Character Arcs in Harry Potter

February 3, 2023

1 Character Arcs in Harry Potter

1.1 Introduction

A **Character Arc** refers to how a character changes after interacting with events, which has a significant impact on drawing audiences. The successful characterization in **Harry Potter** is worth analysing as a world-famous series. This aids scriptwriters decision on creating popular characters with specific character arcs.

This report begins by checking and cleaning data in the **Data Exploration and Cleaning** Section, followed by the creation of a dataframe to organise further analysis. A network is generated to show the relation among 30 important characters in the **Visualising Network of Main Characters** Section. In the final section, **Visualizing Character Arcs**, character arcs across movies for 10 of most popular characters in Harry Potter are displayed based on their emotions revealed in dialogues.

1.1.1 Data Exploration and Cleaning

[1]:

```
import pandas as pd
import plotly.express as px
```

[2]:

```
import warnings
warnings.filterwarnings("ignore")
```

[3]:

```
pd.set_option('display.max_rows', None)
```

[4]:

```
#Data Source - https://www.mavenanalytics.io/blog/maven-magic-challenge?
 ↪utm_source=linkedin&utm_campaign=mavenmagicchallenge_jp20211123
#Importing Data
movies = pd.read_csv('./Harry_Potter_Dataset/Movies.txt',␣
 ↪header=0,sep='\t',encoding = "ISO-8859-1")
characters = pd.read_csv('./Harry_Potter_Dataset/Characters.txt',␣
 ↪header=0,sep='\t',encoding = "ISO-8859-1")
dialogues = pd.read_csv('./Harry_Potter_Dataset/Dialogue.txt',␣
 ↪header=0,sep='\t',encoding = "ISO-8859-1")
chapters = pd.read_csv('./Harry_Potter_Dataset/Chapters.txt',␣
 ↪header=0,sep='\t',encoding = "ISO-8859-1")
print('4 datasets imported')
```

```
4 datasets imported
```

[5]:

```
#Checking the Characteristics of the Dataset
print("The data has "+str(dialogues.shape[0])+" dialogues spoken by␣
 ↪"+str(characters.shape[0])+
      " characters across "+str(movies.shape[0])+" movies")
```

```
The data has 7444 dialogues spoken by 166 characters across 8 movies
```

[6]:

```
#Checking for Duplicates
if max(dialogues['Dialogue ID'])==dialogues.shape[0]:
    print('There are no duplicates in the data')

else:
    print('There are duplicates in the data')
```

```
There are no duplicates in the data
```

[7]:

```
#Merging the Data Sets
hp_df = movies.merge(chapters.merge(characters.merge(dialogues, how='inner',␣
 ↪on='Character ID'),
                                    how='inner', on='Chapter ID'), how='inner',␣
 ↪on='Movie ID')
print('All 4 Datasets merged')
```

```
All 4 Datasets merged
```

[8]:

```
#Picking Relevant Columns
hp_df_final = hp_df[['Dialogue ID', 'Movie Title', 'Character Name',␣
 ↪'Dialogue']].\
                sort_values('Dialogue ID').reset_index().drop(columns='index')
print('Identified relevant columns and stored in a dataframe')
```

```
Identified relevant columns and stored in a dataframe
```

1.1.2 Visualising Network of Important Characters

[9]:

```
#Selecting 30 important characters who have more dialogues in Harry Potter
main_characters=hp_df_final['Character Name'].value_counts()[:30]
main_characters=main_characters.index
print('30 Important Characters in Harry Potter are:',", ".join(str(x) for x in␣
 ↪main_characters ))
```

```
30 Important Characters in Harry Potter are: Harry Potter, Ron Weasley, Hermione
Granger, Albus Dumbledore, Rubeus Hagrid, Voldemort, Severus Snape, Minerva
McGonagall, Horace Slughorn, Neville Longbottom, Remus Lupin, Draco Malfoy,
Alastor Moody, Fred Weasley, Dolores Umbridge, Arthur Weasley, Cornelius Fudge,
Sirius Black, George Weasley, Ginny Weasley, Vernon Dursley, Bellatrix
Lestrange, Lucius Malfoy, Luna Lovegood, Molly Weasley, Dobby, Gilderoy
Lockhart, Seamus Finnigan, Griphook, Dudley Dursley
```

