Sentiment Analysis of Harry Potter Movie Series

Cheesoo Kim, Suebin Kim, Kimyung Kim, Donghyeok Choi

1. Topic Selection

1.1 Interest in text analysis

At the first meeting, we found out we had similar interests; movies, science fiction, novels, texts, and graphics. While talking about the Lord of the Rings, Harry Potter, Star Wars, Samgukji(the three kingdoms), we decided to draw a network graph between opposing groups of power. As a way of drawing a network graph, we digitized positiveness and negativeness between each power, or character by AI. Based upon that information, we drew a graph, and analyzed if that information fits what we naturally know.

1.2 Why movie scripts

We thought movie scripts give us the best information about sentiments between each character, since each line has clear speakers and listeners. If we can digitize each line into scores, we can gather the data of scores between each character. Based upon this information, we could draw a graph, whose nodes are characters and edges are scores between each character.

1.3 From Star Wars to Harry Potter

At the topic proposal, we proposed sentiment analysis and network graphing the Star Wars Series. Our original goal was to draw a power map in the Star Wars, but found out that we know little about each series. It was one of the most popular fantasy movies, but we were younger than we thought, and found out we were more friendly with Harry Potter. The pros of changing from the Star Wars to the Harry Potter series is that we are more familiar and it's easier to find out the correctness of our analysis. Also, we were able to process raw data more effectively. Since our memory of the movies were more clear, and the data of the Harry Potter series were divided by scenes, we were able to handle the data as we wished.

2. Project Objectives

2.1 Making a Network between characters

We labeled each node with a specific character's full name(Ex. Harry Potter). For each node, calculations were made from analyzing the text that was shared between two characters. Values of each edge range from -1.0(most negative) to +1.0(most positive). Continuous mapping was used to color the edges - red for positiveness, blue for negativeness - to provide better visual understanding of the graph.

2.2 Verifying sentiment analysis with what we already know

Sentiment analysis is done by AI, so by comparing the result of our graph to what we know, we can analyze if the AI is giving us the right information. It is our big interest to see if the results fit with what we know.

The Harry Potter series, a world-renowned children's novel, has a clear storyline with little ambiguity. The whole angle of Right versus Wrong remains quite straightforward throughout the 7 books(8 movies). Most characters' personality, their traits and loyalty remain unchanged until the end. This simplicity makes it easier for us to analyze any kind of result derived from text mining. We can tell which results are accurate and which are not.

3.Steps of Analysis

3.1 making raw data(listener)

We got movie scripts of each series. What we had was the scene, the speaker, and the context. What we didn't have was the listener. Based upon the scene description, we added the listener section, where we listed who this line is directed to. We tried to narrow down the speaker as specific as possible, for we wanted to give as much data as possible. When it was unclear, we referred to the movies.

There are 8 Harry Potter movies. Sorcerer's Stone and Room of Secrets were done by Chee Soo. The Prisoner of Azkaban and the Goblet of Fire were done by Kimyung. The Order of the phoenix and the Half Blood Prince were done by Donghyeok. 2 last movies, the Deathly Hallows were done by Suebin. We had a discussion about whether we should handle the data by ourselves or we should look for existing data. We were afraid our intervention might pollute the data, but we agreed that every research needs raw data, and sometimes we need to make raw data ourselves. Since the process was done by pure human labor, we checked each other's data for possible typos and errors.

3.2 NLP, change lines into scores

Harry Potter Series Text Sentiment Analysis Using Text Classifier

Use Flair TextClassifier

In [186…
```
from flair.models import TextClassifier
from flair.data import Sentence
from segtok.segmenter import split_single
import pandas as pd
import numpy as np
```

Load classifier

In [187…
```
classifier = TextClassifier.load('en-sentiment')
```

```
2022-06-03 16:47:30,672 loading file /Users/noah/.flair/models/sentiment-en-mi
x-distillbert_4.pt
```

Load text data

In [188…
```
pd = pd.read_csv("/users/noah/desktop/hp1.csv")
```

Make score predict fuction

In [189…
```
def predict(sentence):
    text = Sentence(sentence)
    classifier.predict(text)
    value = text.labels[0].to_dict()['value']
    if value == 'POSITIVE':
        result = text.labels[0].to_dict()['confidence']
    else:
        result = -(text.labels[0].to_dict()['confidence'])
    return round(result, 3)
```

Apply the function

In [190…
```
pd['score'] = pd['dialog'].apply(predict)
```

Save it to CSV file

In [191…
```
pd.to_csv('/users/noah/desktop/hp1_score.csv', index = False)
```

Repeat from 1 to 8

3.3 making node files, suitable for Cytoscape

Make the nodes from the script of Harry Potter

Load script file

[19]
```
import csv
import pandas as pd
import io
from google.colab import files
```

