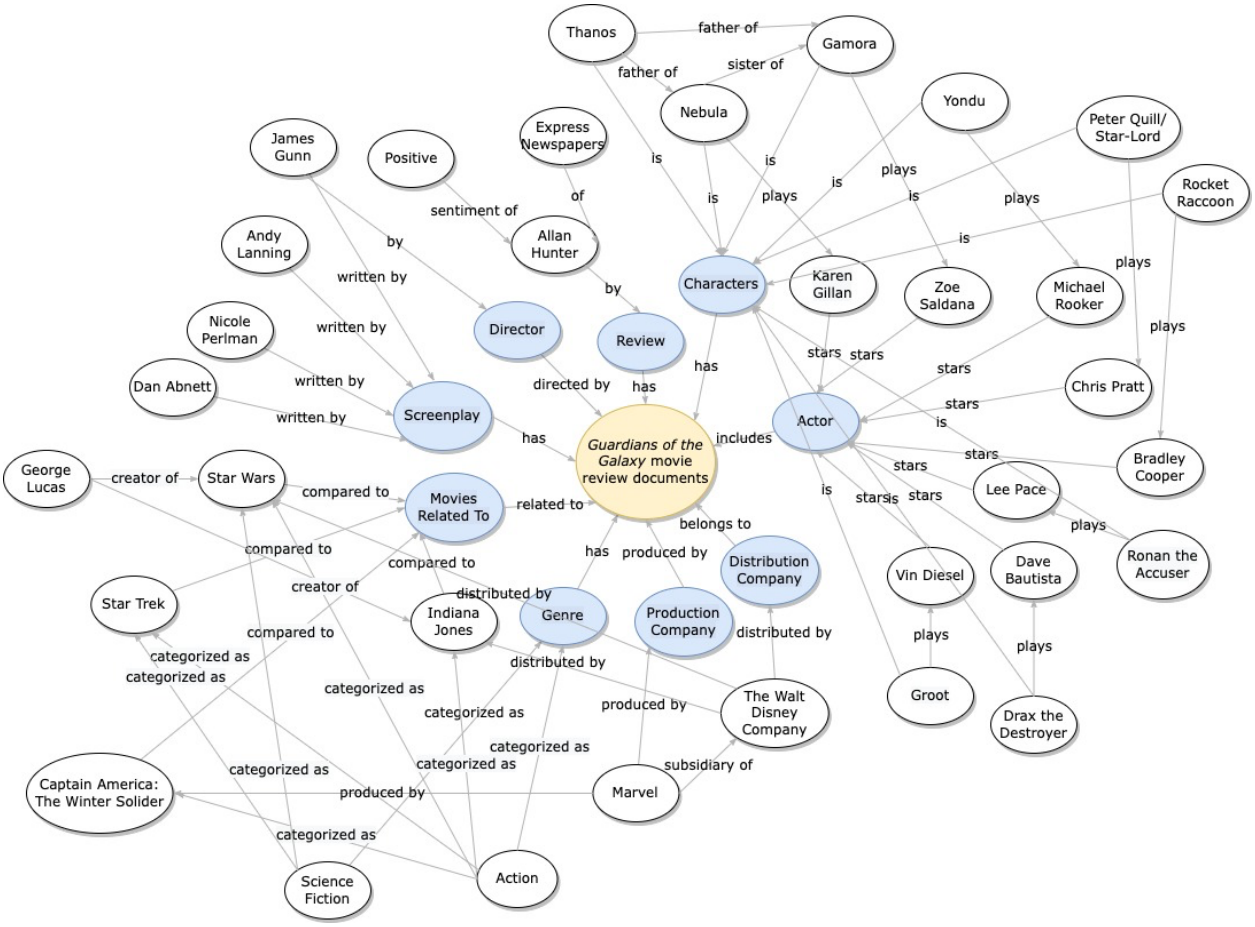
Figure 1 shows the first draft of the movie review ontology for *Guardians of the Galaxy* movie review documents. The blue shapes represent the classes, and the yellow shape represents the domain.

Figure 1. Ontology Draft 1.



The second draft of the ontology is shown in Figure 2. The Character and Movies Related To classes were added. Some characters and actors were removed and added. The movies from Movies Related To were added and their relationships to other classes and instances were identified.

Figure 2. Ontology Draft 2.



The Figure 3 shows the third draft with more detail into the Review class. The ontology becomes unreadable due to the many relationships that were identified despite only four reviews and their sentiment relationships being identified.

Figure 3. Ontology Draft 3.

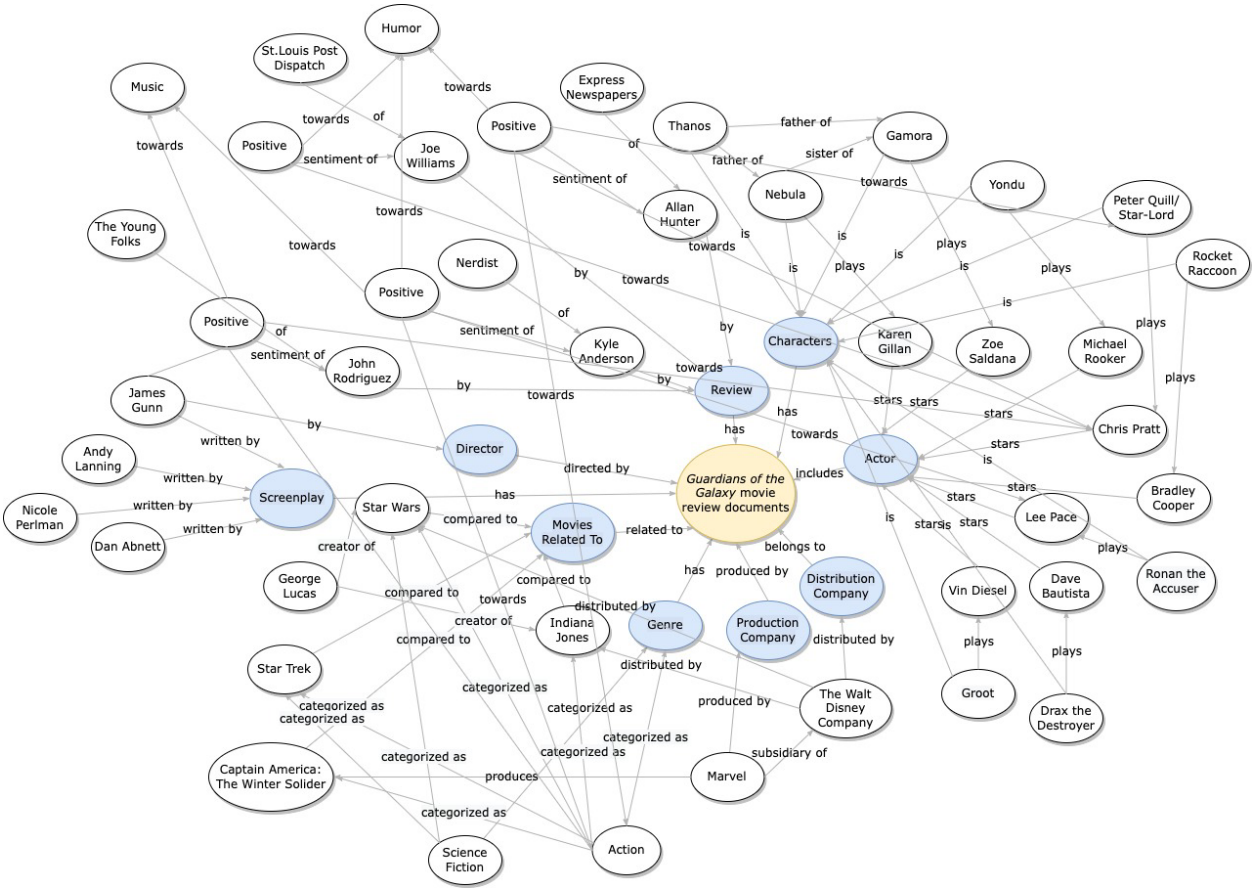


Figure 4 shows the fourth and final draft of the ontology. The relations that correspond to positive and negative sentiment were identified. The elements of the science fiction and action genre are also identified.

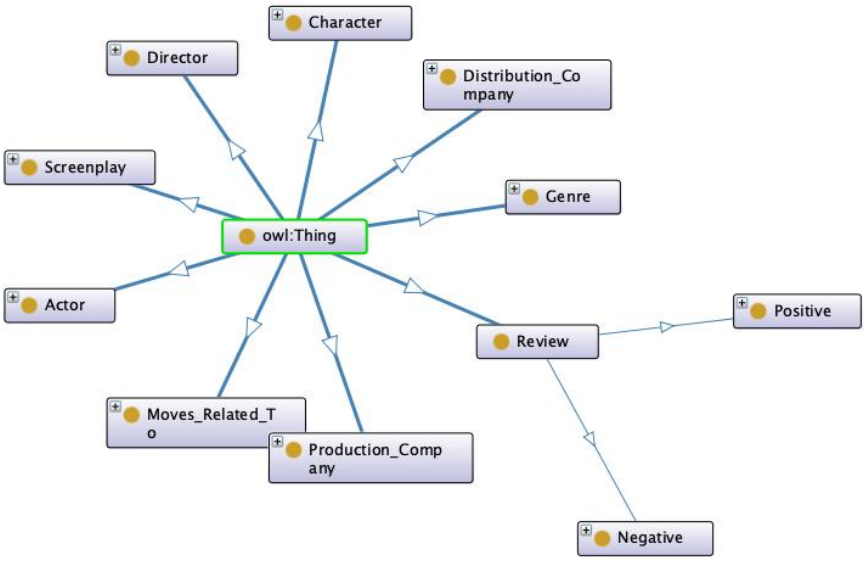
Figure 4. Ontology Final Draft.