[10]:

```
#Might take about a minute to run
!pip install -U spacy
!python3 -m spacy download en_core_web_sm
!pip install nltk
import spacy
nlp=spacy.load('en_core_web_sm')
```

```
Collecting spacy
  Downloading
spacy-3.5.0-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (6.6 MB)
                           6.6/6.6 MB
55.6 MB/s eta 0:00:0000:0100:01
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in
/opt/conda/lib/python3.9/site-packages (from spacy) (4.64.0)
Requirement already satisfied: numpy>=1.15.0 in /opt/conda/lib/python3.9/site-
packages (from spacy) (1.21.6)
Collecting spacy-legacy<3.1.0,>=3.0.11
  Downloading spacy_legacy-3.0.12-py2.py3-none-any.whl (29 kB)
Collecting spacy-loggers<2.0.0,>=1.0.0
  Using cached spacy_loggers-1.0.4-py3-none-any.whl (11 kB)
Collecting srsly<3.0.0,>=2.4.3
  Using cached
srsly-2.4.5-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (490 kB)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in
/opt/conda/lib/python3.9/site-packages (from spacy) (2.27.1)
Requirement already satisfied: packaging>=20.0 in /opt/conda/lib/python3.9/site-
packages (from spacy) (21.3)
Collecting thinc<8.2.0,>=8.1.0
  Downloading
thinc-8.1.7-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (825 kB)
                          825.4/825.4 KB
60.0 MB/s eta 0:00:00
Collecting murmurhash<1.1.0,>=0.28.0
  Using cached murmurhash-1.0.9-cp39-cp39-manylinux_2_5_x86_64.manylinux1_x86_64
.manylinux_2_17_x86_64.manylinux2014_x86_64.whl (21 kB)
Collecting pydantic!=1.8,!=1.8.1,<1.11.0,>=1.7.4
  Downloading
pydantic-1.10.4-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.2
MB)
                           3.2/3.2 MB
80.5 MB/s eta 0:00:00:00:01
Collecting typer<0.8.0,>=0.3.0
  Using cached typer-0.7.0-py3-none-any.whl (38 kB)
Collecting cymem<2.1.0,>=2.0.2
  Using cached
cymem-2.0.7-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (35 kB)
Requirement already satisfied: setuptools in /opt/conda/lib/python3.9/site-
```

```
packages (from spacy) (62.1.0)
Collecting smart-open<7.0.0,>=5.2.1
  Using cached smart_open-6.3.0-py3-none-any.whl (56 kB)
Collecting catalogue<2.1.0,>=2.0.6
  Using cached catalogue-2.0.8-py3-none-any.whl (17 kB)
Collecting pathy>=0.10.0
  Using cached pathy-0.10.1-py3-none-any.whl (48 kB)
Collecting preshed<3.1.0,>=3.0.2
  Using cached preshed-3.0.8-cp39-cp39-manylinux_2_5_x86_64.manylinux1_x86_64.ma
nylinux_2_17_x86_64.manylinux2014_x86_64.whl (128 kB)
Collecting langcodes<4.0.0,>=3.2.0
  Using cached langcodes-3.3.0-py3-none-any.whl (181 kB)
Requirement already satisfied: jinja2 in /opt/conda/lib/python3.9/site-packages
(from spacy) (3.1.1)
Collecting wasabi<1.2.0,>=0.9.1
  Downloading wasabi-1.1.1-py3-none-any.whl (27 kB)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/opt/conda/lib/python3.9/site-packages (from packaging>=20.0->spacy) (3.0.8)
Requirement already satisfied: typing-extensions>=4.2.0 in
/opt/conda/lib/python3.9/site-packages (from
pydantic!=1.8,!=1.8.1,<1.11.0,>=1.7.4->spacy) (4.4.0)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.9/site-
packages (from requests<3.0.0,>=2.13.0->spacy) (3.3)
Requirement already satisfied: charset-normalizer~=2.0.0 in
/opt/conda/lib/python3.9/site-packages (from requests<3.0.0,>=2.13.0->spacy)
(2.0.12)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/conda/lib/python3.9/site-packages (from requests<3.0.0,>=2.13.0->spacy)
(2022.9.24)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/opt/conda/lib/python3.9/site-packages (from requests<3.0.0,>=2.13.0->spacy)
(1.26.9)