[20]
```
uploaded=files.upload()
```

파일 선택 파일 8개

- **hp1_score.csv**(text/csv) - 135798 bytes, last modified: 2022. 6. 3. - 100% done
- **hp2_score.csv**(text/csv) - 147778 bytes, last modified: 2022. 6. 3. - 100% done
- **hp3_score.csv**(text/csv) - 137812 bytes, last modified: 2022. 6. 3. - 100% done
- **hp4_score.csv**(text/csv) - 110712 bytes, last modified: 2022. 6. 7. - 100% done

```
Saving hp4_score.csv to hp4_score (2).csv
Saving hp5_score.csv to hp5_score (2).csv
Saving hp6_score.csv to hp6_score (2).csv
Saving hp7_score.csv to hp7_score (2).csv
Saving hp8_score.csv to hp8_score (2).csv
```

Concatenate the loaded data to make a node file of whole script

```
[21] df1 = pd.read_csv(io.BytesIO(uploaded['hp1_score.csv']))
     df2 = pd.read_csv(io.BytesIO(uploaded['hp2_score.csv']))
     df3 = pd.read_csv(io.BytesIO(uploaded['hp3_score.csv']))
     df4 = pd.read_csv(io.BytesIO(uploaded['hp4_score.csv']))
     df5 = pd.read_csv(io.BytesIO(uploaded['hp5_score.csv']))
     df6 = pd.read_csv(io.BytesIO(uploaded['hp6_score.csv']))
     df7 = pd.read_csv(io.BytesIO(uploaded['hp7_score.csv']))
     df8 = pd.read_csv(io.BytesIO(uploaded['hp8_score.csv']))
     totaldf=pd.concat((df1,df2,df3,df4,df5,df6,df7,df8), sort=False)
     df=[totaldf, df1, df2, df3, df4, df5, df6, df7, df8]
```

Count the number of times each character plays a line

```
[22] characters=[]
     for i in range(9):
       characters.append(df[i]['character'].value_counts())
```

Build a dataframe of the nodes and the number of lines per each character

```
[23] fulldata=[]
     counts=[0]*9
     for num in range(9):
       count=0
       data=[]
       used=[]

       ## character who had speaked
       for i in characters[num].index:
         data.append([count, i, characters[num][i]])
         count+=1

       ## characters who had only listened
       for i in df[num]['listener']:
         if(type(i)==str):
           for k in i.split(", "):
             if((k in characters[num].index)==False):
               if((k in used)==False):
                 used.append(k)
                 data.append([count ,k, 0])
                 count+=1
       fulldata.append(data)
```

```
[24] nodes=[]
     for i in range(9):
       node=pd.DataFrame(fulldata[i], columns=['Id', 'Label', 'Count'])
       nodes.append(node)
```

Save the node data into csv files.

```
[25] from google.colab import drive
     filenames=["totalnode", "node1", "node2", "node3", "node4", "node5", "node6", "node7", "node8"]
     drive.mount('/content/drive', force_remount=True)
     path=[]
     for i in range(9):
       path.append('/content/drive/My Drive/'+filenames[i]+'.csv')

     for i in range(9):
       with open(path[i], 'w', encoding = 'utf-8-sig') as f:
         nodes[i].to_csv(f, index=False)
```

```
Mounted at /content/drive
```

3.4 making edge files, suitable for Cytoscape

Make the edge between nodes

In [13]:
```
import csv
import pandas as pd
```

Load node file

In [14]:
```
node = pd.read_csv('/users/noah/desktop/nodes/node1.csv')
#node = pd.read_csv('/users/noah/desktop/nodes/totalnode.csv')
```

Load text and score to dataframe

In [15]:
```
df1 = pd.read_csv('/users/noah/desktop/hp1_score.csv')
df2 = pd.read_csv('/users/noah/desktop/hp2_score.csv')
df3 = pd.read_csv('/users/noah/desktop/hp3_score.csv')
df4 = pd.read_csv('/users/noah/desktop/hp4_score.csv')
df5 = pd.read_csv('/users/noah/desktop/hp5_score.csv')
df6 = pd.read_csv('/users/noah/desktop/hp6_score.csv')
df7 = pd.read_csv('/users/noah/desktop/hp7_score.csv')
df8 = pd.read_csv('/users/noah/desktop/hp8_score.csv')
```

Concat the dataframes and make it to one dataframe

In [18]:
```
scores = df1
#scores = pd.concat((df1,df2,df3,df4,df5,df6,df7,df8), sort=False, ignore_index = True)
```

Replace the NULL or NA

In [19]:
```
scores['listener'].fillna('', inplace = True)
```

In [20]:
```
from itertools import permutations
import numpy as np
import time
```

Make source and target by permutation function. Make weight, count and mean variable to use later

In [21]:
```
edge_base = pd.DataFrame(list(permutations(node['Label'], 2)))
edge_base.rename(columns = {0:'source', 1:'target'}, inplace = True)
weight = np.zeros(len(edge_base))
count = np.zeros(len(edge_base))
mean = np.zeros(len(edge_base))
print(edge_base)
```

```
                source              target
0         Harry Potter          Ron Weasley
1         Harry Potter     Hermione Granger
2         Harry Potter     Albus Dumbledore
3         Harry Potter        Rubeus Hagrid
4         Harry Potter        Severus Snape
...                ...                 ...
41407  Minerva Mcgonagall  Gellert Grindelward
41408  Minerva Mcgonagall          Ron Weasley
41409  Minerva Mcgonagall              Bogrod
41410  Minerva Mcgonagall    Ariana Dumbledore
41411  Minerva Mcgonagall  Neville Longbottom

[41412 rows x 2 columns]
```

