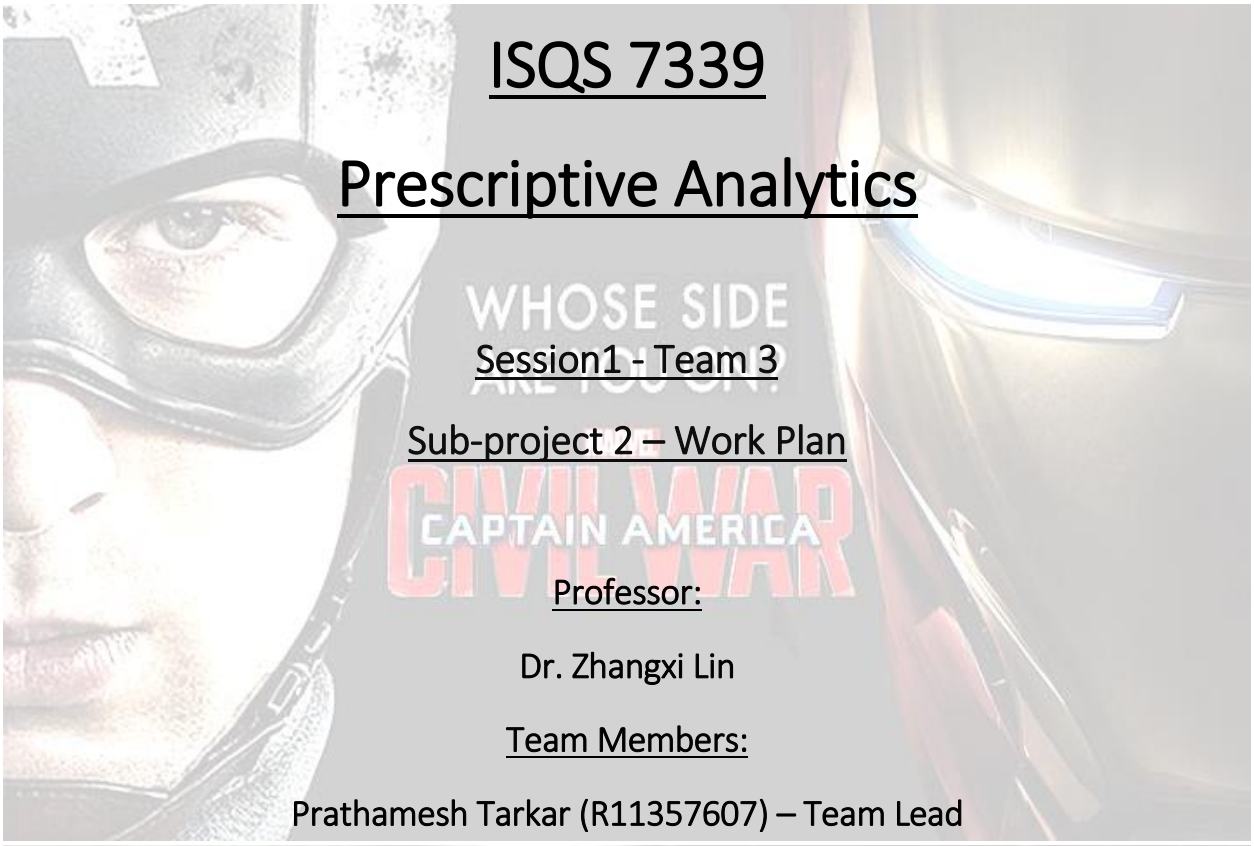
ISQS 7339

Prescriptive Analytics



Session1 - Team 3

Sub-project 2 – Work Plan

Professor:

Dr. Zhangxi Lin

Team Members:

Prathamesh Tarkar (R11357607) – Team Lead

Fazil Sheik Mohammed (R11360243)

Praveena Kathera (R11372933)

Tiju Mathew George (R11287760)

Social network title – Facebook

Project title – Comparing Facebook comments for Movies

Data category – Comments in Facebook

12/07/2015

Contents

1 Social network to be analyzed

We are planning to use real-time Facebook data to analyze social plugins like posts, comments, shares etc. on pages for movies or products etc. The analysis of the extracted data from the social plugins will help us to determine the latest trending buzz.

2 Specific topic to be focused

We are going to extract the data for 'shares', 'posts', 'comments' etc. posted on pages for movies in entire Facebook network. Based on this information we are trying to determine the most popular and trending facts which will help us to design or improve our business solutions in target marketing.

We are going to analyze the data gathered by the following social plugins:

Share Button: Allow people to share to Facebook, share with particular friends or with a group.

Comments: Allows people to comment on posts.

Posts: Allows people to write posts on any walls or other pages.

3 Category of the data to be collected

Shares, Comments and Posts are the category of the data we are planning to extract. These contain the data of people sharing, commenting on the contents available on Facebook and it determines the sentiment and social view of the public which will immensely help in determining and designing solutions for the market to be targeted or competitive advantage.

If a person shares, comments or posts on a page we can say that they are interested on that particular content. We can cash this interests and design solutions for business problems like customer segmentation, target marketing and competitive advantage etc.

4 Motivation:

The motivation of our project is to analyze the comments from pages for two upcoming famous movies **Batman v Superman: Dawn of Justice** (batmanvsuperman) and **Captain America: Civil War** (captainamericacw). The popularity of the page depends on various aspects like:

- Comments on Popularity
- Comments on Actors
- Comments on Director
- Comments on Characters in the movie

- Comments on Production Houses of the movie

By using the above mentioned aspects, we are trying to help understand target marketing and competitive advantage for movies. A model is being built which will relate all the relevant comments and help us understand its competitors.

5 Objective:

The objective of the project is to come to an inference on different aspects which would be significant in determining the popularity of movies on social media like Facebook. This would help the movie makers in designing their working and deployment pattern on social media.

We went on to define our objective in terms of below aspects:

Comparing positive and negative comments for both the Movies: We tried to analyze all the posts and their comments using TextBlob. Based on the number of positive and negative comments, we can talk about the most popular movie.

Comparing comments on Actors of the Movies: Also we tried to check the popularity of Actors in Movies so as to decide which one has the most popularity (positive/negative/neutral) and can be used for marketing. For that purpose, we use Python lists having relevant keywords for the actors.

Comparing comments on Directors of the Movies: We tried to check the popularity of Directors in Movies so as to decide which one has the most popularity (positive/negative/neutral) and can be used for marketing. For that purpose, we use Python lists having relevant keywords for the directors.

Comparing comments on Characters of the Movies: Also we tried to check the popularity of Characters in Movies so as to decide which one has the most popularity (positive/negative/neutral) and can be used for marketing. For that purpose, we use Python lists having relevant keywords for the characters.