Collecting confection<1.0.0,>=0.0.1
  Downloading confection-0.0.4-py3-none-any.whl (32 kB)
Collecting blis<0.8.0,>=0.7.8
  Using cached
blis-0.7.9-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (10.2 MB)
Requirement already satisfied: click<9.0.0,>=7.1.1 in
/opt/conda/lib/python3.9/site-packages (from typer<0.8.0,>=0.3.0->spacy) (8.1.3)
Requirement already satisfied: MarkupSafe>=2.0 in /opt/conda/lib/python3.9/site-
packages (from jinja2->spacy) (2.1.1)
Installing collected packages: cymem, wasabi, typer, spacy-loggers, spacy-
legacy, smart-open, pydantic, murmurhash, langcodes, catalogue, blis, srsly,
preshed, pathy, confection, thinc, spacy
Successfully installed blis-0.7.9 catalogue-2.0.8 confection-0.0.4 cymem-2.0.7
langcodes-3.3.0 murmurhash-1.0.9 pathy-0.10.1 preshed-3.0.8 pydantic-1.10.4
smart-open-6.3.0 spacy-3.5.0 spacy-legacy-3.0.12 spacy-loggers-1.0.4 srsly-2.4.5
thinc-8.1.7 typer-0.7.0 wasabi-1.1.1
```

```
/opt/conda/lib/python3.9/site-packages/requests/__init__.py:102:
RequestsDependencyWarning: urllib3 (1.26.9) or chardet
(5.0.0)/charset_normalizer (2.0.12) doesn't match a supported version!
  warnings.warn("urllib3 ({}) or chardet ({})/charset_normalizer ({}) doesn't
match a supported "
Collecting en-core-web-sm==3.5.0
  Downloading https://github.com/explosion/spacy-
models/releases/download/en_core_web_sm-3.5.0/en_core_web_sm-3.5.0-py3-none-
any.whl (12.8 MB)
                           12.8/12.8 MB
81.9 MB/s eta 0:00:0000:010:01
Requirement already satisfied: spacy<3.6.0,>=3.5.0 in
/opt/conda/lib/python3.9/site-packages (from en-core-web-sm==3.5.0) (3.5.0)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in
/opt/conda/lib/python3.9/site-packages (from spacy<3.6.0,>=3.5.0->en-core-web-
sm==3.5.0) (2.0.8)
Requirement already satisfied: jinja2 in /opt/conda/lib/python3.9/site-packages
(from spacy<3.6.0,>=3.5.0->en-core-web-sm==3.5.0) (3.1.1)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<1.11.0,>=1.7.4 in
/opt/conda/lib/python3.9/site-packages (from spacy<3.6.0,>=3.5.0->en-core-web-
sm==3.5.0) (1.10.4)
Requirement already satisfied: setuptools in /opt/conda/lib/python3.9/site-
packages (from spacy<3.6.0,>=3.5.0->en-core-web-sm==3.5.0) (62.1.0)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in
/opt/conda/lib/python3.9/site-packages (from spacy<3.6.0,>=3.5.0->en-core-web-
sm==3.5.0) (2.27.1)
Requirement already satisfied: thinc<8.2.0,>=8.1.0 in
/opt/conda/lib/python3.9/site-packages (from spacy<3.6.0,>=3.5.0->en-core-web-
sm==3.5.0) (8.1.7)
Requirement already satisfied: smart-open<7.0.0,>=5.2.1 in
/opt/conda/lib/python3.9/site-packages (from spacy<3.6.0,>=3.5.0->en-core-web-
sm==3.5.0) (6.3.0)
Requirement already satisfied: numpy>=1.15.0 in /opt/conda/lib/python3.9/site-
packages (from spacy<3.6.0,>=3.5.0->en-core-web-sm==3.5.0) (1.21.6)
Requirement already satisfied: packaging>=20.0 in /opt/conda/lib/python3.9/site-
packages (from spacy<3.6.0,>=3.5.0->en-core-web-sm==3.5.0) (21.3)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in
/opt/conda/lib/python3.9/site-packages (from spacy<3.6.0,>=3.5.0->en-core-web-
sm==3.5.0) (1.0.4)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in
/opt/conda/lib/python3.9/site-packages (from spacy<3.6.0,>=3.5.0->en-core-web-
sm==3.5.0) (3.0.12)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in
/opt/conda/lib/python3.9/site-packages (from spacy<3.6.0,>=3.5.0->en-core-web-
sm==3.5.0) (2.4.5)
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in
/opt/conda/lib/python3.9/site-packages (from spacy<3.6.0,>=3.5.0->en-core-web-