Make copy of edge_base and turn into the list

In [22]:
```
edge_test = edge_base.copy()
edge_list = edge_test.values.tolist()
```

Find the weight and count. Then calculate the mean sentimental value between source and target

In [23]:
```
start_time = time.time()
```

```
for i in range(len(edge_list)):

    print(f"{i+1}/{len(edge_list)}", end='\r')

    for j in range(len(scores)):

        if ((edge_list[i][1] in scores['listener'][j]) & (edge_list[i][0] == scores['character'][j])):

            count[i] +=1
            weight[i] += scores['score'][j]

        else:
            weight[i] += 0

        if count[i] != 0:
            mean[i] = weight[i] / count[i]

end_time = time.time()

print(f"Time is {end_time - start_time} sec")
```

```
Time is 13753.822365045547 sec
```

Make them to the dataframe

In [24]:
```
edge_base['mean'] = mean
edge_base['weight'] = weight
edge_base['count'] = count
print(edge_base)
```

```
                source              target      mean  weight  count
0          Harry Potter          Ron Weasley -0.011111  -4.589  413.0
1          Harry Potter     Hermione Granger -0.022344  -9.429  422.0
2          Harry Potter     Albus Dumbledore -0.120302 -17.925  149.0
3          Harry Potter        Rubeus Hagrid -0.161384 -13.879   86.0
4          Harry Potter        Severus Snape -0.083850  -3.354   40.0
...                 ...                 ...       ...     ...    ...
41407  Minerva Mcgonagall  Gellert Grindelward  0.000000   0.000    0.0
41408  Minerva Mcgonagall          Ron Weasley  0.000000   0.000    0.0
41409  Minerva Mcgonagall              Bogrod   0.000000   0.000    0.0
41410  Minerva Mcgonagall    Ariana Dumbledore  0.000000   0.000    0.0
41411  Minerva Mcgonagall  Neville Longbottom   0.000000   0.000    0.0

[41412 rows x 5 columns]
```

Use some techniques to what we want to make finanlly

In [25]:
```
edge_id = pd.DataFrame(list(permutations(node['Id'], 2)))
edge_id.rename(columns = {0:'source', 1:'target'}, inplace = True)
edge_id['mean'] = mean
print(edge_id)
```

```
       source  target      mean
0           0       1 -0.011111
1           0       2 -0.022344
2           0       3 -0.120302
3           0       4 -0.161384
4           0       5 -0.083850
...       ...     ...       ...
41407     203     198  0.000000
41408     203     199  0.000000
41409     203     200  0.000000
41410     203     201  0.000000
41411     203     202  0.000000

[41412 rows x 3 columns]
```

Save it to edge csv file

In [26]:
```
edge_id.to_csv('/users/noah/desktop/edge/totaledge.csv', index = False)
```

Repeat 1 to 8 and total to make all edge files

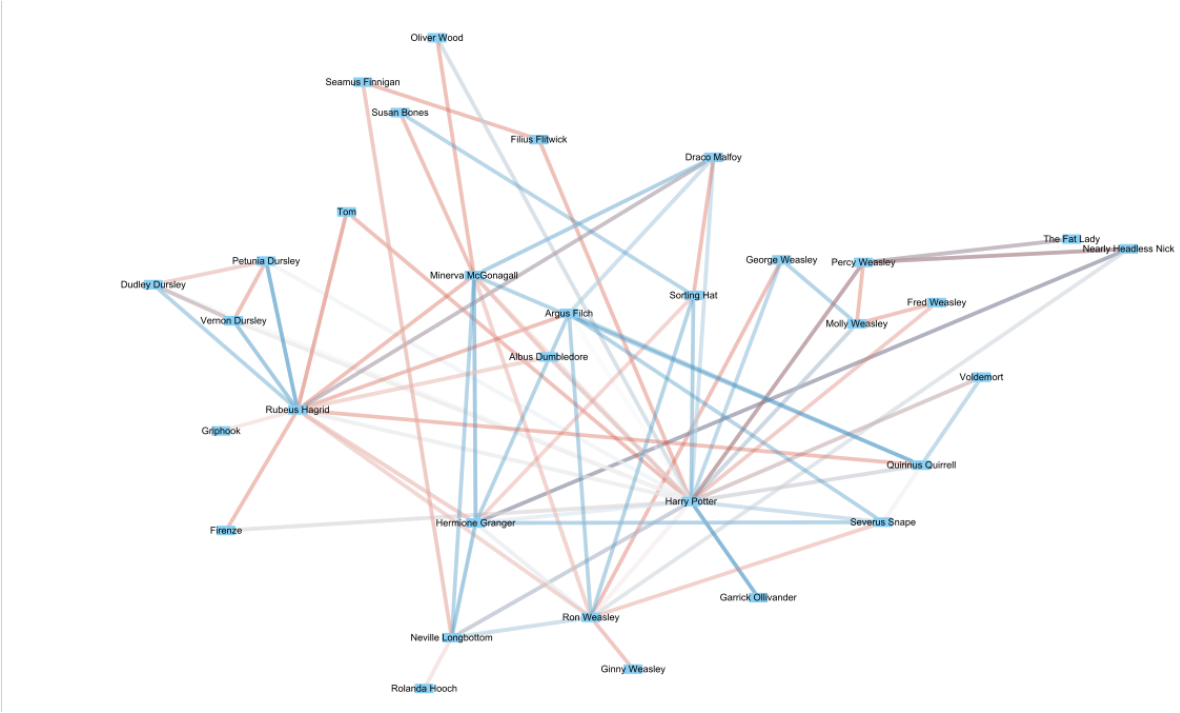
3.5 how to handle each files in Cytoscape