Comparing comments on Production Houses of the Movies: We tried to check the popularity of Production Houses in Movies so as to decide which one has the most popularity (positive/negative/neutral) and can be used for marketing. For that purpose, we use Python lists having relevant keywords for the production houses.

We intend to achieve to the following objectives in our analysis:

- Compare the overall popularity between the two movies
- Compare the buzz around the actors involved in the two movies viz. Ben Affleck, Robert Downey Jr., Chris Evans, and Henry Cavill
- Compare the buzz around the directors of the movies viz. Zack Snyder
- Compare the popularity among the characters in the two movies- Batman, Superman, Iron man, Captain America
- Compare the buzz around the production houses of the two movies- DC and Marvel

6 Project Overview:

The project focus is on the sentiment analysis of the comments or posts on the Facebook community page of the movies. We are trying to compare the comments of two different upcoming movies that are popular. The movies are Batman v Superman: Dawn of Justice and Captain America: Civil War. The comparison will be done with the help of aspects of sentiment analysis. We have decided upon few aspects and are trying to compare both movies on the basis of those aspects.

7 Data Set Description:

7.1 Data Availability

The data for the project is open source as we are retrieving it from the Facebook. The data is retrieved using the API and processed with Python. The data consists of the comments from two different Facebook movie pages, viz., Batman v Superman: Dawn of Justice and Captain America: Civil War. Data includes the aspects we are analyzing. It includes the comparison of music, actors, directors, competitors, characters, etc.

7.2 Data for Movies:

- Posts and Comments for movie - Batman v Superman: Dawn of Justice
- Posts and Comments for movie - Captain America: Civil War

7.3 Descriptions:

- **Posts and Comments for movie -** Batman v Superman: Dawn of Justice: The community page comments for this movie includes the views of people that conveys the sentiments of the public. We are using those sentiments to determine the polarity of the comment. The comments includes the likeness, popularity of the director, actors and critics from the viewers. Comparing the comment will determine the best among the two movies.

Here we have unstructured data in the form comments and posts. For the page – Batman v Superman movie, there are a total of 45 posts and 1125 comments overall. Also, we can see that 2491 unique words and a total of 7810 words in all comments.

- **Posts and Comments for movie - Captain America: Civil War**: Similar to the other movie this page also includes the sentiment comments of the people for the movie Captain America: Civil War. As mentioned above the comparison determines the final outcome of the movie. Here we have unstructured data in the form comments and posts. For the page – Captain America: Civil War movie, there are a total of 65 posts and 470 comments overall. Also, we can see that 1015 unique words and a total of 2611 words in all comments.

7.4 Data Quality:

The data quality for each of the provided datasets is as described below:

1. Posts and Comments for movie - Batman v Superman: Dawn of Justice: The comments had a lot of irrelevant content like random URL's, emoticons, random characters, extra spaces and newline characters unrelated to the context.
2. Posts and Comments for movie - Captain America: Civil War: The comments had a lot of irrelevant content like random URL's, emoticons, random characters, extra spaces and newline characters unrelated to the context.

7.5 Data Pre Processing Tasks

Data preprocessing is a technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues. Data preprocessing prepares raw data for further processing.

Data preprocessing is used database-driven applications such as customer relationship management and rule-based applications (like neural networks).

We are doing data preprocessing to do

- Data cleaning
- Data transformation
- Data separation

1. **Data Cleaning:** Using REGEX (Regular Expression) and JSON libraries in Python, we are replacing the extraneous characters, spaces and strings that are not relevant to understand the sentiments of comments for movie. Below is the code that we used to do this:

```
nohttp = two['message'].replace('\n', '')
nohttp = re.sub(r'http://\b.*',' site ',nohttp)
```

```
nohttp = re.sub(r'https://\b.*',' site ', nohttp)
```

Below are some screenshots of the same:

```
New Comment #886-->  You were wearing those same pants today Zack.
Old Comment #886-->  You were wearing those same pants today Zack.  □□□
```

```
New Comment #863-->  Gods among us  site
Old Comment #863-->  Gods among us https://www.facebook.com/photo.php?fbid=930917556938774&set=a.315720545125148.79692.100000
```

```
New Comment #748-->   Good Movie..Watch Batman v Superman Dawn of Justice Full Movie  Download Free   site
Old Comment #748-->  □□ Good Movie..Watch Batman v Superman: Dawn of Justice Full Movie □□ Download Free: ▸http://batmanvsuper
```

2. Data Transformation: We did not transform any comments as we took the comments in unstructured format and analyzed the comments using TextBlob library in Python.

3. Data Separation: We separated the comments from individual posts and analyzed the comments individually for the movie in consideration.

```
for one in g.get_connections(pepsi_id,"posts", limit=2500)['data']:
#for one in g.get_object('/walmart/' + 'posts', since='2013-01-01', until='2014-01-
10', limit=500)['data']:
    j = j + 1
    print("Post # "+str(j))
    for two in one['comments']['data']:
_________nohttp = two['message'].replace('\n', '')
```

7.6 Data Exploration:

After successfully carrying out the data cleansing and preprocessing tasks, we finally obtained the data exploration task. The results of various explorations are described below:

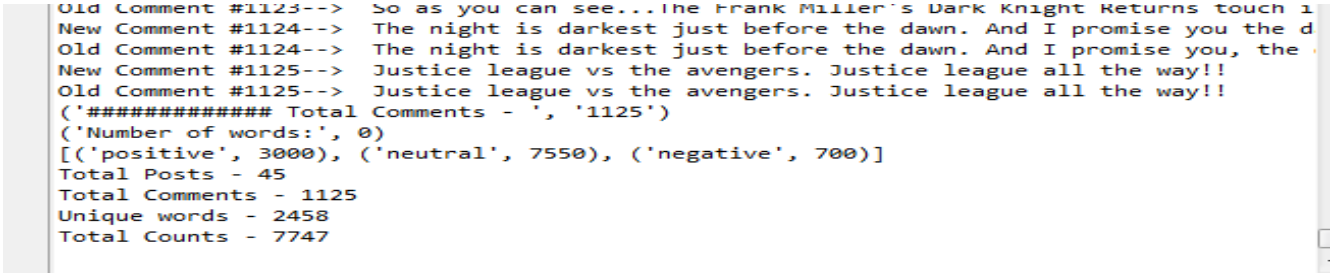
1. We can see that there are some random URL's, extra spaces, extra tabs, extra new line characters etc. Below is the screenshot for the same:

```
Old Comment #994-->  This movie is going to be 13/10
Old Comment #995-->  ??????? Yes!!!!
Old Comment #996-->  Wow!
Old Comment #997-->  Wonder Woman :* <3
Old Comment #998-->  I can't wait!! :D
Old Comment #999-->  AWESOME!
Old Comment #1000-->  so full of win!
('############## Total Comments - ', '1000')
Post # 41
Old Comment #1001-->  http://media.tumblr.com/68a52cbe7d6ec4d25a54da2ec4f0e158/tumblr_inline_mpxg68ajos1qz4rgp.gif
Old Comment #1002-->  Ben Affleck will do a great job.. just wait and see!!
Old Comment #1003-->  FIRST
Old Comment #1004-->  \O/
Old Comment #1005-->  He'll be the best Batman, I'm telling you.
Old Comment #1006-->  Looks good, veeeeeeeery good
Old Comment #1007-->  fatman..nahh :v
Old Comment #1008-->  Now for Wonder Woman.
Old Comment #1009-->  that chin :3
Old Comment #1010-->  Finally a true looking batman
Old Comment #1011-->  (y)
Old Comment #1012-->  tb wlahe catman msh batman b elmask da :D Marwa Mahmoud
Old Comment #1013-->  Such a dope picture!
Old Comment #1014-->  The Frank Miller and Jim Lee Batman live action. Can it be more awesome?
Old Comment #1015-->  I WANT BALE BACK!!!!
Old Comment #1016-->  It looks amazing
Old Comment #1017-->  Anthony McMullen
Old Comment #1018-->  I have the weirdest boner right now...
Old Comment #1019-->  From a daredevil to the batman, mmmmm
Old Comment #1020-->  For those complaining about his chin, I bet his chin alone would whoop yo @$$
Old Comment #1021-->  Michael Gonzalez Johnny Herrera
Old Comment #1022-->  Adam Spinola Alan Spinola Hassan BeFresh ben affleck batman actually looks cool
Old Comment #1023-->  Cynthia
Old Comment #1024-->  Yay yay
Old Comment #1025-->  why are the ears short??? :/
('############## Total Comments - ', '1025')
```