sm==3.5.0) (3.3.0)
```

```
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in
/opt/conda/lib/python3.9/site-packages (from spacy<3.6.0,>=3.5.0->en-core-web-
sm==3.5.0) (1.1.1)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in
/opt/conda/lib/python3.9/site-packages (from spacy<3.6.0,>=3.5.0->en-core-web-
sm==3.5.0) (4.64.0)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in
/opt/conda/lib/python3.9/site-packages (from spacy<3.6.0,>=3.5.0->en-core-web-
sm==3.5.0) (1.0.9)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in
/opt/conda/lib/python3.9/site-packages (from spacy<3.6.0,>=3.5.0->en-core-web-
sm==3.5.0) (2.0.7)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in
/opt/conda/lib/python3.9/site-packages (from spacy<3.6.0,>=3.5.0->en-core-web-
sm==3.5.0) (3.0.8)
Requirement already satisfied: pathy>=0.10.0 in /opt/conda/lib/python3.9/site-
packages (from spacy<3.6.0,>=3.5.0->en-core-web-sm==3.5.0) (0.10.1)
Requirement already satisfied: typer<0.8.0,>=0.3.0 in
/opt/conda/lib/python3.9/site-packages (from spacy<3.6.0,>=3.5.0->en-core-web-
sm==3.5.0) (0.7.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/opt/conda/lib/python3.9/site-packages (from
packaging>=20.0->spacy<3.6.0,>=3.5.0->en-core-web-sm==3.5.0) (3.0.8)
Requirement already satisfied: typing-extensions>=4.2.0 in
/opt/conda/lib/python3.9/site-packages (from
pydantic!=1.8,!=1.8.1,<1.11.0,>=1.7.4->spacy<3.6.0,>=3.5.0->en-core-web-
sm==3.5.0) (4.4.0)
Requirement already satisfied: charset-normalizer~=2.0.0 in
/opt/conda/lib/python3.9/site-packages (from
requests<3.0.0,>=2.13.0->spacy<3.6.0,>=3.5.0->en-core-web-sm==3.5.0) (2.0.12)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.9/site-
packages (from requests<3.0.0,>=2.13.0->spacy<3.6.0,>=3.5.0->en-core-web-
sm==3.5.0) (3.3)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/conda/lib/python3.9/site-packages (from
requests<3.0.0,>=2.13.0->spacy<3.6.0,>=3.5.0->en-core-web-sm==3.5.0) (2022.9.24)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/opt/conda/lib/python3.9/site-packages (from
requests<3.0.0,>=2.13.0->spacy<3.6.0,>=3.5.0->en-core-web-sm==3.5.0) (1.26.9)
Requirement already satisfied: blis<0.8.0,>=0.7.8 in
/opt/conda/lib/python3.9/site-packages (from
thinc<8.2.0,>=8.1.0->spacy<3.6.0,>=3.5.0->en-core-web-sm==3.5.0) (0.7.9)
Requirement already satisfied: confection<1.0.0,>=0.0.1 in
/opt/conda/lib/python3.9/site-packages (from
thinc<8.2.0,>=8.1.0->spacy<3.6.0,>=3.5.0->en-core-web-sm==3.5.0) (0.0.4)
Requirement already satisfied: click<9.0.0,>=7.1.1 in
/opt/conda/lib/python3.9/site-packages (from
typer<0.8.0,>=0.3.0->spacy<3.6.0,>=3.5.0->en-core-web-sm==3.5.0) (8.1.3)
```

```
Requirement already satisfied: MarkupSafe>=2.0 in /opt/conda/lib/python3.9/site-
packages (from jinja2->spacy<3.6.0,>=3.5.0->en-core-web-sm==3.5.0) (2.1.1)
Installing collected packages: en-core-web-sm
Successfully installed en-core-web-sm-3.5.0
  Download and installation successful
You can now load the package via spacy.load('en_core_web_sm')
Requirement already satisfied: nltk in /opt/conda/lib/python3.9/site-packages
(3.6.7)
Requirement already satisfied: regex>=2021.8.3 in /opt/conda/lib/python3.9/site-
packages (from nltk) (2022.10.31)
Requirement already satisfied: joblib in /opt/conda/lib/python3.9/site-packages
(from nltk) (1.2.0)
Requirement already satisfied: tqdm in /opt/conda/lib/python3.9/site-packages
(from nltk) (4.64.0)
Requirement already satisfied: click in /opt/conda/lib/python3.9/site-packages
(from nltk) (8.1.3)
```