We have edge tables and node tables. Edge tables contain source and destination information, written in number, and attribute value of mean score between each number. In node tables, we have numbers and their corresponding Label, or names. We also have count information, which shows how many times each person has spoken. From these two datas, we drew a graph of each Harry Potter movie. At first, we made a graph from edge files, and to enhance our visibility, we hid all 0 edges. These are the edges that show there was no relationship between two characters. Then, we deleted nodes that are ambiguous, such as All, Students, Class, or Boys, Girls, Witch etc. By these two steps we were able to draw a more clear graph.

4. Results

Below are 8 network graphs and our analysis.

4.1 Harry Potter and the Sorcerer's Stone



Character	Positive	Negative
Harry Potter	George, Ollivander, Snape, Hermione, Molly, Sorting Hat, Hagrid	Quirrell, Tom, Voldemort, Percy, Fred, Ron
Ron Weasley	Sorting Hat, Filch, Neville	Harry, Hermione, Snape
Hermione Granger	Snape, Neville, Harry, McGonagall, Filch	Hagrid, Sorting Hat

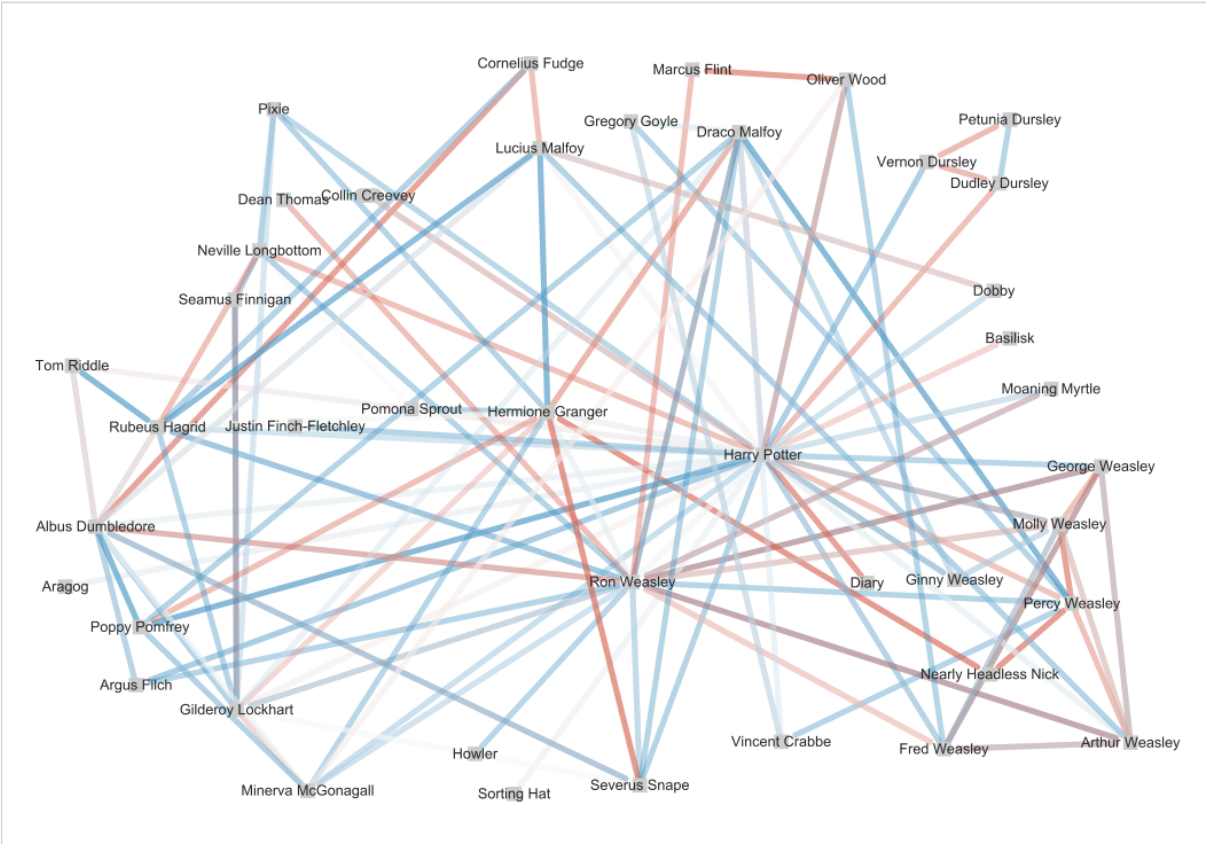
Interesting Relationships