2. We tried getting exploring the size of data for both the movies

Batman v Superman: Dawn of Justice:

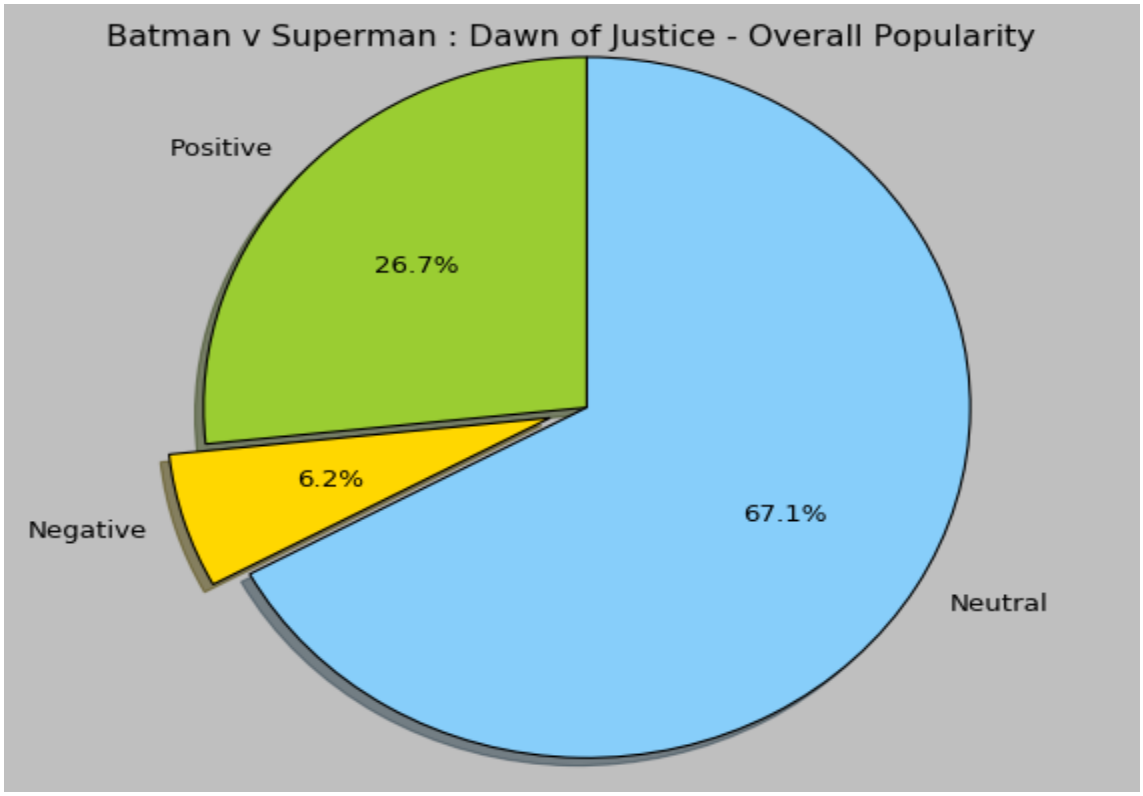
```
Total Posts - 45
Total Comments - 1125
Unique words - 2491
Total Counts – 7810
```