There are four clusters that can be identified qualitatively from the ontology. The Character and Actor classes make up one cluster as people who appear in the movie. Another cluster is made up of Movies Related To, Genre, Production Company, Distribution Company classes because they are made up of documents that compare the movie to other movies based on the action and science fiction genre. The third cluster is made up of the Director class, Genre class, Review class, the concepts of Character and Screenplay, Chris Pratt, Peter Quill, and Rocket Raccoon. This cluster is defined by the positive and negative sentiment where the

relationships show the reasoning behind the corresponding sentiment for the documents. The last cluster is evident from breaking the third cluster into just positive sentiment and its relationships as well as negative sentiment and its relationships.

Part 2: Protégé to develop ontology

Figure 5. Protege ontology classes.



The classes that were defined in the ontology are shown in Figure 5. The figure shows that the classes are Screenplay, Director, Genre, Production Company, Distribution Company, Actor, Character, and Review. Review has two sub-classes of Positive and Negative.

Figure 6. Protege ontology Character and Actor classes.

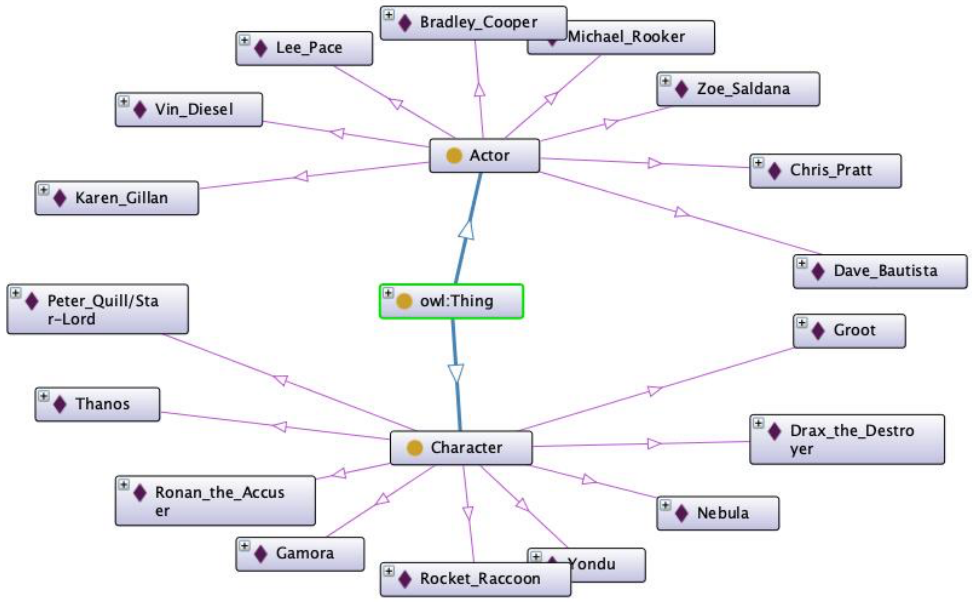


Figure 7. Protege ontology Character and Actor classes with its relations.

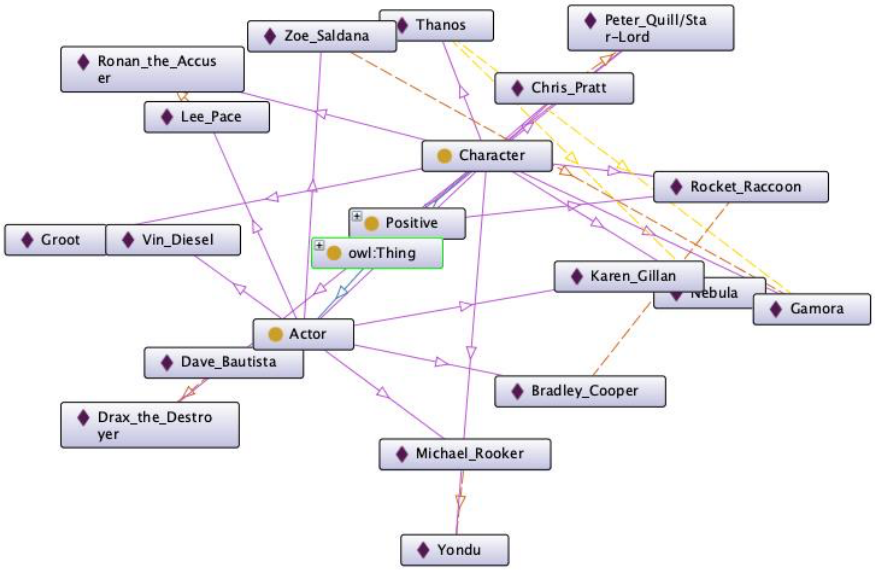
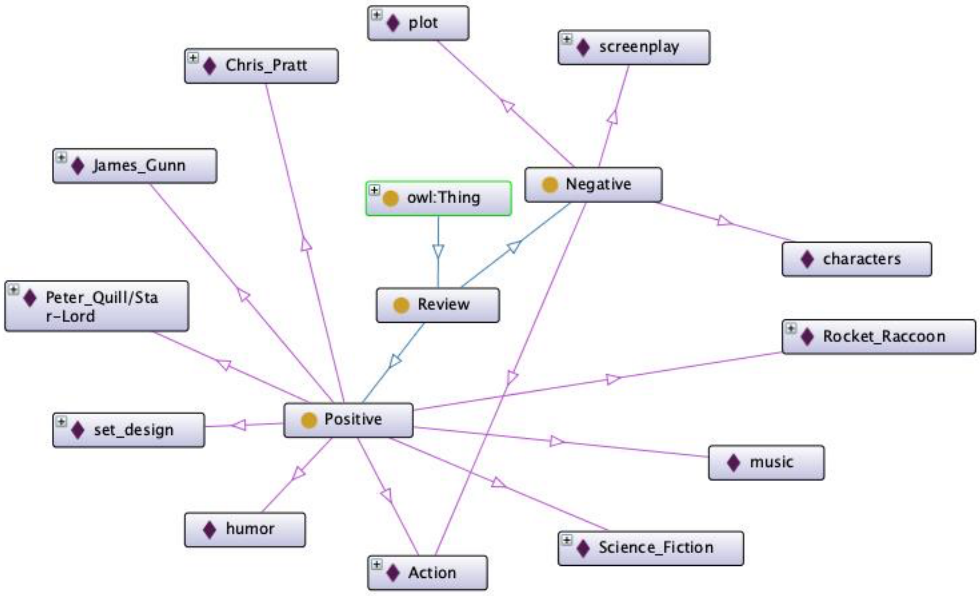


Figure 6 shows the individuals of the Character and Actor classes only. Figure 7 shows the ontology with all the relationships of all the individuals in the Actor and Character classes. The relationships between each actor and the character he/she played is shown as well as the relationship between the characters Gamora, Nebula, and Thanos. The sub-class Positive of the Review class shows up in this visual as well because many of the characters and one actor has a relationship with that sub-class.

Figure 8. Protege ontology Positive and Negative classes.



The Review class is shown in Figure 8 with its two sub-classes Positive and Negative to denote the two sentiments of the documents. Each sub-class has relationships with instances that factored into the positive or negative sentiment.

Figure 9. Protege ontology Positive and Negative classes expanded relations.

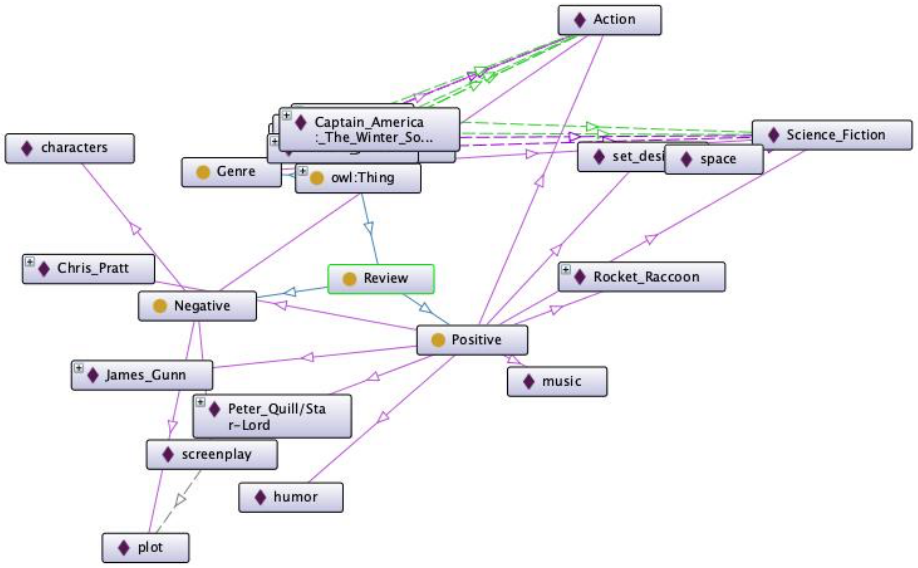
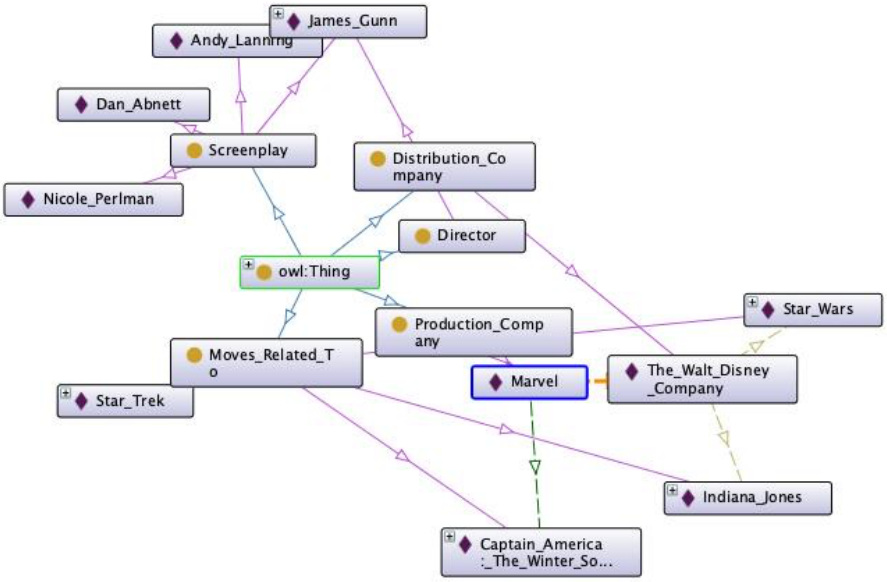


Figure 9 shows the expanded version of Figure 8's ontology with more details into Action and Science Fiction relationships as well as the relation between plot and screenplay and that of set design and space.