[11]:

```
#Creating file to store all people spoken about, and main characters who spoke␣
 ↪about them
header="main_characters\tperson\n"
file_content=header

for i in range(len(main_characters)):
    name=main_characters[i]
    character_dialog=hp_df_final[hp_df_final['Character Name']==name]
    str_dialog = ' '.join(character_dialog['Dialogue'])
    doc=nlp(str_dialog)
    named_entities = [(ent.text, ent.label_)
                      for ent in doc.ents]
    people = [word
              for (word, tag) in named_entities
              if (tag == 'PERSON') and (not word.startswith('\n'))]
    if len(people) > 2:
        for person in people:
            file_content=file_content + name + "\t" + person + "\n"
```

[12]:

```
f = open("Main_Characters.tsv", "w")
f.write(file_content)
f.close()
relationship=pd.read_csv('Main_Characters.tsv', sep='\t',header=0)
```

[13]:

```
#Defining Function to split full name into first name and last name
def cut_name(name):
    parts_of_name=str.split(name)
    return parts_of_name
```

[14]:

```
#Defining Function to identify whether these two names mean the same person
def compare_name(name,parts_of_name):
    flag=parts_of_name in name
    return flag
```

[15]:

```
#Transforming main characters' different titles into their full name
for person_name in relationship['person']:
    for character in main_characters:
        for string in cut_name(character):
            if compare_name(person_name,string)==True:
                relationship=relationship.replace(person_name,character)
```

[16]:

```
#Tom Riddle is Voldemort
relationship=relationship.replace('Tom Riddle','Voldemort')
```

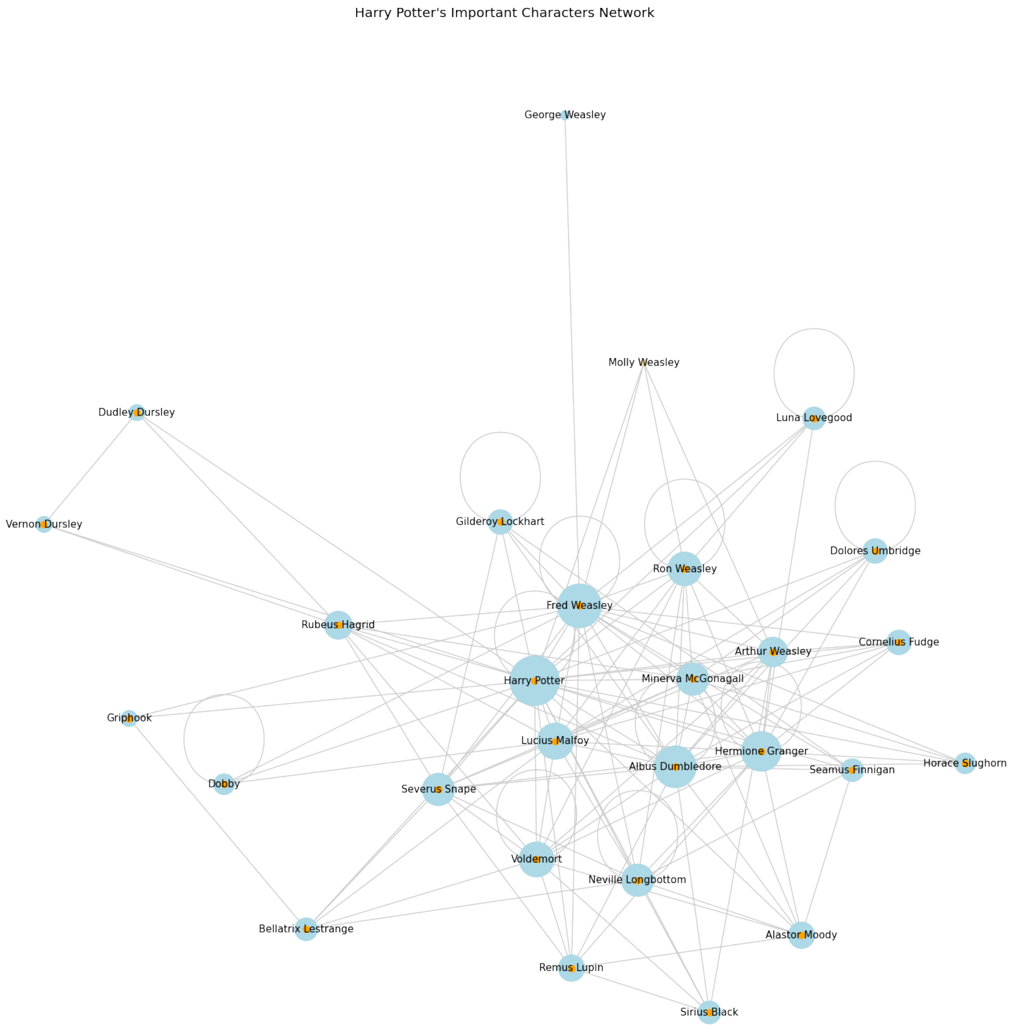
[17]:

```
#Selecting main characters to draw their network gragh
relationship=relationship[relationship["person"].isin(main_characters)]
```