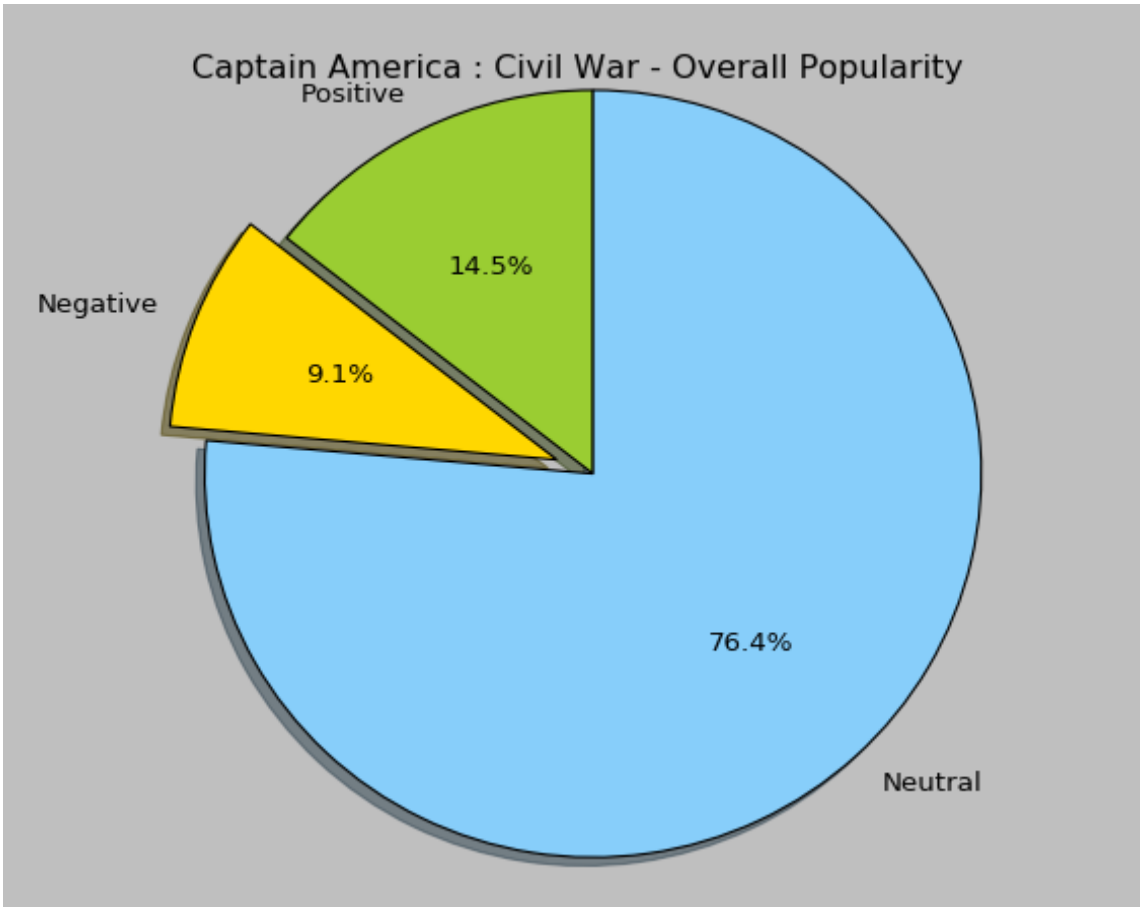


Captain America: Civil War

Total Posts - 65
Total Comments - 470
Unique words - 1015
Total Counts – 2611

```
Comment #466-->  :}
('############## Total Comments - ', '466')
Post # 65
Comment #467-->  FENOMENAL!
Comment #467-->  FENOMENAL!
Comment #468-->  Sound like Iron Man VS Captain America? with Black Panther in this movi
Comment #468-->  Sound like Iron Man VS Captain America? with Black Panther in this movi
Comment #469-->  cap is gonna die in this film . '
Comment #469-->  cap is gonna die in this film . :'(
Comment #470-->  Daniela Senesi
Comment #470-->  Daniela Senesi
('############## Total Comments - ', '470')
('Number of words:', 0)
[('positive', 680), ('neutral', 3590), ('negative', 430)]
Total Posts - 65
Total Comments - 470
Unique words - 991
Total Counts - 2582
```





7.7 Analysis Outputs

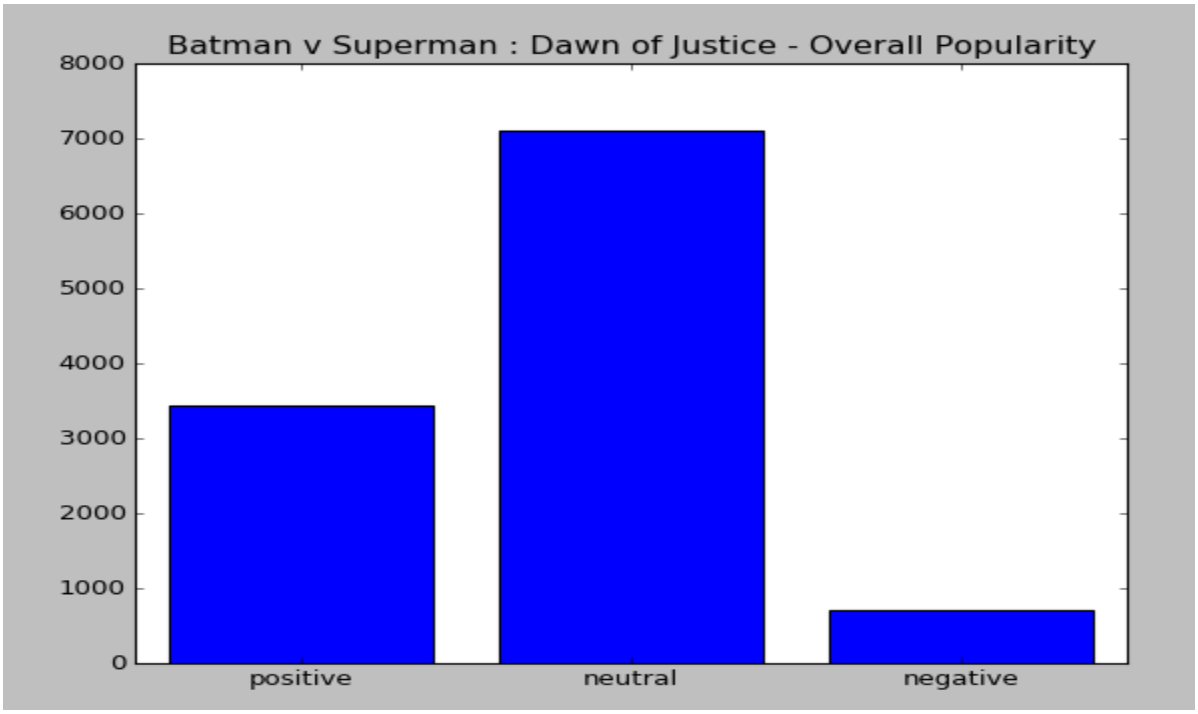
Through this project we are trying to answer the following questions of value for business.

1) Compare the overall popularity between the two movies.

Comparing the statistics of the two movies, we can see that though the total number of comments for Batman v Superman is more than Captain America. The posts on Captain America is more in comparison to Batman v Superman. The graph shows that Batman V Superman is more popular than Captain America. In both movies the positivity is more prevalent than the negativity. So overall the comments are neutral with a bend towards positive polarity.