Figure 10. Protege ontology Production and Distribution Company, Screenplay, Director, Movies Related To classes.



The Screenplay, Distribution and Production Company, Movies Related To classes are observed in Figure 10. The relationships between the instances in those classes are defined.

Part 3: Knowledge Graph experiments

Knowledge graphs are constructed from the dataframe of subject, object, and relations. The directed graph is built from the subject and objects as seen in Figure 11. Figure 12 shows the knowledge graph without labels.

A graph containing only the edge word “is” and its corresponding entity pairs are plotted. The distance between nodes is set at 0.1 for the graph in Figure 13 and at 0.5 for the graph in Figure 14. Figure 15 shows the graphed subset with the relation “are” and distance between

nodes at 1. The subset with the edge word “has” is plotted in Figure 16 with distance between nodes at 0.8. Figure 17 shows the graph of the subset of “are” relations but with lowercased entities. Figure 18 shows the knowledge graph of the relations “is,” “has,” and “are” with its lowercased entities.

Figure 11. Directed graph using relation word “is” and its corresponding entity pairs at k=0.1.

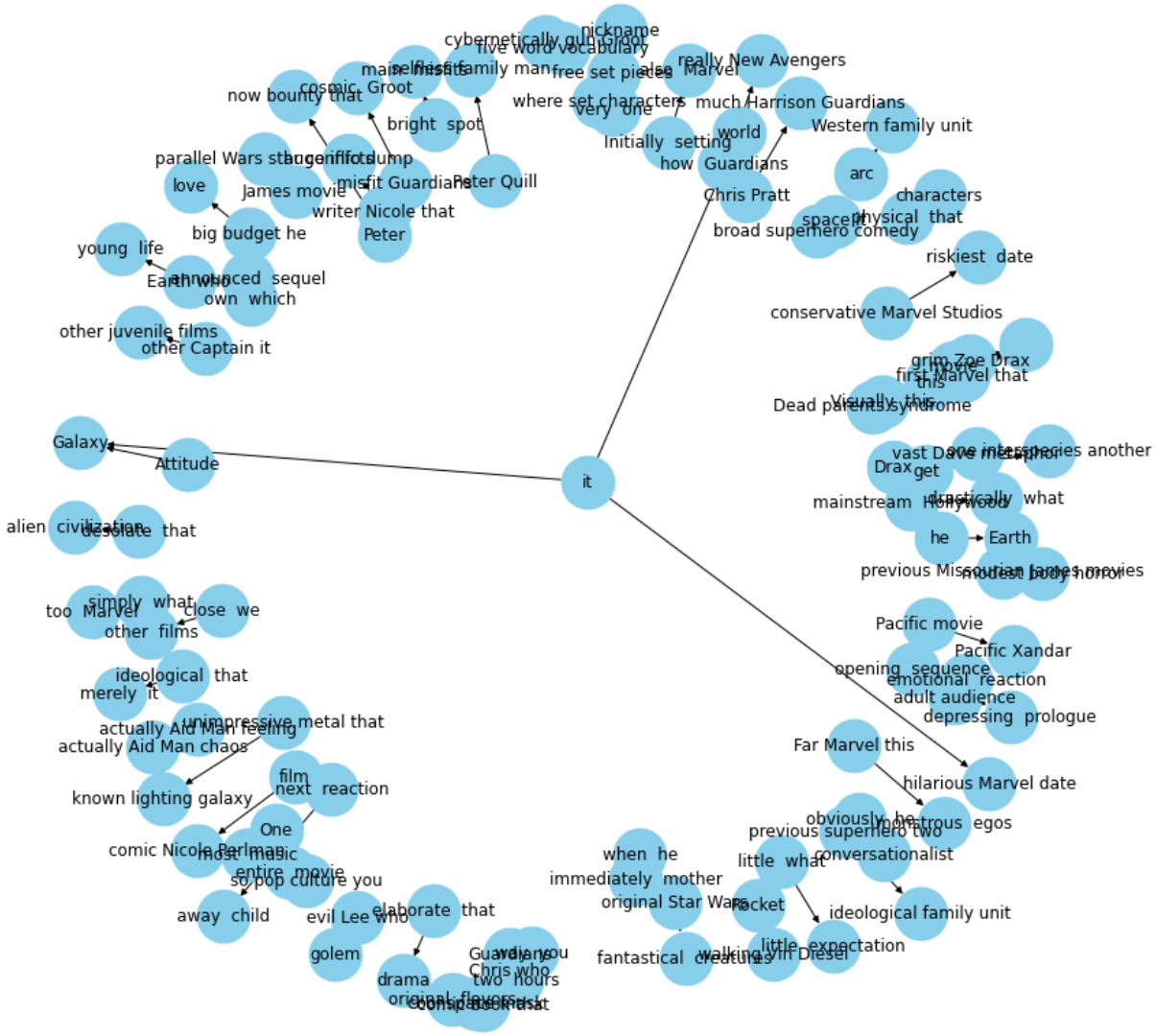


Figure 12. Directed graph using relation word "is" and its corresponding entity pairs at k=0.5.

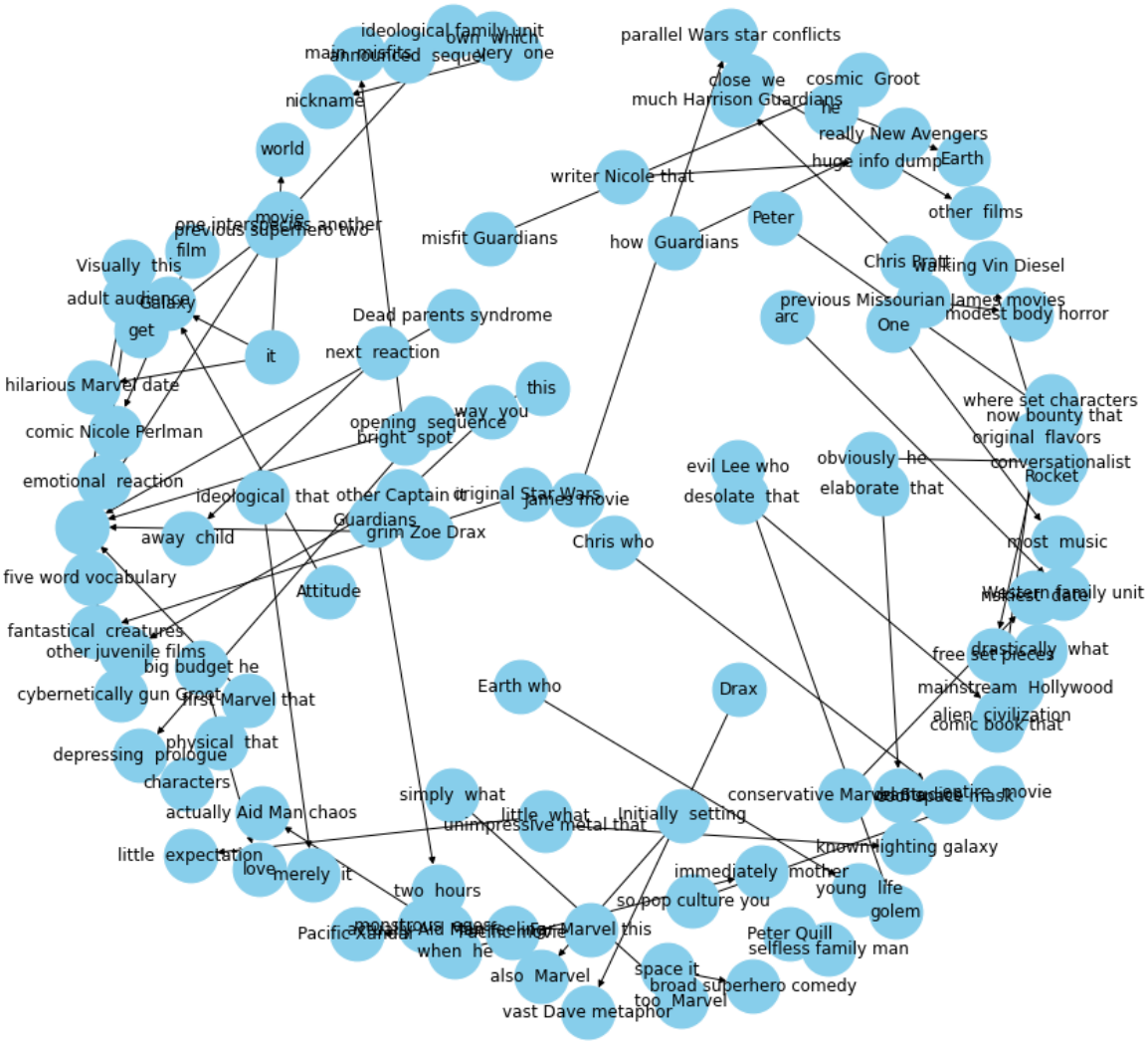


Figure 13. Directed graph using relation word "is," "are," "has" and its corresponding entity pairs at k=0.2 with lowercased entities.