1. Hagrid with the Dusley Family. Even though Hagrid turned their son into a pig, and they had a heavy fight over Harry entering Hogwart, their relationship appeared positive.

2. Conversation between the Dusley Family appeared Negative. We made sure each line falls beyond a specific listener, so the negative relationship doesn't appear to be related to Harry. It is their tone and nuisance of their speech that shows a negative relationship. Maybe their way of speaking is damaging their own relationship.

3. Although their faces show their hatred toward each other, Malfoy and Harry's relationship has given us a positive score.

4-2 Harry Potter and the Room of Secrets



Character	Positive	Negative
Harry Potter	Vernon, Dobby, George, Myrtle, Ginny, Hagrid, Fred, Snape, Filch, Pomfly, McGonagall	Neville, Diary, Oliver Wood, Doodley
Ron Weasley	Filch, Lockhart, Howler, Snape, Percy, Neville, Hagrid	Marcus Flint, Dumbledore, Fred, Molly, Geroge, Myrtle, Author

Hermione Granger	Sprout, Lucius Malfoy, McGonagall, Pixie	Nearly Headless Nick, Snape, Pomfly, Malfoy

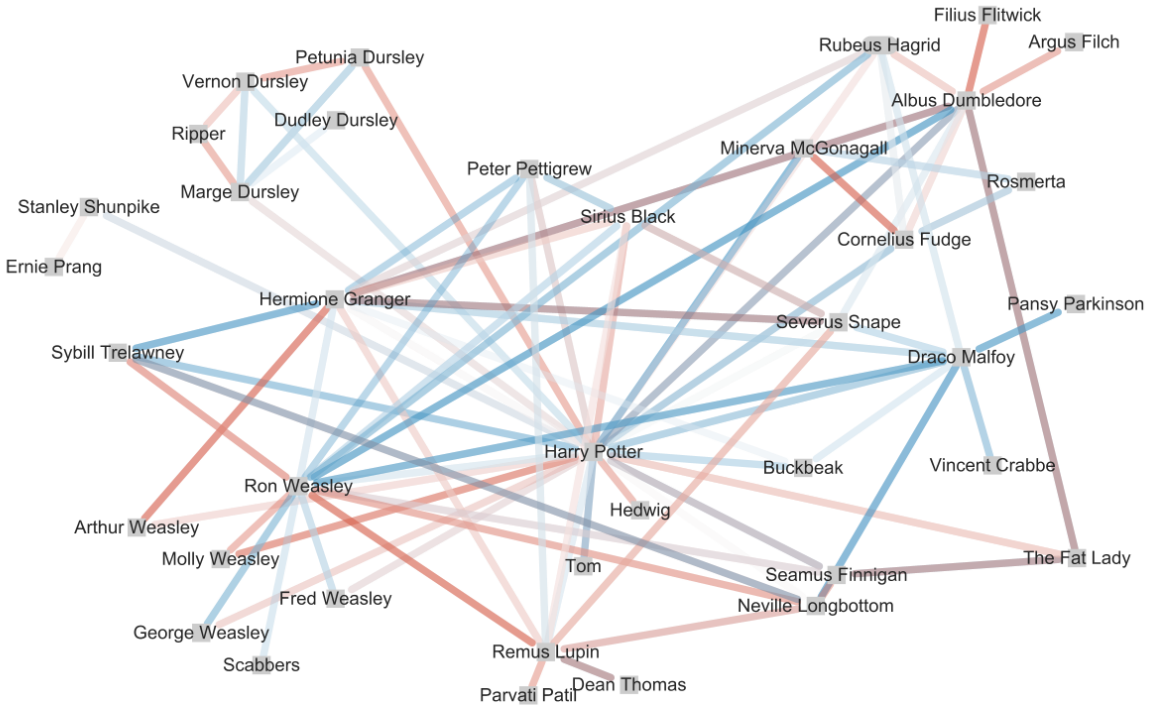
Interesting Relationships

1. Dobby's negative score toward Lucius Malfoy and positive score toward Harry appears really well, which is true to what we already know.