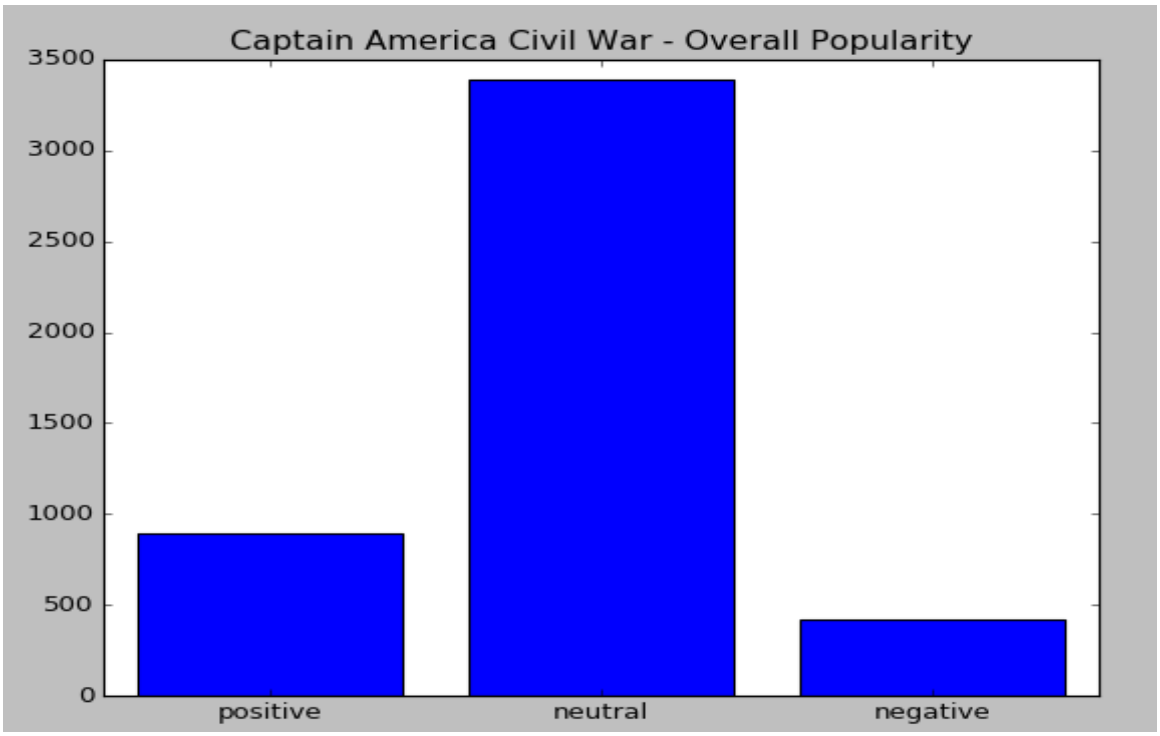
Batman v Superman:

```
% positive - 26.7777777778
% negative - 6.22222222222
% neutral – 67.1
```



Captain America:

% positive – 14.52333
% negative – 9.11121
% neutral – 76.42352



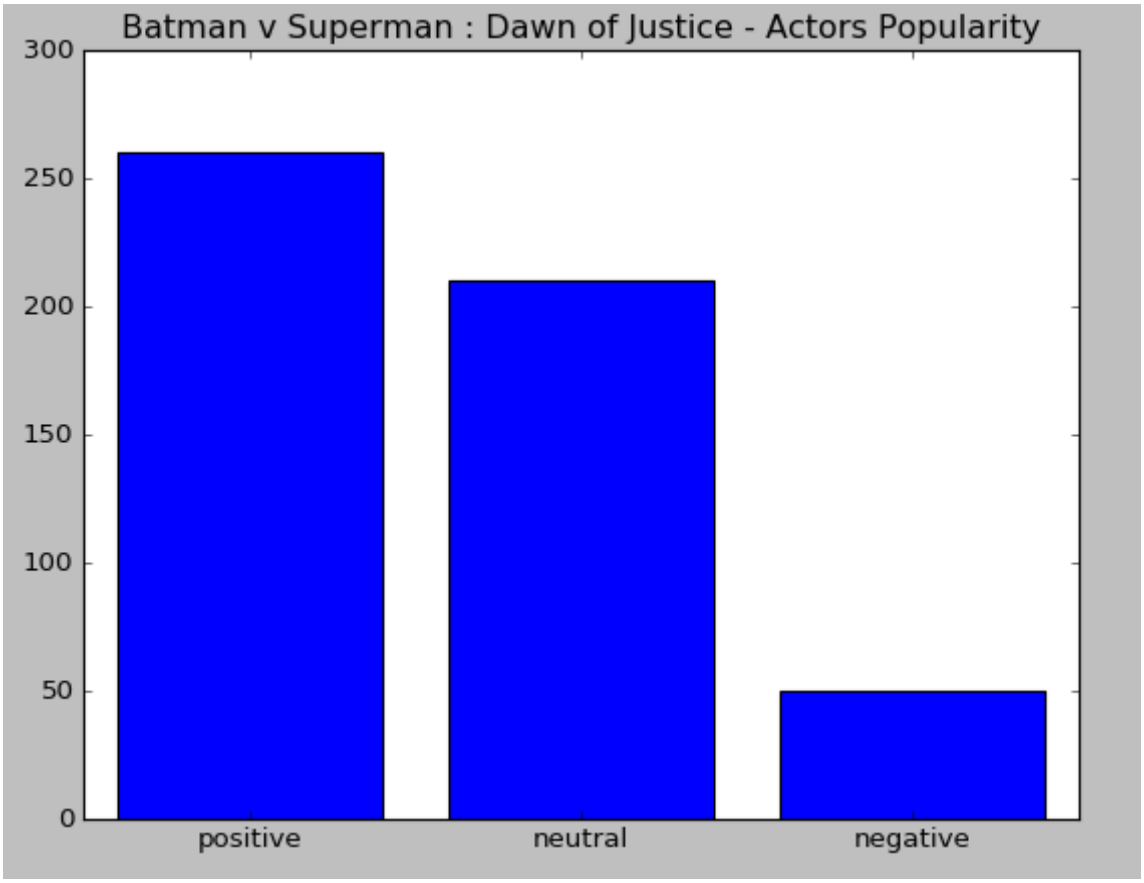
2) Compare the buzz around the actors involved in the two movies viz. Ben Affleck, Robert Downey Jr., Chris Evans, and Henry Cavill.

We compared the popularity of the actors of both the movies. We can see from the graphs that in Batman V Superman movie the comments for the actors are more positive polarized whereas if we see the graph for Captain America almost all comments are neutral. ***This study will help the director for future actor casting based on public sentiments.***

We used the following keywords for retrieving actors of movie Batman v Superman, 'ben', 'affleck', 'henry', 'cavill' and for the movie Captain America the keywords are 'robert', 'downey', 'chris', 'evans'.