Part 4: Deep Learning experiments (LSTM)

The models are built using an LSTM model architecture to classify the documents into their corresponding sentiment class of positive or negative. Each model's performance metrics are outputted as well as visualized in performance metric plots. The performance metric plots

show the accuracy and loss of the training and validation graphs throughout training. The performance accuracy of the experiments can be seen in Figure 19. From the performance, the model's hyperparameters are tweaked. There was overfitting indicated in the first three experiments. Regularization techniques such as L2 regularization, early stopping, and dropout were used throughout the experiments to prevent overfitting. The recommended model is Experiment 5's with one LSTM layer with a bidirectional wrapper and one node, a dropout layer of 0.3, a dense layer of 2 nodes and L2 regularization, another dropout layer of 0.3, and a dense layer of 2 nodes.

Analysis and Interpretation

Part 1: Document Ontology and its Clusters

Draft 1

The first classes I considered for the domain of *Guardians of the Galaxy* movie review documents included Review, Actor, Director, Genre. These initial classes were added to the ontology with relations to the domain. The Actor class was created and made up of Vin Diesel, Sean Gunn, Michael Rooker, Chris Pratt, Bradley Cooper, and Zoe Saldana, Dave Bautista, and Karen Gillan. These names include those that acted in the movie. I chose the actors that seemed like important characters including the main crew and a few minor characters that stood out in the movie. It was a challenge to decide which actors should be included and who should not be. At the next level of the actor names, I added the corresponding characters that each actor played. Chris Pratt's character goes by two names, Star-Lord and Peter Quill. I considered if I should put them as two separate entities or just one and decided on one as Chris Pratt is not playing both characters since those names both represent the same person that he is playing.

The next class I built was the Director class where James Gunn was identified. As I researched the director of the movie, I found that he also helped write the screenplay. Screenplay writers often write for movies with similar genres, so I added the Screenplay class to identify the screenplay writers. The screenplay writers were James Gunn, Andy Lanning, Nicole Perlman, and Dan Abnett. All these screenwriters also wrote for other movies under the same production company as this movie, Marvel. I added a Production Company class to identify Marvel. Because Marvel is a subsidiary of The Walt Disney Company, I also added a Distribution Company class to identify The Walt Disney Company's relations to movie and the production company. The Genre class and its subclasses of science fiction and action were defined. For the Reviews class, I referenced only one example of a review for the movie. I identified the author of the review as the subclass. On the next level, I identified the author's sentiment of the review and where the author posted the review.

Draft 2

A new class, Characters, was added to this ontology. The Characters class has relations to the already identified characters. Its relationships with who they are played by will remain.

As I was creating the last ontology, I realized I needed to focus more on the documents themselves rather than the movie. Because of that, I considered which actors and characters were defined in the documents themselves. The main crew that consists of Gamora played by Zoe Saldana, Peter Quill or Star-Lord played by Chris Pratt, Rocket Raccoon played by Bradley Cooper, Drax the Destroyer played by Dave Bautista, and Groot played by Vin Diesel. Ronan the Accuser played by Lee Pace is also mentioned in the multiple documents, so I added the actor and character into the class. Yondu was mentioned in three documents. The document does go into specifics on his place in the plot and relation to the characters, so I kept Yondu in. The

characters Nebula and Thanos were mentioned in two documents to point out their relationship with Gamora. I decided to keep Nebula and the actress who played her Karen Gillan in since the actress' name was mentioned in the documents. Thanos will be added as a character to show the relation between his character and Nebula and Gamora. The actor who plays him is not specified in the documents, so I will not add the actor's name to the ontology. Because Sean Gunn's character Kraglin Obfonteri was not mentioned in the documents, I removed him. The Collector and Prime Nova were mentioned in one of the documents, but there were no important relationships that were mentioned in the documents pertaining to them other than just a callout.

As I was looking over the documents, the screenwriters were all mentioned in at least one of the documents, which supported the importance of that class in the ontology. I also observed that the documents referred to many movies that were compared to this movie including *Star Wars, Star Trek, Indiana Jones, Captain America: The Winter Solider*. Because of this observation, I created a Movies Related To class. There were other movies that were mentioned such as *X- Men: Days of Future Past, Edge of Tomorrow, Snowpiercer, Dawn of the Planet of the Apes, Iron Man 2, Iron Man 3* but none of them were mentioned more than in one document. From the Genre class, I identified the relationships of the two genres science fiction and action to the movies in the Movies Related To class. *Captain America: The Winter Solider* is also produced by Marvel, so the relationship is identified there as well. Specific elements such as George Lucas the creator of *Star Wars* and *Indiana Jones,* was mentioned many times in the document, so I indicated the relationship between him and the films. These two films are also owned by The Walt Disney Company, so those relationships were identified as well.

Draft 3

The Review class is edited to give more detail into the reviews themselves. The review written by Allan Hunter is given more context into why that review is positive. Hunter likes the Chris Pratt's portrayal of Peter Quill as well as Peter Quill's character himself. He also likes the action and funny elements of the movie. The relationship between these elements were identified. Joe Williams had a positive review because he liked the science fiction and humor of it as well as Chris Pratt. Kyle Anderson likes the humor, Lee Pace's performance, music, action. John Rodriguez like the action, soundtrack, performance by Chris Pratt, and James Gunn. The ontology is looking very noisy at this point, so I decided to revise the I was identifying the reviews.

Final Draft

Instead of identifying the author and where the author posted the review, the Review class will only have the subclasses of positive and negative. The authors and the websites that the reviews were posted in does not appear in the actual documents so that information is not as relevant as the positive and negative subclasses. The disadvantage of taking those other subclasses out is that the number of authors that like a certain element in a positive or negative way cannot be seen.

As seen from the last draft, the positive sentiment reviews liked the humor, music, performance by Chris Pratt, the character Peter Quill, James Gunn, science fiction and action elements. The positive sentiment towards space, aliens, comic books are all labeled as science fiction elements. Similarly, battles and explosions are labeled as action elements. These elements of the two genres are identified in the ontology. Rocket Raccoon and the set design is also identified as a factor of positive sentiment. The set design mainly is due to the portrayal of the planets that exist due to the science fiction genre. The relationship between the set design and the

space element in the science fiction genre is identified. Despite the movie not being a comedy genre, many of the positive sentiment is due in part to humor.

There is negative sentiment towards the characters and their lack of personality, action sequences, screenplay, and the unoriginal plotline. The plot was created from the screenplay so that relationship was identified. Interestingly, there is no overlap of positive and negative sentiment on factors such as music, humor, set design, director James Gunn, and science fiction. There is overlap in characters and the action of the movie.

Qualitative Clusters

The final draft of the ontology is used to observe the clusters that take form to describe the documents. One cluster is made up of the Characters and Actor class. This cluster groups the people that appear in the movie together. The actors and their corresponding character names are often mentioned together in the same context in the documents.

Another cluster is made of up Movies Related To, Genre, Production Company, and Distribution Company classes. This cluster's key words are science fiction and action as most of the words are related to these two genres. The document that mentions the Movies Related To movies are often comparing *Guardians of the Galaxy* to those movies. In their comparison, they are focused on the genre that these movies are labeled as. For example, *Indiana Jones* is an action film that is mentioned to compare the action sequences with that of *Guardians of the Galaxy*'s. The exception is *Captain America: The Winter Solider* where the movie is compared because they are both Marvel movies, attributed to being in the same production company as *Guardians of the Galaxy.* The
distribution company is pulled in as well because of its relationships with *Star Wars* and *Indiana Jones* as well as Marvel.

The Review class, Genre class, Director class, Chris Pratt from the Actor class, Peter Quill and Rocket Raccoon from the Character class, and the concept of Character and Screenplay make up a cluster. The key words of the cluster are positive and negative because they relate to each of the other words. As mentioned above in the Final Draft analysis, the reviews with a positive sentiment refer to set design, humor, music, Chris Pratt, Peter Quill, Rocket Raccoon, science fiction, James Gunn, and action. On the other hand, the reviews with negative sentiment are related to the plot, the characters, action sequences, and the screenplay which relates to the plot. This cluster can be broken down into two clusters, one being defined by the positive sentiment and the other defined by the negative sentiment. The positive sentiment cluster would have its corresponding relationships and words in that cluster including set design, humor, music, action, science fiction, etc. The negative sentiment cluster would have its relations in its clusters including action, plot, characters, and screenplay.

Part 2: Protégé to develop ontology

To develop an ontology, the classes were first created to define the class hierarchy. The classes Screenplay, Director, Genre, Review, Character, Actor, Distribution Company, Production Company, and Movies Related To were created with a relation to the Thing, which is the documents for *Guardians of the Galaxy*, as shown in Figure 5. The Review class also has two sub-classes of Positive and Negative. I did not put science fiction and action as sub-classes of Genre because of the issue that the object properties do not allow me to define the relation between classes or subclasses and instances. To be able to identify the movies that are in Movies Related To and its corresponding genre(s), I added the genre types as individuals rather than sub-classes. This was one of the disadvantages of working with Protégé in that there are limitations

to what object properties can be defined, so planning does have to be done prior to creating. However, the software is very flexible, so that was not a very big issue.