[18]:

```
#Visualizing a network
import matplotlib.pyplot as plt
import networkx as nx
g = nx.from_pandas_edgelist(relationship, source='main_characters',␣
 ↪target='person')

characters = list(relationship.main_characters.unique())
people = list(relationship.person.unique())

g.degree("Reagan")

plt.figure(figsize=(30, 30))
layout = nx.spring_layout(g, k = 0.5)
person_size = [g.degree(person) * 200 for person in people]

nx.draw_networkx_nodes(g,
                       layout,
                       nodelist=people,
                       node_color='lightblue',
                       node_size=person_size)

nx.draw_networkx_nodes(g, layout, nodelist=characters, node_color='wheat',␣
 ↪node_size=100)
popular_characters = [name for name in people if g.degree(name) > 1 ]
nx.draw_networkx_nodes(g, layout, nodelist=popular_characters,␣
 ↪node_color='orange', node_size=100)
nx.draw_networkx_edges(g, layout, width=1.5, edge_color="#cccccc")
nx.draw_networkx_labels(g, layout, font_size=15)
```

```
plt.axis('off')
plt.title("Harry Potter's Important Characters Network", fontsize=20)
plt.show()
```



Most film stories progress through characters, and the scriptwriter or director narrates the plot and character relationships through character dialogue. As a result, dialogue aids in the formation of character networks and distinguishing main characters.

This complicated **Harry Potter's Important Characters Network** links the speaker to the person mentioned. It is undeniable that the film's main characters are mentioned more frequently, such as Harry Potter, Hermione Granger, Ron Weasley, and Albus Dumbledore (Voldemort is not mentioned as much as others, since he is commonly referred to as "You-Know-Who" or "He Who Must Not Be Named"). These central characters are critical to the plot's development.

Audiences tend to watch movies with a broad worldview and interwoven relationships between characters with diverse personality and motivation, which cause more conflicts.

1.1.3 Visualizing Character Arc

```
[19]: #Creating List of Most Popular Characters with at least 100 dialogues - https://
       ↪movieweb.com/best-harry-potter-characters/
      most_popular_characters = ['Hermione Granger', 'Ron Weasley', 'Minerva␣
       ↪McGonagall', 'Voldemort', 'Harry Potter',
                                  'Neville Longbottom', 'Severus Snape', 'Draco␣
       ↪Malfoy', 'Rubeus Hagrid',
                                   'Albus Dumbledore']
      print('Created a list of 10 Most Popular Characters with a minimum of 100␣
       ↪dialogues')
```

```
Created a list of 10 Most Popular Characters with a minimum of 100 dialogues
```

```
[20]: #Filtering Dataset to select diaglogues by most popular characters
      hp_df_filtered = hp_df_final[hp_df_final['Character Name'].
       ↪isin(most_popular_characters)]
      print('Created a filtered dataset for the 10 Characters')
      print("The data has "+str(hp_df_filtered.shape[0])+" dialogues")
```

```
Created a filtered dataset for the 10 Characters
The data has 5028 dialogues
```

```
[21]: #Installing Libry to Detect Emotions - https://pypi.org/project/NRCLex/
      !pip install NRCLex
```

```
Collecting NRCLex
  Using cached NRCLex-4.0-py3-none-any.whl (4.4 kB)
  Using cached NRCLex-3.0.0-py3-none-any.whl
Requirement already satisfied: textblob in /opt/conda/lib/python3.9/site-
packages (from NRCLex) (0.15.3)
Requirement already satisfied: nltk>=3.1 in /opt/conda/lib/python3.9/site-
packages (from textblob->NRCLex) (3.6.7)
Requirement already satisfied: click in /opt/conda/lib/python3.9/site-packages
(from nltk>=3.1->textblob->NRCLex) (8.1.3)
Requirement already satisfied: joblib in /opt/conda/lib/python3.9/site-packages
(from nltk>=3.1->textblob->NRCLex) (1.2.0)
Requirement already satisfied: regex>=2021.8.3 in /opt/conda/lib/python3.9/site-
packages (from nltk>=3.1->textblob->NRCLex) (2022.10.31)
Requirement already satisfied: tqdm in /opt/conda/lib/python3.9/site-packages
(from nltk>=3.1->textblob->NRCLex) (4.64.0)
Installing collected packages: NRCLex
Successfully installed NRCLex-3.0.0
```

```
[22]: from nrclex import NRCLex
```