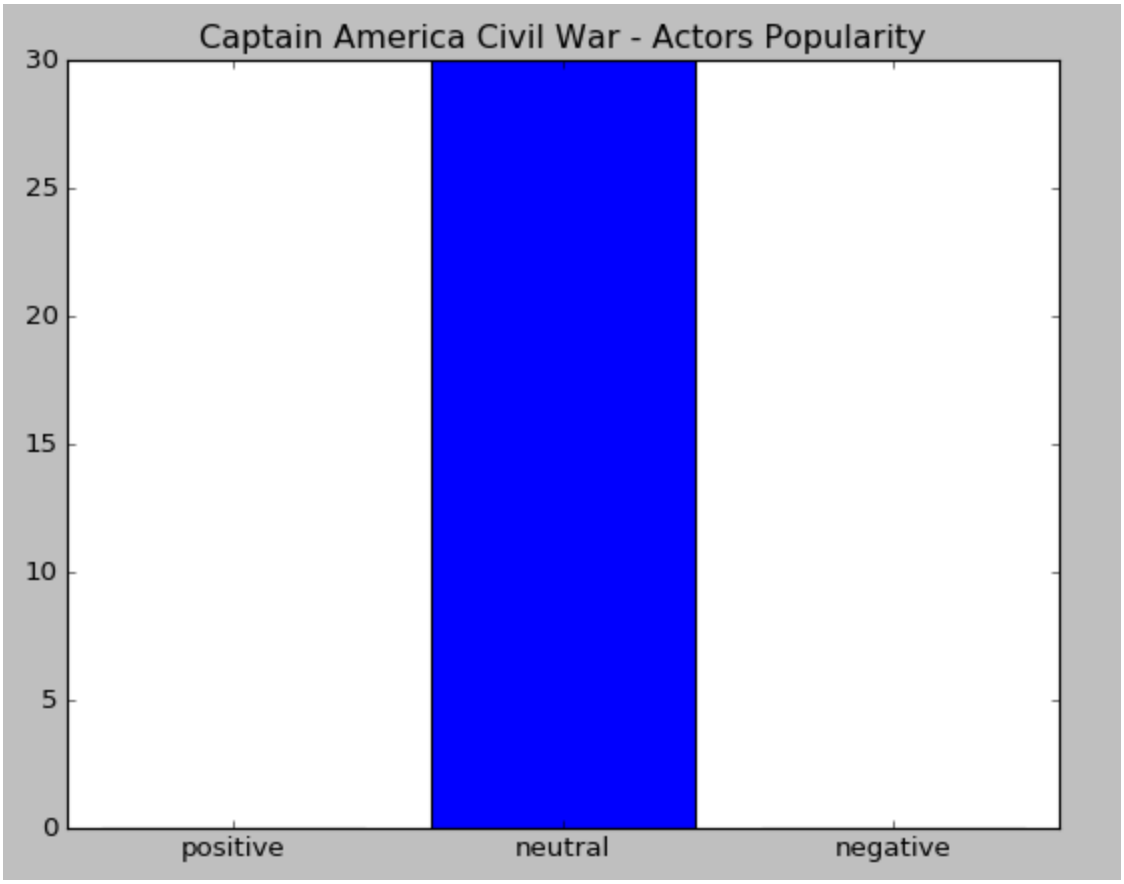
Batman v Superman:

```
dc_actors = ['ben', 'affleck', 'henry', 'cavill']
Total Posts - 45
Total Comments - 52
Unique words - 394
Total Counts - 932
% positive - 50.0
% negative - 9.61538461538
% neutral - 40.3846153846
```



Captain America:

```
mv_actors = ['robert', 'downey', 'chris', 'evans']
Total Posts - 65
Total Comments - 3
Unique words - 5
Total Counts - 5
% positive - 0.0
% negative - 0.0
% neutral - 100.0
```

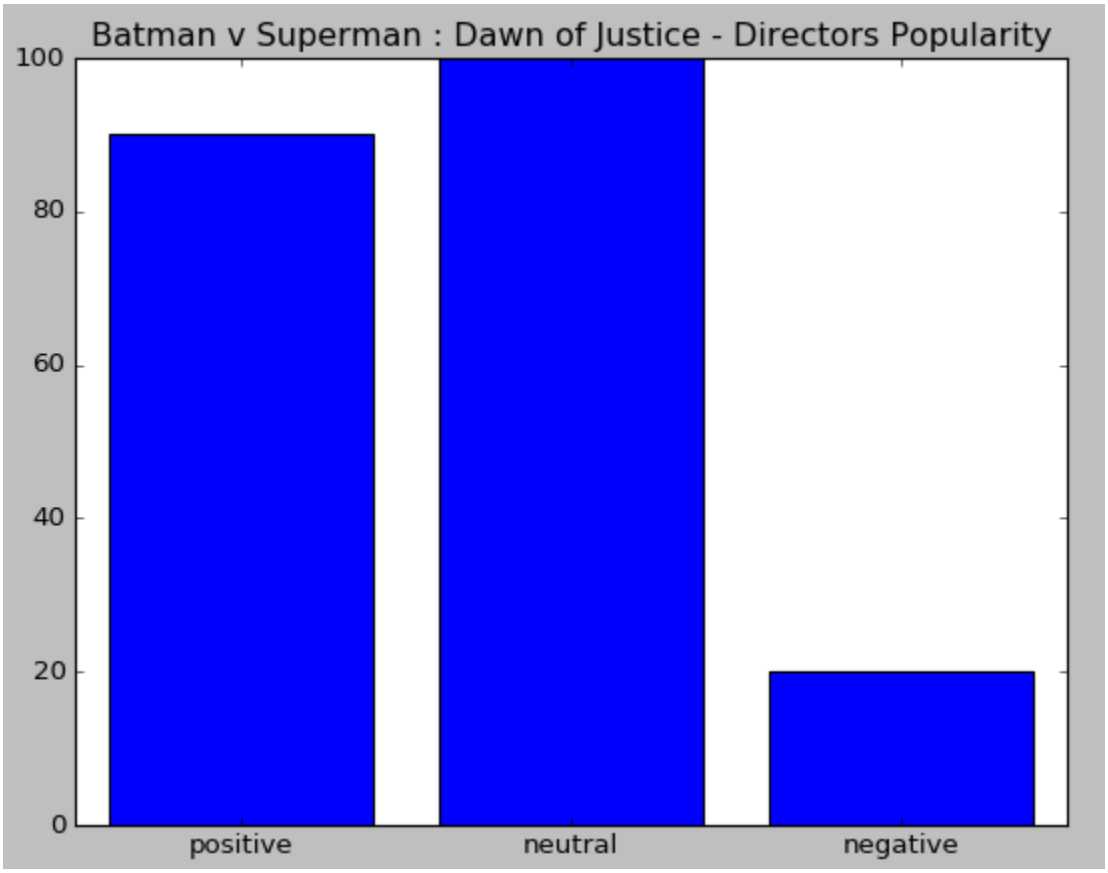


3) Compare the buzz around the directors of the movies viz. Zack Snyder.

These graphs determines the comments for the directors. The popularity of the directors can be estimated by the comments of the public. This study will help the general public to determine the collective view of the people regarding a particular director. This is a more robust way as the result when taken through public sentiment helps us to decide upon the director popularity. Comparing both movies we see that the director of Batman V superman has multiple polarity though positivity precedes the negative and in the movie Captain America as almost all neutral comments. This shows that Batman V Superman's director is more popular in this season.