2. Pixies are small devils that caused havoc in class, but Harry, Hermione, Lockhard are all positive.

3. The Weasleys show negative relationships toward each other, but only to a small degree.

4-3 Harry Potter and the Prisoner of Azkaban



Character	Positive	Negative
Harry Potter	McGonagall, Trelawney, Dumbledore, Malfoy, Bukbeak, Tom, Ron, Vernon	Molly, The Fat Lady, Hedwig, Sirius, Petunia, Pettigrew, Geroge, Fred
Ron Weasley	George, Hermione, Malfoy, Pettigrew, Sirius, Dumbledore, Hagrid	Trelawney, Lupin, Molly, Neville
Hermione Granger	Trelawney, Pettigrew,	Author, Dumbledore, Snape

	Malfoy, Ron	

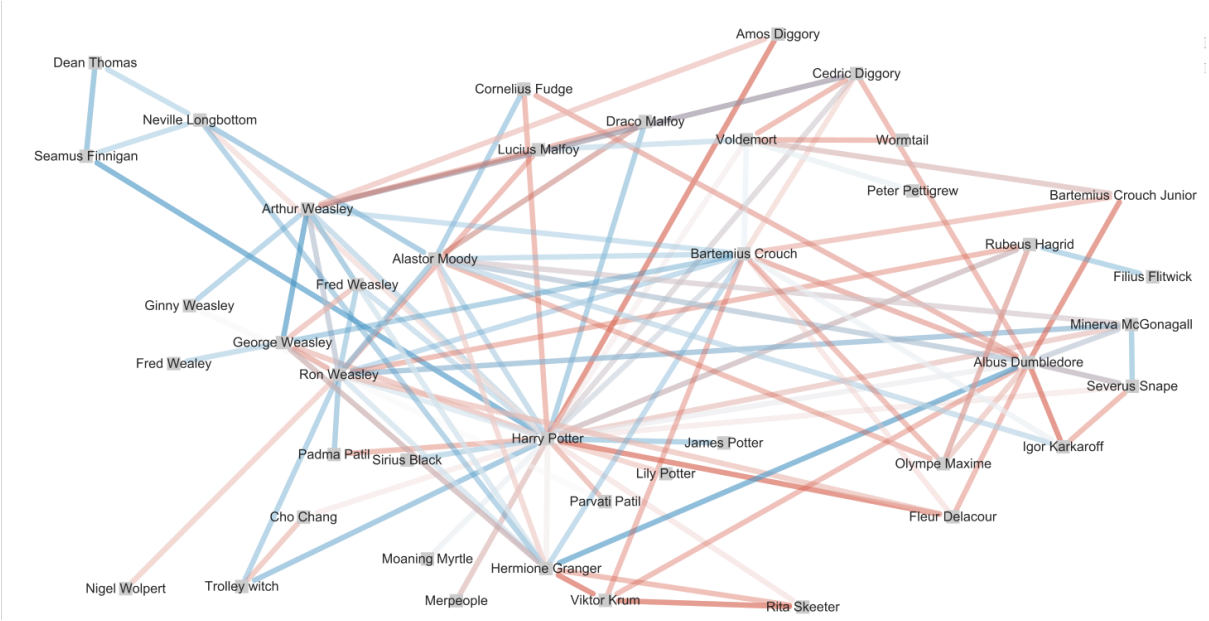
Interesting Relationships

1. Remus Lupin got negative scores from everyone, although he appeared as a very thoughtful teacher.

2. Malfoy is the man of the year, he got positive scores from everyone.

3. On the other hand, Harry showed negative scores on the Weasleys, Hedwig and his friends.

4.4 Harry Potter and the Goblet of Fire



Character	Positive	Negative
Harry Potter	Trolly Witch, Malfoy, Seamus, Moody, James	Fudge, Amos Diggory, Padma, Merpeople, Parvati, George, Delacour, Snape
Ron Weasley	Hermione, Padma, Trolly Witch, Fred, Crouch	George, Delacour, Hagrid
Hermione Granger	Dumbledore, Author, George, Neville	Krum, Rita Skeeter, George, Moody

Interesting Relationships

1. Positive and Negative relationships become more clearer. Especially the conflict between the Weasley's and Malfoy's.

2. Harry's love toward Cho Chang looks very pathetic, seeing the negative score between them.

3. The Bad Guys get more negative scores, and the Good Guys get more positive scores. Maybe it's because Harry Potter is a novel, and the writer has a certain taste.

4-5. Harry Potter and the Order of the Phoenix

Clear Relations that fit the movie's description

1-1) Cornelius Fudge-Lucius Malfoy : Positive Relation

1-2) Harry Potter, Ron Weasley, Hermione Granger, Neville Longbottom, Luna Lovegood : Positive Relation. 5 students are positively intertwined with each other in the graph.

1-3) Bellatrix Lestrange – Harry Potter, Neville Longbottom : Negative Relation. Negative feelings between two opposing characters are represented accurately.

1-4) Dolores Umbridge – Minerva McGonagall – Albus Dumbledore : Negative/Positive Relation.

Relations that do not fit the movie's description

2-1) Arthur Weasley – Fred, George Weasley : Negative Relation. In the movie, they are close family members.

2-2) Sirius Black – James Potter : Both positive and negative relations are showed. In the movie, two are close friends with no negative feeling.