One key advantage of working with this tool is that the classes that I want to view can be selected so that I can observe only those classes without all the noise that comes with the other instances and relationships. For example, I selected only the classes to be displayed to show the main point of the classes I created in Figure 5 rather than have those classes get lost in the noise of the instances. It also formats the graph automatically to best accommodate the positions of each class, individual, and relationship in the visual. This helps to minimize the noise and format it in a more organized manner. There are also choices on formatting the visual such as radial as seen in Figure 5 or horizontal tree, vertical tree, grid, etc.

For the Character and Actor classes, the relationships between each actor and character are shown in Figure 7. The software automatically groups the individuals that have relationships closer together, forming clusters. The relationship between the characters Thanos, Nebula, and Gamora are identified as well as Peter Quill/Star-Lord, Chris Pratt, and Rocket Raccoon with the Positive sub-class. The sub-class has a relationship with those three individuals, so it appears in this visual. It represents that there is positive sentiment towards those three instances.

For the Review class visualization, the Positive and Negative sub-classes are defined along with its relationships. Figure 8 shows that there was positive sentiment towards individuals such as James Gunn, Chris Pratt, Peter Quill/Star-Lord, set design, humor, Action, Science Fiction, music, and Rocket Raccoon. There was negative sentiment from the plot, screenplay, characters, and Action. There is overlap, as seen in the visual, where there was both positive and negative sentiment towards Action. For screenplay, I wanted to specify the relationship of negative sentiment towards the class Screenplay as I did in the manual ontology, but as

mentioned above, the tool does not allow relations to be defined for classes and individuals. Instead, I created the instance of screenplay instead.

Figure 9 shows the expanded version of the Review class relationships. The instances plot and screenplay are related as the screenplay creates the plot. Similarly, the set design is of space, so the relationship is defined between them. The Action and Science Fiction instances have relationships with other movies in those genres such as *Indiana Jones, Star Wars,* etc. as well as its elements such as alien, planet, battles, and explosions. The software formed a cluster to show that those instances all have relationships with the Genre class.

The ontology with Screenplay, Distribution and Production Company, Director, and Movies Related To classes in Figure 10 show the relationships between their instances. This visual is simpler than the other ones looked at since there are not as many relationships. The clusters show that the relationships exist mainly among the three movies *Star Wars, Captain America: The Winter Soldier,* and *Indiana Jones*. These movies are related because of the same distribution companies. Overall, Protégé made creating an ontology much easier than manually constructing it. The software is straightforward and allow easy editing of classes, relationships, and instances. The visualization aspects of the software also make analysis simpler because it reduces the noise to allow the desired class and instances to be observed rather than everything at once.

Part 3: Knowledge Graph experiments

The entity pairs extracted from the consideration of their semantic relationship within a sentence. A few of these pairings are outputted to analyze their relationships. There are some pairings' relationships that are vague, but some do make sense together. The pairing "I" and "forward it" where "I" is the author of the review and "forward it" has a positive connotation

because of the word “forward.” This pairing gives the relationship that the author had positive sentiment towards the movie. Another meaningful pairing was “Together they” and “intergalactic A heroes.” These two words refer to the main characters, as a whole, as space heroes, which accurately described the crew in the movie. Some vague pairings include “later Peter” and “iconic Cherry Feeling” as well as “then A list I” and “sonic screwdriver.” The entities in these pairings do not have an obvious relationship to each other. There are also some pairings with one or no entities.

The sample output of the extracted relations show that most of the relations that were extracted were “is,” “are,” and “has.” These words do not hold much meaning within them. There are some relations that do not seem like relations such as “Stifle” and “’m.” There are more meaningful relational words as well including “start” “interactions between,” “provides.” The dataframe of the matched source, target, and edge words or phrases is outputted. Most of the pairings between the three words/phrases do not make much sense together. Some have ambiguous entity pairings, relation, or both such as “I,” “it,” and “work” where every word is vague, respectively. There are some pairings that vaguely make sense together such as “space it,” “broad superhero comedy,” “is.” The pairing identifies that the movie is a space superhero comedy. Another one that makes sense is “ideological that,” “merely it,” “is.” It gives a negative connotation from the two entities, and the relation identifying that the first entity is related to the second.

The directed graph is created from the source and target. The graph in Figure 11 is noisy due to the number of nodes and its labels. The labels are removed in Figure 12 to observe the structure more easily between the nodes. However, this is not an ideal graph either since the labels are not there to identify the relationships. Most of the relationships between the nodes are

located relatively close to each other, making up the perimeter. The nodes clustered in the middle include “Zoe,” “Drax,” “when he,” “screenplay,” “best James Gunn,” “Chris Pratt,” “I,” from the words and phrase that can be deciphered. These words have some relationship with “Chris Pratt,” “Zoe,” “Drax” being characters or actors in the movie. “James Gunn” also wrote the “screenplay.” The other words are vaguer, which makes it harder to make a connection from the other entities in that cluster.

A subset of the directed graph is taken by plotting only the entity pairs with the relation of “is.” The word “is” was the most common relation extracted at 61 counts. The distance between nodes is set at 0.1. It can be seen that “It” is used as the subject three times with the objects being “Galaxy,” “hilarious Marvel date,” and “world.” The “it” refers to the movie, so these objects do make sense in being related to the movie. The “is” relation identifies that “it” as the same thing as those objects, so those objects describe the movie. The object that gives the most information is “hilarious Marvel date” because it shows that it was funny and a Marvel movie. The object “Galaxy” is vague because the capitalization of the word shows that it was taken from the title, so it may be referring to the movie title rather than the concept of the space galaxy. The subject “Attitude” is also paired with the object “Galaxy.” This relationship shows that there is attitude in the movie but does provide any context into if it is good or bad. The subset taken using “is” as the relation with 0.5 distance between the nodes show that there are a few entity pairings with no object. This is observed by the subject nodes and their directed graphs towards a blank node. This graph clusters the ‘it” subject with its corresponding objects to one side of the perimeter with other nodes such as “Visually this,” “film,” “previous superhero two,” “movie,” “world,” “adult audience,” “get.” Those entities together along with “it,” “Galaxy,” “hilarious Marvel date,” “world,” make some sense together. The entity “it” is the

movie which is the same as “film,” “movie.” The other words point out specific aspects of the movie such as “Visually this,” “previous superhero two,” “adult audience.” These words refer to the visual, superhero, and adult elements in the movie.

The word “are” was the second most common relation at 14. The 14 entity pairings are plotted. “They” and “they” are separate nodes that most likely refer to the same thing. There is a blank node that acts as a placeholder for one object and one subject. There is no intersection of relationships other than at the blank node. With the subset using the relation “has” which has 9 pairings, there are only straightforward relations from one subject to its corresponding object.

A dataframe with lowercased entity and relation words is created from the original dataframe. As seen by the “are” relation subset plot, there are separate nodes for an uppercase and lowercase version of the same word. The subset of entity pairings with the relation “is” is plotted at a distance of 0.1. There does not seem to be a difference in this graph than its previous counterpart. There are still only three “it” subjects and similar relationships as the counterpart. The “are” subset is graphed. The “they” is a subject and object in this graph. It is the subject for “still it” and object for an empty node. Otherwise, there are no other intersection or chain of relationships.

From the lowercased entity dataframe, a new dataframe was filtered to only the word relations of “are,” “is,” and “has.” The plot shows that there are a few interactions with the pronouns “it,” “this,” and “they.” The subject “it” is still used only three times. Similar to the other “are” subset graphs, “they” is used twice. The subject “this” and “he” are also used twice. Overall, there are only multiple relationships for more general words.

Part 4: Deep Learning experiments (LSTM)

Experiment 1

The first model uses a 2000 token vocabulary with one LSTM layer with 20 nodes. The next layer is a dense layer with 15 nodes and an activation function of ReLu. A dropout layer is used to use drop a random 0.3 nodes each iteration to help with overfitting. The last layer is a dense layer with 2 nodes, representing the 2 classes. An early stopping callback is also used for regularization by stopping training when the validation accuracy stops improving after two epochs. The test accuracy is very bad at 0.37. The validation and training accuracy are also not similar to each other at a 0.05 difference. The performance plots in Figure 20 also show that there is overfitting because the training and validation graphs are going in opposite directions.

Experiment 2

This model uses a reduced model complexity as compared to Experiment 1's architecture. The LSTM and dense layer take 5 nodes. The dense layer also has L2 regularization to help with overfitting. The test accuracy improves with a less complex model. However, the test accuracy of 0.42 is still not good. It does worse than classifying randomly at 0.50. The performance plots, Figure 21, show overfitting as well because the training and validation graphs are going in opposite directions.

Experiment 3

The model hyperparameters are further reduced to combat overfitting. The LSTM layer has 1 node, and the dense layer has 2 nodes. Otherwise, the architecture is the same as Experiment 2's. The test accuracy improves to 0.63, which is better than randomly guessing. However, the test accuracy is much better than either the training or validation accuracy. The gap between the training and validation graphs are still evident in the performance metric plots in Figure 22, but they are not as extreme as before. There is still overfitting due to the gap between them, but they are both going in the same direction.

Experiment 4

The model architecture stayed the same as Experiment 3's but another dropout layer was added between the LSTM and dense layer to prevent overfitting. The dropout layer helped because the gap between training and validation accuracy is very small. The test accuracy is also closer in value to the other two scores. However, the test accuracy score is 0.45, which is not good. The performance metric plots in Figure 23 show that the validation and training graphs are going in the same direction. The y-axis scale for the loss plot is in small increments, so the difference in loss value is less than 0.01.