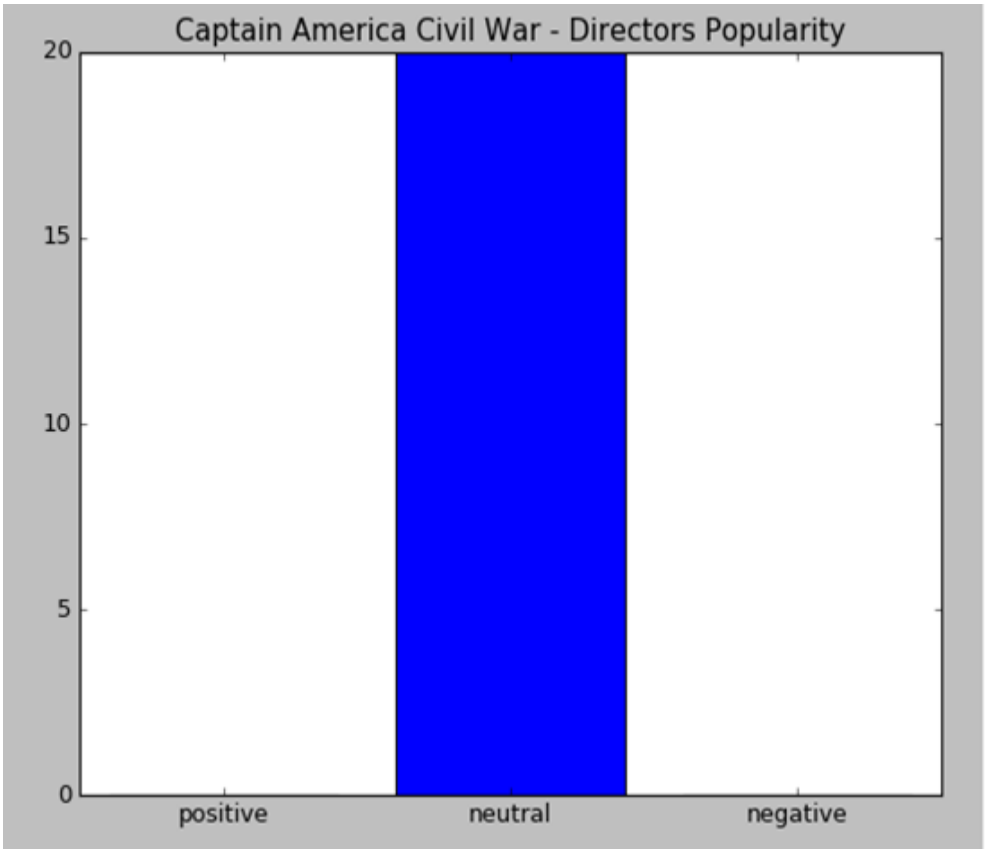
Batman v Superman:

```
dc_director = ['zack', 'snyder']
Total Posts - 45
Total Comments - 21
Unique words - 204
Total Counts - 331
% positive - 42.8571428571
% negative - 9.52380952381
% neutral - 47.619047619
```



Captain America:

```
mv_director = ['joe', 'russo', 'anthony']
Total Posts - 65
Total Comments - 2
Unique words - 6
Total Counts - 5
% positive - 0.0
% negative - 0.0
% neutral - 100.0
```

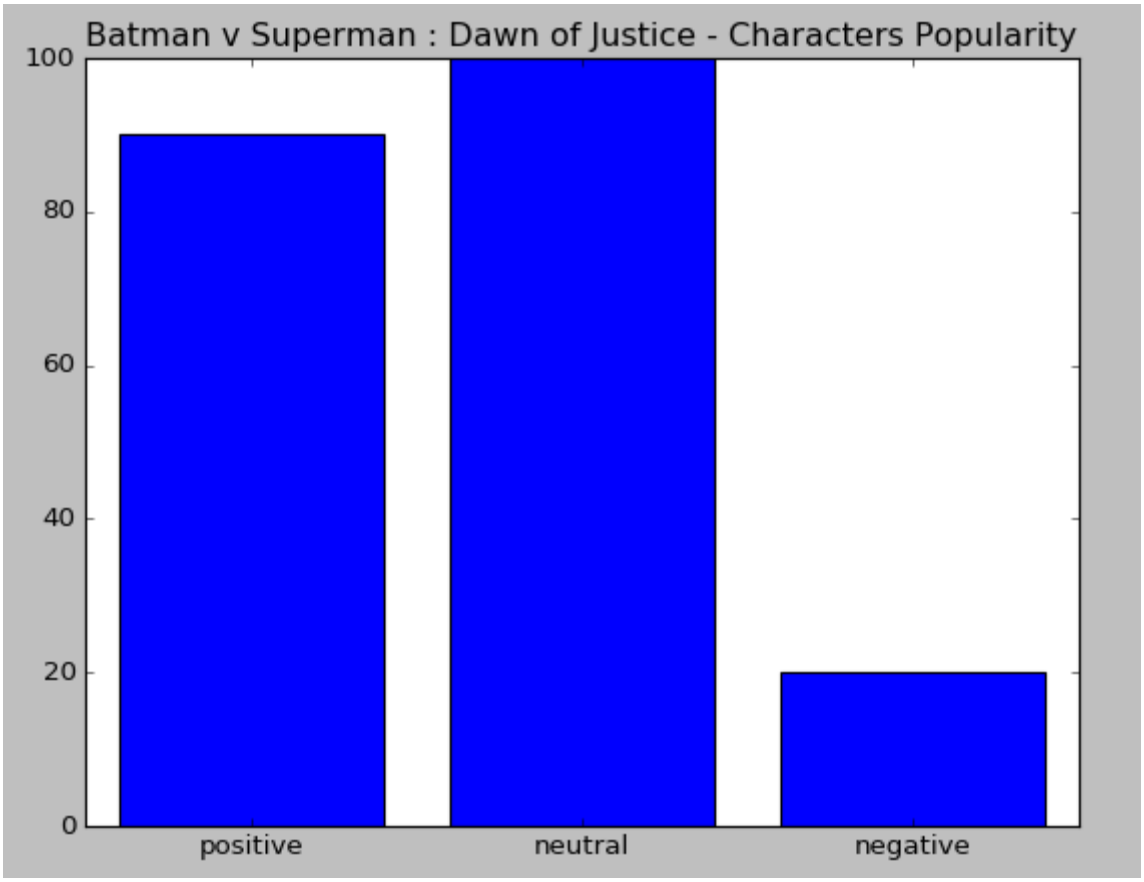


4) Compare the popularity among the characters in the two movies- Batman, Superman, Iron man, Captain America.

Comparing the characters of both movies from the above statistics and graph we can conclude that the characters of Batman v Superman is more popular than the characters of the Captain America. This shows the work of the marketing team that projects the characters along with the work of the actors that determines the popularity. In this aspect the Batman v Superman scored better.