[23]:
```
#Defining Function to Capture Dictionary of Emotions & Counts for each dialogue
def assess_emotions(text):
    emotions = NRCLex(text.lower())
    dict_of_emotions = emotions.raw_emotion_scores
    return dict_of_emotions
```

[24]:
```
#Applying Abovementioned Function
hp_df_filtered['Dictionary of Emotions'] = hp_df_filtered['Dialogue'].
 ↪map(assess_emotions)
print('Created a Column with a Dictionary of Emotions & Counts detected')
```

```
Created a Column with a Dictionary of Emotions & Counts detected
```

[25]:
```
#Storing List of Emotions for Use Later
list_of_emotions = ['fear', 'anger', 'anticipation', 'trust', 'surprise',␣
 ↪'sadness', 'disgust', 'joy']
```

[26]:
```
#Defining Function to convert dictionary to columns for emotions with counts
def get_count_of_emotions(text, emotion):
    if emotion in text.keys():
        count = text[emotion]
        return count
    else:
        return 0
```

[27]:
```
#Applying Function to get columns for emotions
for feeling in list_of_emotions:
    hp_df_filtered[feeling] = hp_df_filtered['Dictionary of Emotions'].
 ↪apply(get_count_of_emotions, emotion=feeling)
    print('Scanning for '+feeling+' completed.')
```

```
Scanning for fear completed.
Scanning for anger completed.
Scanning for anticipation completed.
Scanning for trust completed.
Scanning for surprise completed.
Scanning for sadness completed.
Scanning for disgust completed.
Scanning for joy completed.
```

[28]:
```
hp_df_filtered['sum_of_emotions'] =␣
 ↪hp_df_filtered['fear']+hp_df_filtered['anger']+\
                                                  ␣
 ↪hp_df_filtered['anticipation']+hp_df_filtered['trust']+\
                                                  ␣
 ↪hp_df_filtered['surprise']+hp_df_filtered['sadness']+\
                                                    ␣
 ↪hp_df_filtered['disgust']+hp_df_filtered['joy']
print('Created a column to calculate sum of frequency of emotions')
```

```
Created a column to calculate sum of frequency of emotions
```

[29]:

```
hp_df_emotion_counts = hp_df_filtered[['Movie Title','Character␣
 ↪Name','fear','anger','anticipation','trust',
                                                    ␣
 ↪'surprise','sadness','disgust','joy','sum_of_emotions']].\
                                    groupby(['Movie Title','Character Name']).
 ↪sum()
print('Grouped by Movie & Character to get Aggregated Occurrences')

hp_df_emotion_counts_filtered =␣
 ↪hp_df_emotion_counts[hp_df_emotion_counts['sum_of_emotions'] >= 10].\
                                drop(columns = 'sum_of_emotions')
print('Dropping Cases where a character had <10 dialogues in a movie to␣
 ↪maintain validity of results')


hp_df_emotion_summary= hp_df_emotion_counts_filtered.
 ↪div(hp_df_emotion_counts_filtered.sum(axis=1), axis=0).\
                        reset_index()
print('Created a Summary for each Character for each movie with % of emotions␣
 ↪exhibited')
```

```
Grouped by Movie & Character to get Aggregated Occurrences
Dropping Cases where a character had <10 dialogues in a movie to maintain
validity of results
Created a Summary for each Character for each movie with % of emotions exhibited
```