Experiment 5

This model uses the same architecture as Experiment 4's model except there is a bidirectional wrapper around the LSTM layer. The training and validation accuracy are still pretty close in value, so there is no overfitting. The test accuracy also improved from the preceding experiment, so the bidirectional wrapper did extract temporal correlations that just going forwards did not extract. The test accuracy is better than random. The performance plots, Figure 24, support that there is no overfitting.

Experiment 6

This model increases the complexity of Experiment 5's model. The LSTM layer has 2 nodes, and the dense layer has 4. The resulting test accuracy is not good at 0.39. The model does not overfit because the training and validation accuracy are close to each other. The loss plot, Figure 25, shows that the training and validation are going in the same direction and close in value. The accuracy plot shows that the validation accuracy flattens while the training accuracy falls.

Summary

The best model in terms of test accuracy is Experiment 3 consisting of a LSTM layer with 1 node, a dense layer with 2 nodes with L2 regularization, and a dropout layer. However, this model overfits as seen by the training and validation accuracy difference as well as the test accuracy difference between the other two accuracy scores. The model from Experiment 5 uses the same architecture as Experiment 3 but uses a bidirectional LSTM layer and another dropout layer. The model does not overfit, and the bidirectional wrapper helped in extracting relevant patterns from the sequences. Overall, the model from Experiment 5 is the best model of the models tested in classifying the documents into its sentiment class.

Conclusions

An ontology provides information based on the different concepts that are related to the domain. They can be created in multiple ways including manually, through Protégé, and with Python algorithms. Each method displayed relationships to the topic, providing more context into the documents and their common themes and how those relate to each other. For example, the observations on the relationships of the instances for positive and negative sentiment provided context into which areas most authors had positive or negative sentiments about. Knowing this, the methods in which to classify the sentiments could focus on those words in each document. The use of knowledge graphs also provided a way to observe entity relationships where the documents had a few entities in common. This relationship helped cluster some of the entities together in the knowledge graph. The deep learning model provided another method in which to classify the documents into their corresponding class. An LSTM architecture was used to keep the temporal correlations of the sequence of words in its memory so that it can provide context into the predicted output.

Directions for future work

From the knowledge graphs, there was the issue of coreference that needed to be addressed. There were many words that referred to the same entity but were represented as separate nodes such as “it,” “film,” and “movie” as well as “them” and “they.” It can be inferred that “it,” “film,” and “movie” all refer to the movie itself and its contents, experiments can be run to see if this assumption is true and which relationships are identified if all three of those words are changed to one form of the word. Similarly, “they” and “them” most likely refers to the main crew of characters, so those two words are also coreferences. The ontology created manually and with Protégé show that Star-Lord and Peter Quill are the same person, so any combination of Quill, Star-Lord, Peter, Peter Quill can be combined into one form. By resolving these coreferences, the relationships between entities and concepts may provide more context to the documents.

The deep learning network could not extract more complex abstractions due to the issue of overfitting when the model is too complex. A possible solution is to get more movie review documents so that the model has more to train with. A different sized vocabulary or more preprocessing of the tokens can also be experimented with to improve model results.

References

Noy, N. F., & McGuinness, D. L. (n.d.). *Ontology Development 101: A Guide to Creating Your First Ontology*. Retrieved 2022, from https://protege.stanford.edu/publications/ontology_development/ontology101.pdf

Srinivasan, S. (2022). *MSDS 453 Natural Language Processing: Week 7 – Entity Coreference Resolution & Deep Learning* [Zoom Cloud Recordings].

Appendix

Figure 14. Directed graph using source and target with labels.

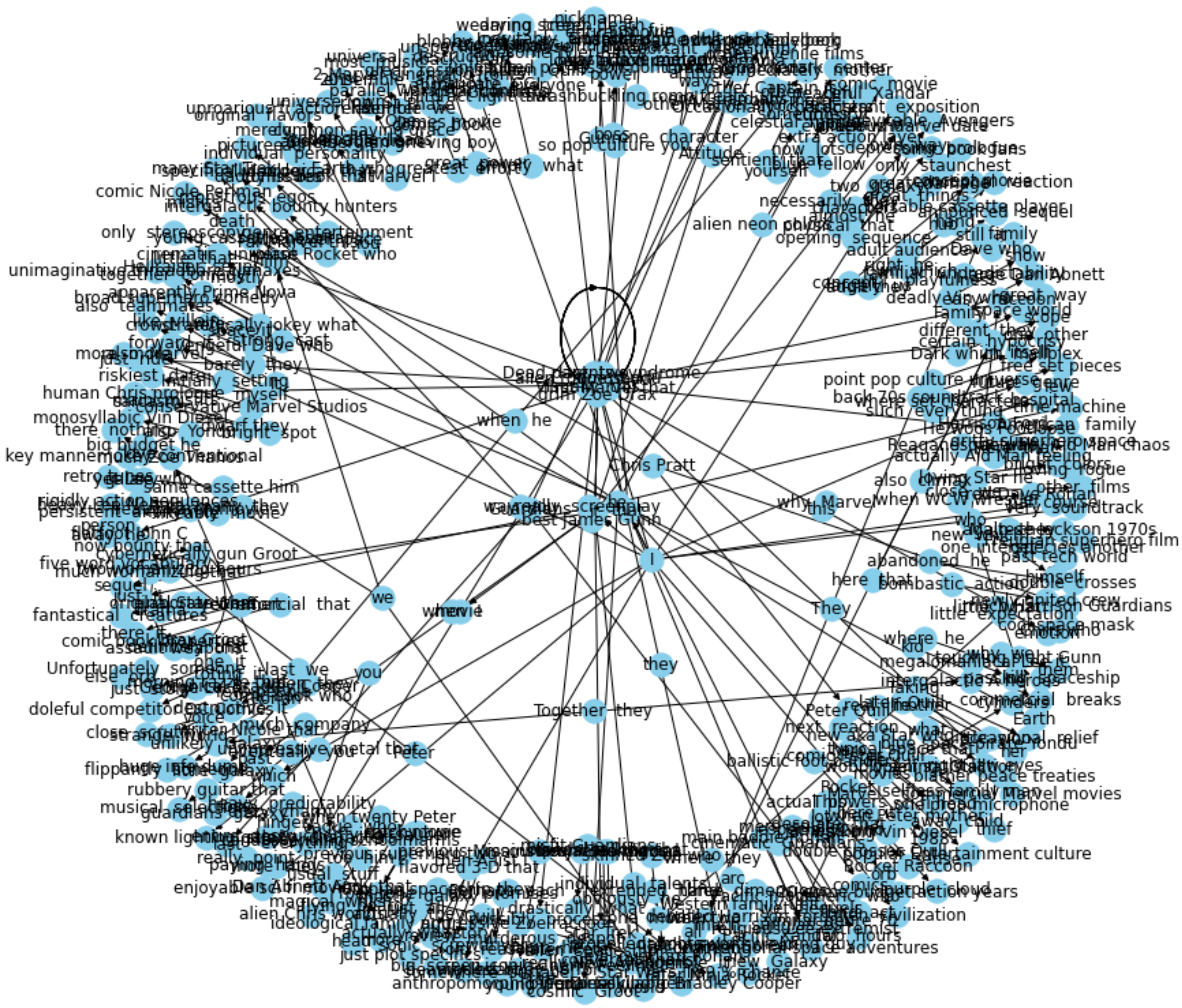


Figure 15. Directed graph using source and target without labels.

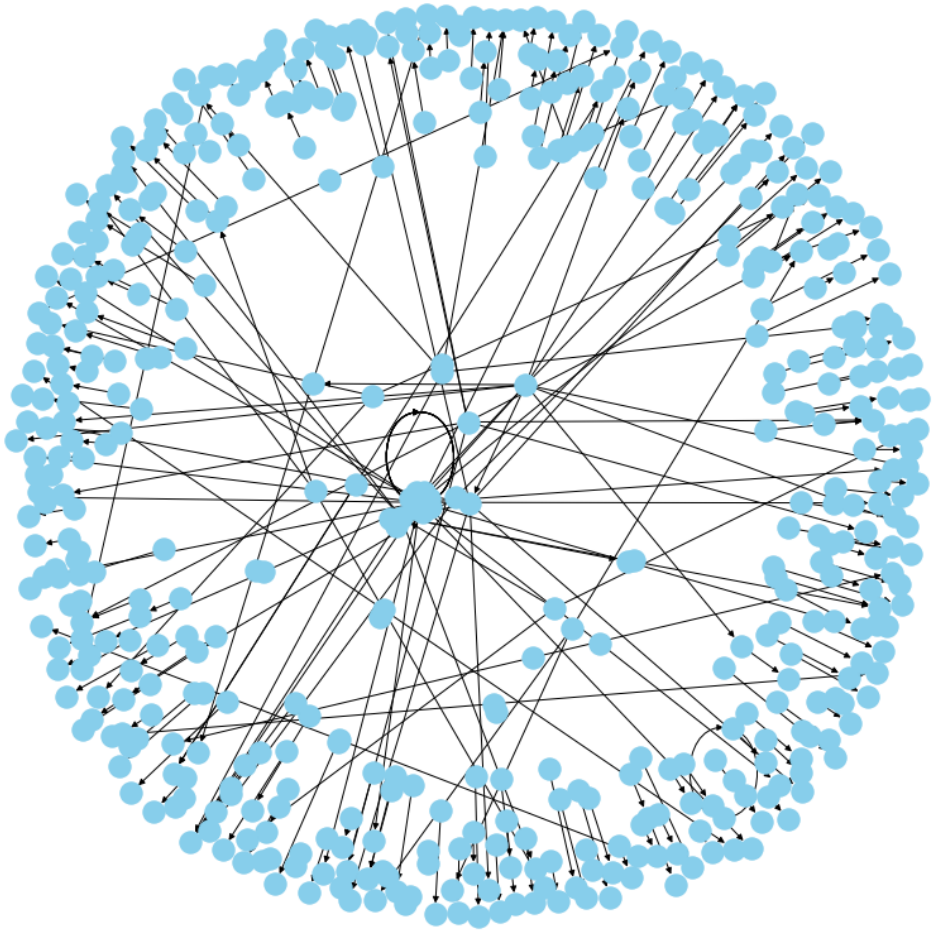


Figure 16. Directed graph using relation word "are" and its corresponding entity pairs at k=1.