4-6. Harry Potter and the Half-blood Prince

Clear Relations that fit the movie's description :

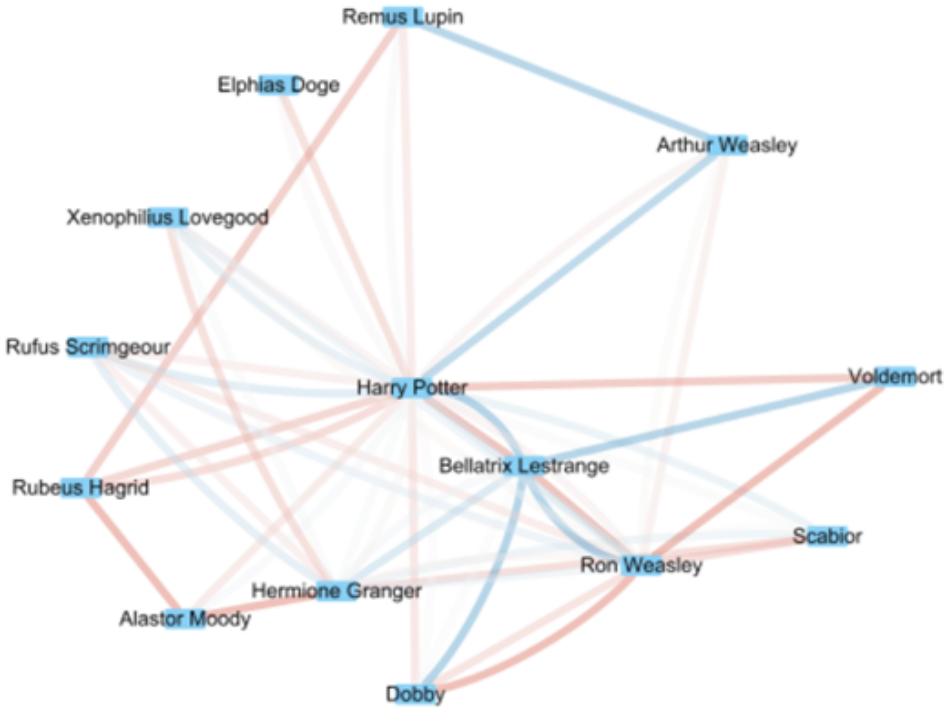
1-1) Albus Dumbledore's Positive Relations with : Horace Slughorn, Tom Riddle, Severus Snape

Relations that do not fit the movie's description :

2-1) Overall, Harry Potter and his close friends – Ron Weasley, Hermione Granger, Luna Lovegood, Fred Weasley, Ginny Weasley share a negative dynamic.

2-2) Albus Dumbledore's Positive Relation with Draco Malfoy – Albus Dumbledore's character is kind and understanding; he applies positive, courteous manner of speech even to his enemies. In the movie, Draco Malfoy tries to kill Albus Dumbledore.

4-7. Harry Potter and the Deathly Hallows – Part 1



Clear Relations that fit the movie's description :

1-1) Harry Potter – Ron Weasley, Dobby : Positive Relation

1-2) Bellatrix Lestrange – Harry Potter, Dobby, Ron Weasley : Negative Relation

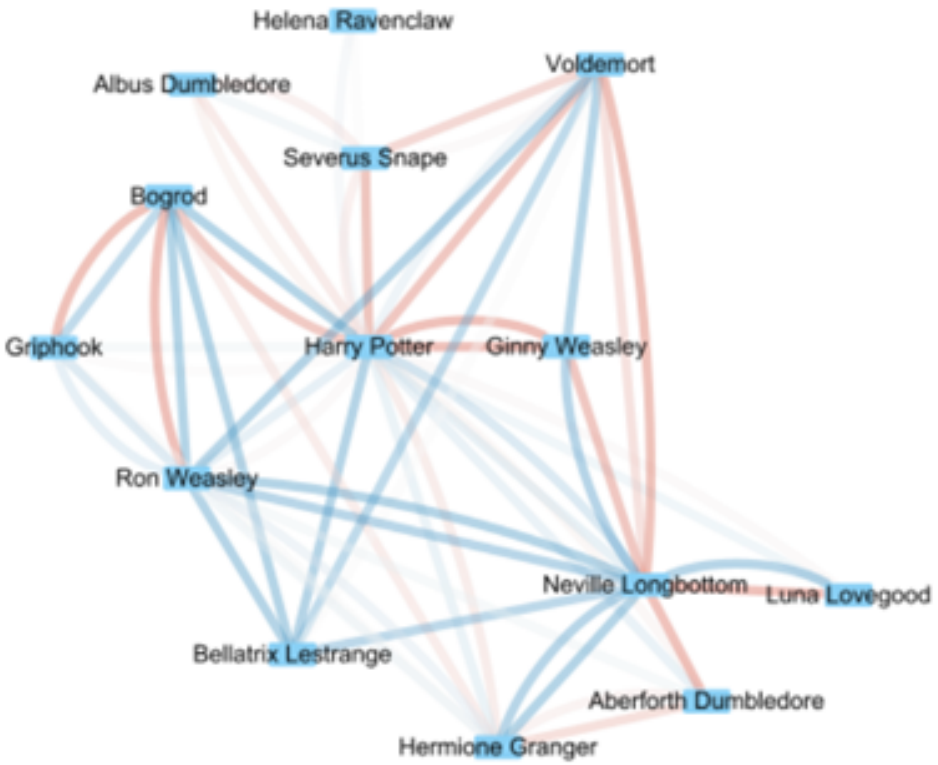
1-3) Overall Positive Relation between Harry Potter's side of characters (Left side of the Graph)