[30]:

```
#Plotting Character Arcs Across Movies for Most Popular Characters
fig=px.bar(hp_df_emotion_summary,
           y="Movie Title",
           x=['fear','anger','sadness',␣
 ↪'disgust','anticipation','trust','surprise','joy'],
           title="Mini-Report Graph 1: Character Arc Across Movies for",
           orientation='h',
           facet_col='Character Name',
           facet_col_wrap=2,
           facet_row_spacing=0.05,
           width=1000,
           height=1000,
           category_orders={'Character Name' :['Hermione Granger', 'Ron␣
 ↪Weasley', 'Minerva McGonagall', 'Voldemort', 'Harry Potter',
                           'Neville Longbottom', 'Severus Snape', 'Draco␣
 ↪Malfoy', 'Rubeus Hagrid',
                            'Albus Dumbledore'],
                            'Movie Name' :["1 Harry Potter and the␣
 ↪Philosopher's Stone",
                            '2 Harry Potter and the Chamber of Secrets',
                            '3 Harry Potter and the Prisoner of Azkaban',
                            '4 Harry Potter and the Goblet of Fire',
                            '5 Harry Potter and the Order of the Phoenix',
                            '6 Harry Potter and the Half-Blood Prince',
                            '7 Harry Potter and the Deathly Hallows Part 1',
                            '8 Harry Potter and the Deathly Hallows Part 2']})


fig.show()
```

This graph depicts the percentage of eight emotions (fear, anger, anticipation, trust, surprise, sadness, disgust, joy) that change across movies for ten popular characters. Significant changes are shown in character arcs of **Draco Malfoy** and **Neville Longbottom**, so we will draw line charts to research further.

[31]:

```
df = hp_df_emotion_summary[hp_df_emotion_summary['Character Name']=='Draco␣
 ↪Malfoy']
fig = px.line(df, x="Movie Title", y=['anger','anticipation','trust','sadness'],
             color_discrete_sequence=['#EF553B', '#FFA15A',␣
 ↪'#19D3F3','#00CC96'],
             labels = {'value' : '%Emotions'},
             title="Mini Report Graph 2 : Draco's Character Arc",width=900)
fig.show()
```

[32]:

```
df = hp_df_emotion_summary[hp_df_emotion_summary['Character Name']=='Neville␣
 ↪Longbottom']
fig = px.line(df, x="Movie Title", y=['anger','fear','joy'],
             color_discrete_sequence=['#EF553B','#636EFA','#B6E880'],
             labels = {'value' : '%Emotions'},
             title="Mini Report Graph 3: Neville's Character Arc", width=900)
fig.show()
```

Throughout the movies, we see **Neville** become more fearless & happier. We see a spike in his anger, but one of the reasons why the fans adore him might be attributed to the fact that he makes efforts to overcome his fear instead of keeping timid and ends up as a very balanced character.

Malfoy has been the anti-hero right from the start, yet fans have sympathised with him. Over the seasons we see his anticipation and trust reduce, maybe as he uncovers the truth about his family. His anger & sadness increases over the seasons, adding more depth to his character.

Neville and Draco are not the main protagonists, but they were still loved by the audience. This can be used to develop new characters.

2 Mini-Report

2.0.1 Problem Statement:

We've studied the character arcs of the 10 most popular Harry Potter movie characters & the relationship between the 30 characters with the most dialogues. Our goal is to draw insights about their character arcs - including the change in their emotions & their interaction with others to understand what the audience likes. These insights can be used to model new characters for upcoming shows.

2.0.2 Insights Drawn

Create a set up with interwoven relationships Audiences tend to watch movies with a broad worldview and interwoven relationships between characters with diverse personality and motivation, which cause more conflicts. **Give a character advantages and flaws** Characters don't have to be perfect. With highlights and weaknesses, a character comes to life and makes the audience feel closer. **Create a character with significant arc.** When the characters experience significant events that motivate them to transform, it is more likely to audiences to empathise instead of focusing on a stereo-types without freshness. **Create a character with rich inner conflicts** to make the character arc more convincing. As seen in Graph 2, Malfoy's complicated background and contradictory state of thoughts attracts most audiences, since it is a reflection of struggle in the real world on these characters, as well as the embodiment of people's inner contradictions. Additionally, the changes of characters draw more attentions by **subverting audiences expectation**. Neville (Graph 3), for example, is becoming fearless to show his feelings and motivations, which audiences haven't expected. However, a static secondary characters (Graph 1) whose arc keeps relatively flat can be set to illuminate others.