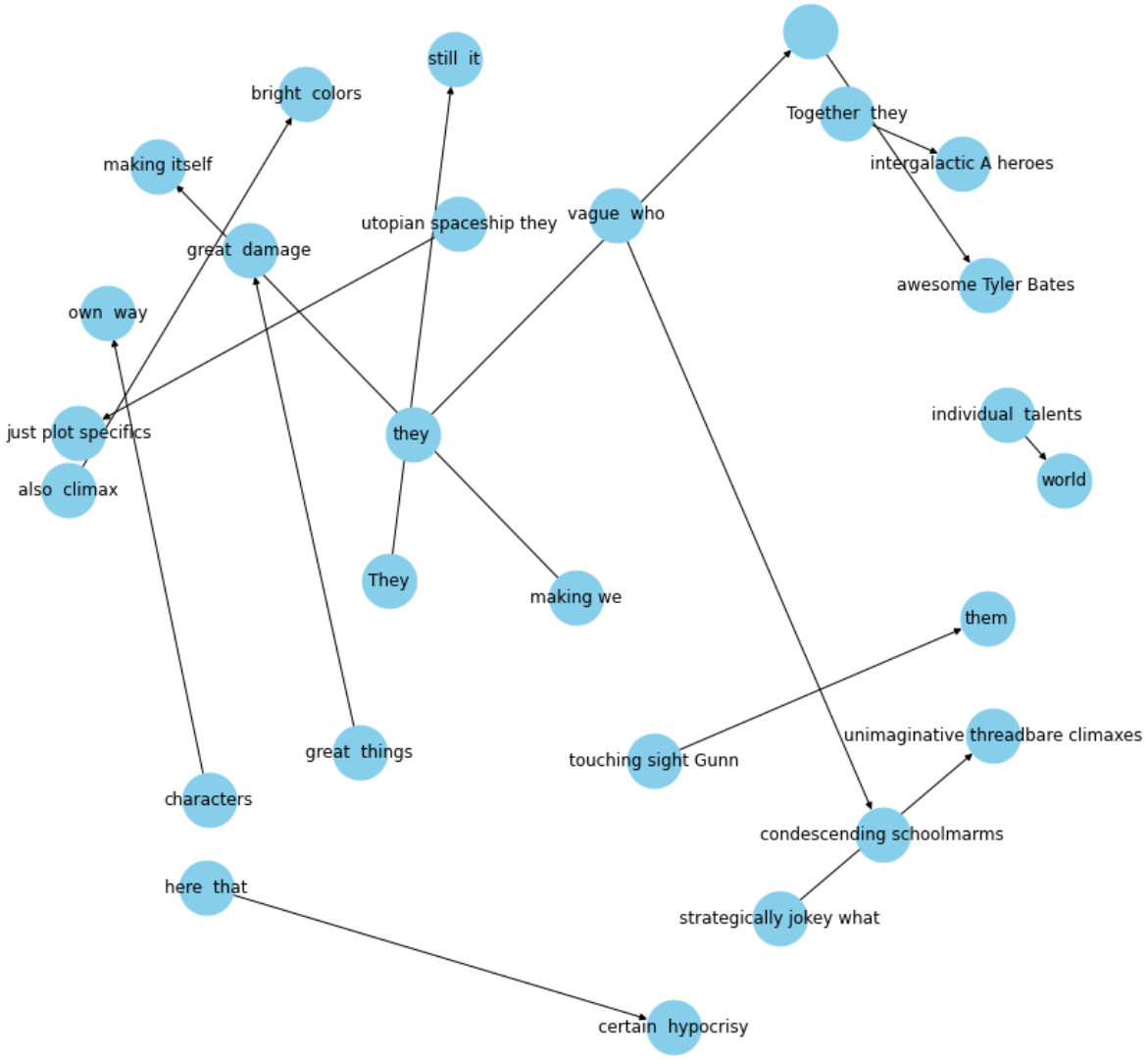


Figure 17. Directed graph using relation word "has" and its corresponding entity pairs at k=0.8.

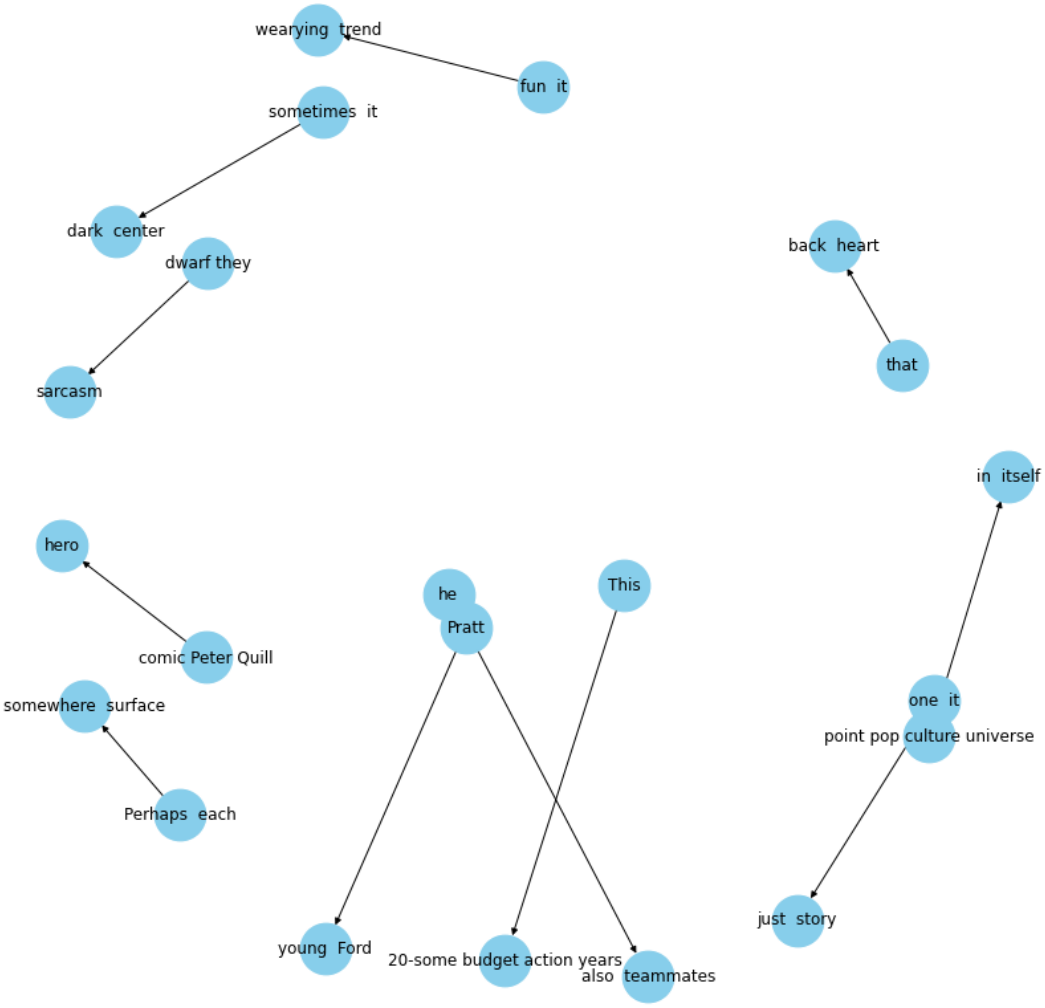


Figure 18. Directed graph using relation word "are" and its corresponding entity pairs at k=1 with lowercased entities.