Relations that do not fit the movie's description :

2-1) Voldemort – Harry Potter, Ron Weasley : Positive Relation (Harry and Voldemort are best friends?) – Not many lines were shared between two characters. Few lines include phrases like 'bring him to me', 'I have seen your heart'. These were interpreted as positive sentiments

2-2) Negative Relation between Arthur Weasley and other characters

4-8. Harry Potter and the Deathly Hallows – Part 2



Clear Relations that fit the movie's description :

1-1) Harry Potter – Ginny Weasley : Positive Relation (Two are lovers)

1-2) Bellatrix Lestrange – Harry Potter, Ron Weasley, Neville Longbottom : Negative Relation(Enemies)

Relations that do not fit the movie's description :

2-1) Voldemort – Harry Potter, Neville Longbottom : Positive Relation. Clearly this form of text analysis is inaccurate for characters that use a lot of sarcasm/irony/mockery in their speech.

5. Conclusion

5.1 Does our result fit what we know?

There were cases where the graph showed us what we expected. Small negative relationships between the Weasley's are understandable, for they are a real family, and they sometimes fight and sometimes get scolded. From time to time there are positive relationships between certain members and it not only fits with what we think about the

Weasleys, but it also fits with what we think about real families. And the graph gives us a rough idea of the power map; how characters are divided into groups, and what seems to be the main conflict in the movies.

Some of the edges showed opposite results in respect to the original Harry Potter series' character dynamic. Characters fighting on opposing parties often shared a positive relation. Namely, we found positive relations between Harry Potter and Voldemort in movies 7-8 interesting. Since the fight between the two takes up most of the movies' plot, this result seemed highly flawed.

We came up with several possibilities why the graph shows these results.

There are simply limited lines in the movies. As the number of lines increases, the scores are familiar with what we think. If there were more than 50 lines, the results were almost the same as what we already knew. The problem was that movies have certain limits, mostly time. Not all relationships had enough data to show their true relationships. Only main characters have enough time and focus. We, as humans, score relationships from many aspects, from the tone of the voice, their facial expressions, how they react using body languages, and the looks in their eyes. Among so much valuable datas, we only used texts, the verbal interactions between characters only. We can say there was not enough data to analyze the relationships between characters correctly.

Both in real life and literature, a character's way of speaking doesn't always reflect what he/she is thinking inside. One may make a positive remark while feeling the exact opposite. When characters employ irony, sarcasm or mockery in their speech, text analysis becomes inaccurate. From the results above we were able to specify certain characters in the Harry Potter series whose manner of speech caused complications. First, there is Voldemort. His frequent use of sarcasm and mockery rendered many of his words - for example, 'bring him to me(to kill)', 'you are brave'(mockery) - as 'positive comments' while in fact he is a violent tyrant. Albus Dumbledore also proved to be a tricky character. Quite opposite to Voldemort, Dumbledore always keeps a calm, courteous vibe in his speech; even to his enemies. This led to some confusing relations between Dumbledore and his foes.

5.2 What we can improve

From a technical point of view, the NLP model we used in this project classifies and scores the sentiment of a sentence by splitting it into the words. Sentence has its own flow and should not be seen as a combination of words. In the movie, the writer tries to build a characteristic of each character, and it leads to the difference in the way of speaking. This might decrease the accuracy of sentiment prediction. To avoid the problem, we should train the NLP model with the lines of each character. Since the lines are limited, so applying the way each character speaks, we should make the lines of the characters. This allows the model to be trained enough by enlarging the train dataset.

As for the amount of data, if we used books instead of movies, we could have drawn a more correct graph. However, we are a little bit skeptical about this, for the book contains more data, but most of them are focused on the main characters, Harry, Ron, and Hermione. It would be interesting to see how graphs from movies and graphs from books differ.

5.3 Contribution of each member

Making raw data was distributed equally between each member. Movies 1,2 were done by Cheesoo, 3,4 were done by Kimyung, 5,6 were done by Donghyeok, and 7-1,7-2 were done by Suebin. Donghyeok worked on sentiment analysis, and changed raw data into scores. Kimyung made each character into a node and checked the frequency. Cheesoo worked on analyzing movies 1-4 using cytoscape. Suebin worked on drawing the node/edge graph of movies 5-8 and each graph's analysis.