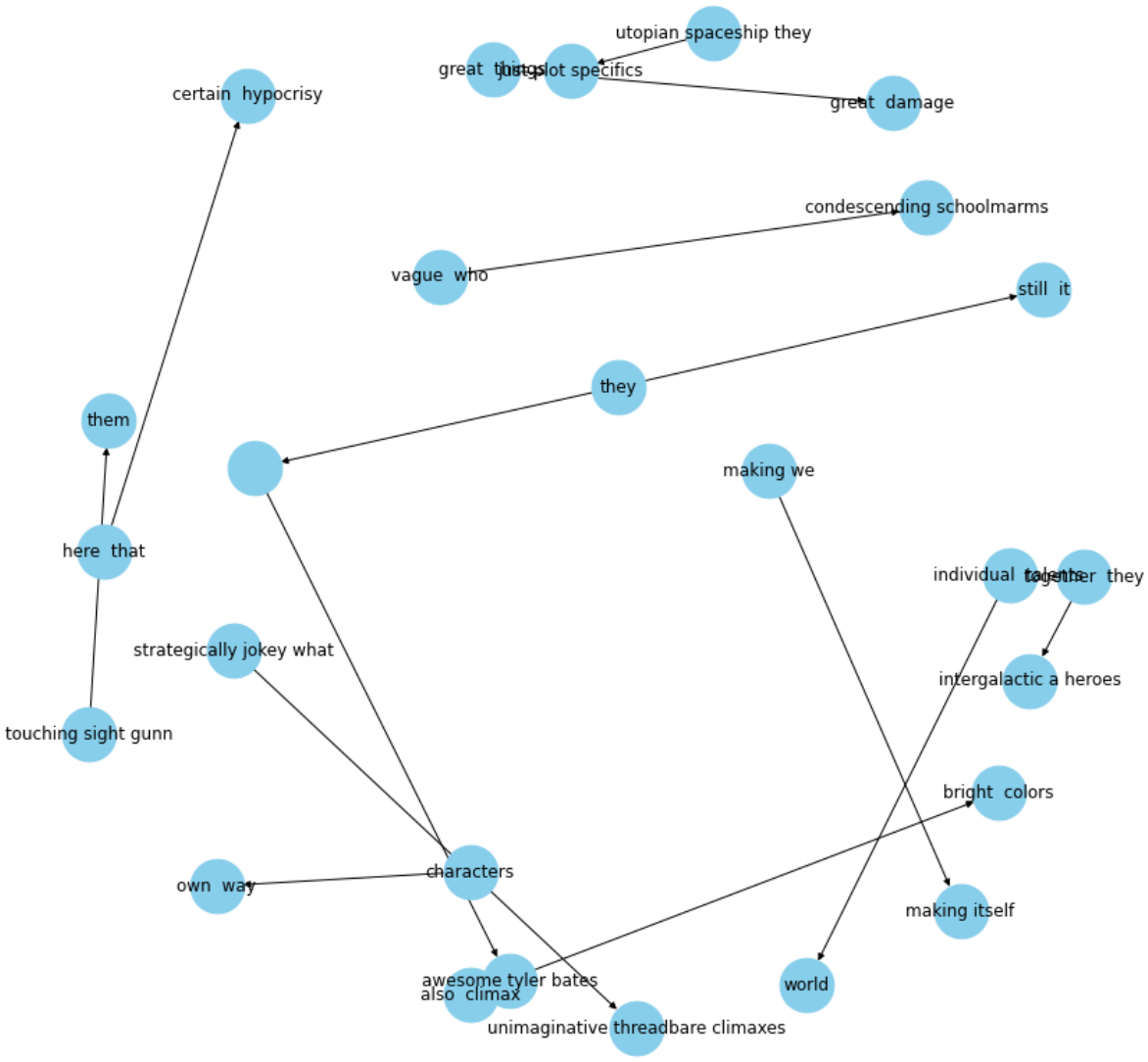


Figure 19. Performance metrics of the LSTM models.

Experiment	Training	Validation	Test	Processing Time
1	0.52	0.47	0.37	5.92
2	0.64	0.40	0.42	5.90
3	0.51	0.42	0.63	5.80
4	0.52	0.51	0.45	6.84
5	0.54	0.47	0.58	13.04
6	0.51	0.58	0.39	11.25

Figure 20. Experiment 1 performance metric plots.

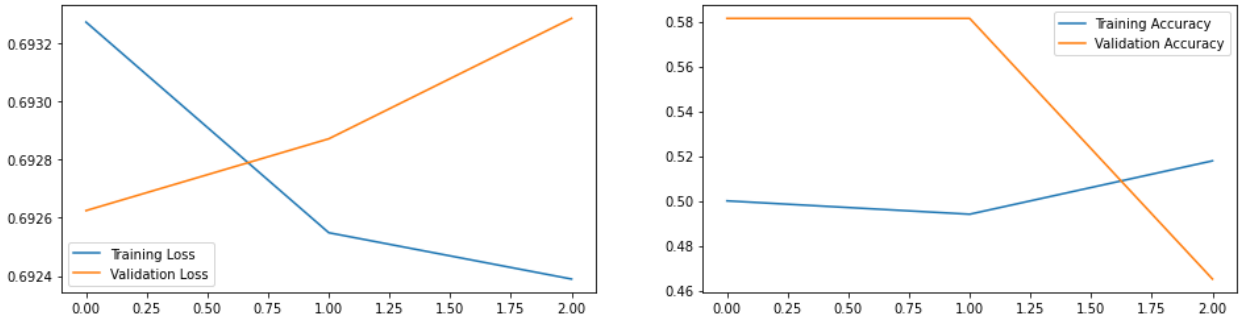


Figure 21. Experiment 2 performance metric plots.

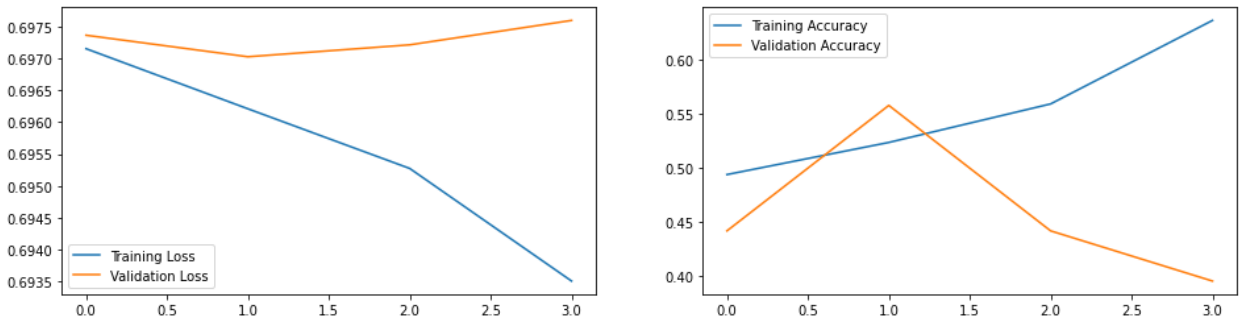


Figure 22. Experiment 3 performance metric plots.

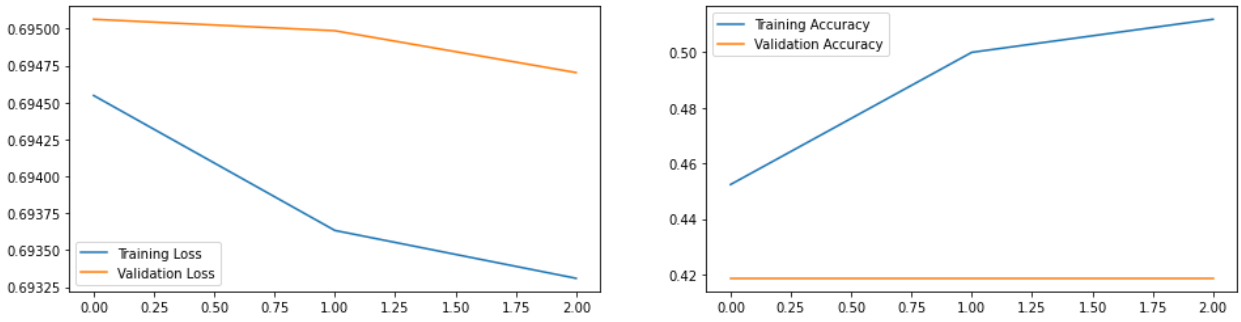


Figure 23. Experiment 4 performance metric plots.

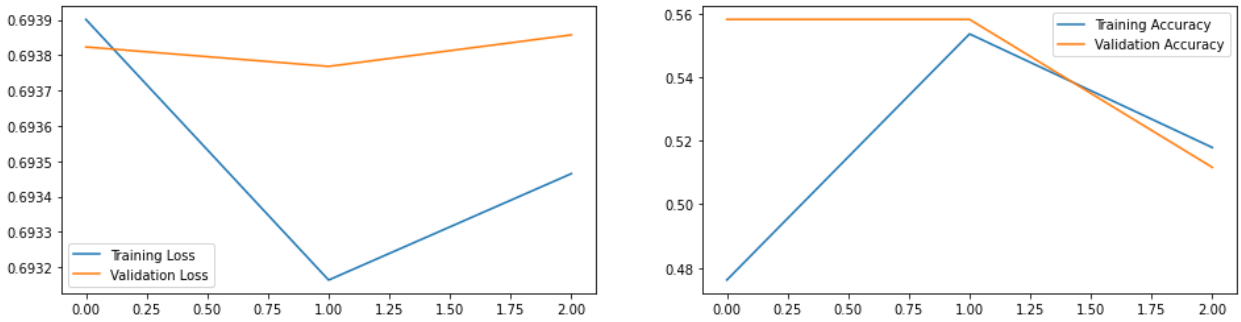


Figure 24. Experiment 5 performance metric plots.

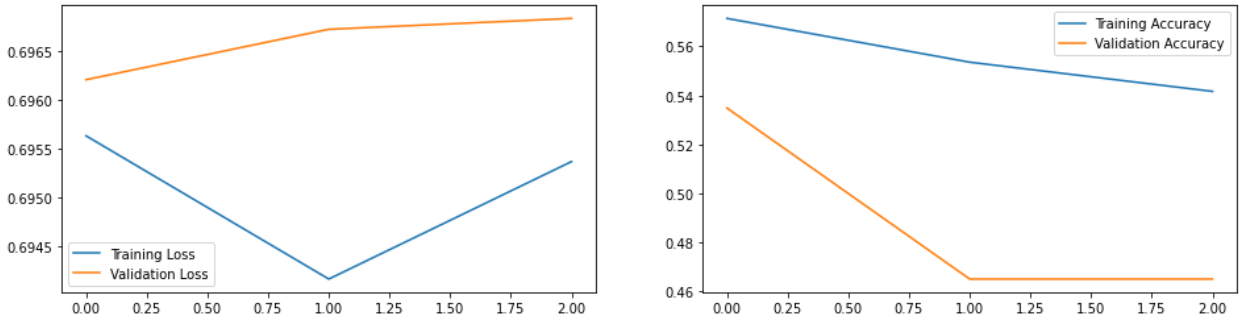


Figure 25. Experiment 6 performance metric plots.