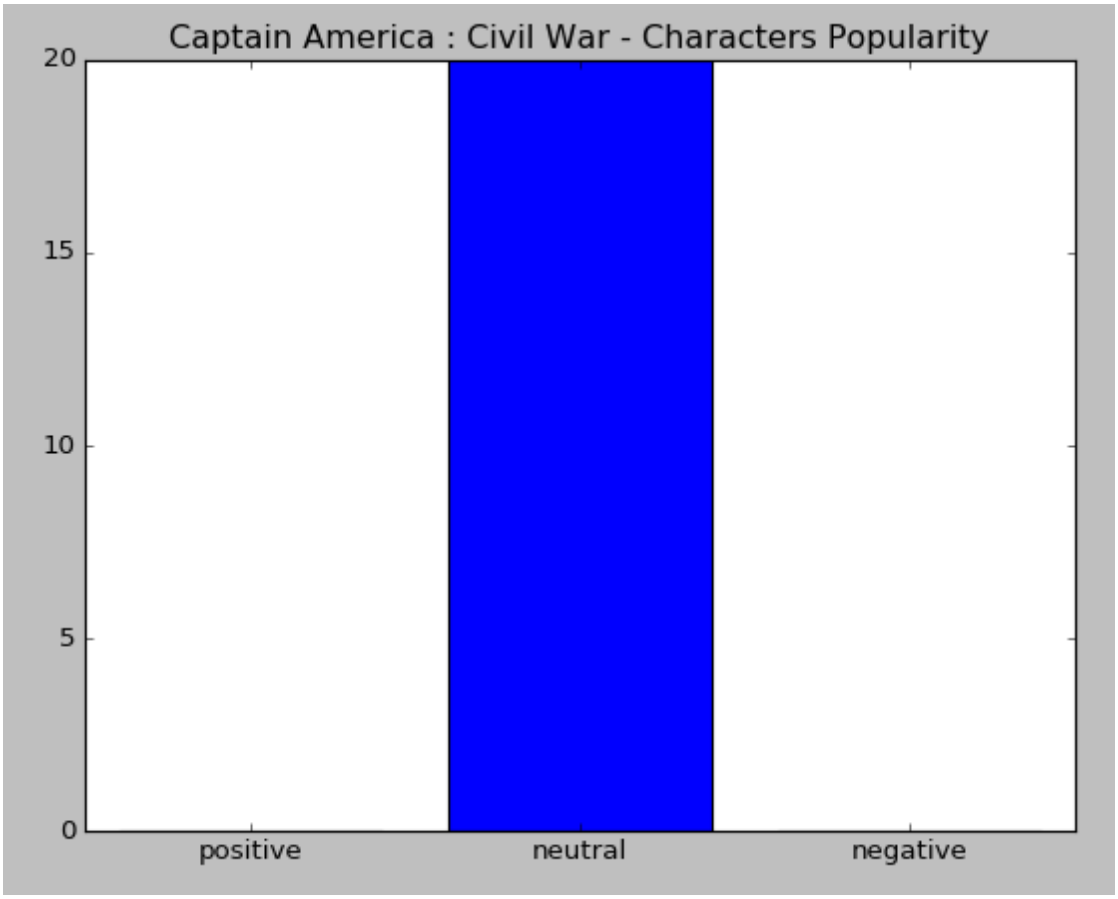
Batman v Superman:

```
dc_characters = ['superman', 'batman', 'wonder','knight', 'clark', 'kent', 'bruce',
'wayne']
Total Posts - 45
Total Comments - 21
Unique words - 204
Total Counts - 331
% positive - 42.8571428571
% negative - 9.52380952381
% neutral - 47.619047619
```



Captain America:

```
mv_characters = ['ironman', 'captain', 'america', 'tony', 'stark', 'rogers']
Total Posts - 65
Total Comments - 2
Unique words - 6
Total Counts - 5
% positive - 0.0
% negative - 0.0
% neutral - 100.0
```

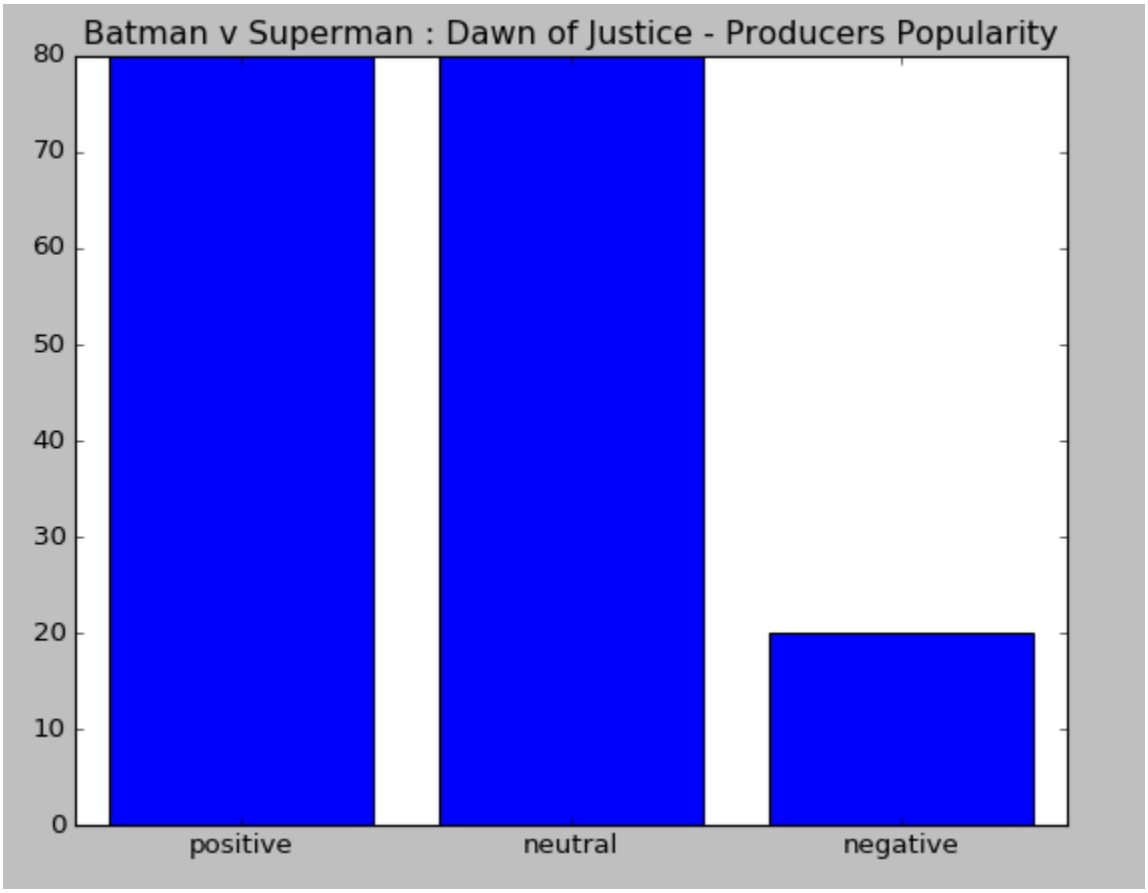


5) Compare the buzz around the production houses of the two movies- DC and Marvel.

Comparing the comments from the above statistics we can conclude that even in this aspect Batman V Superman scored better than Captain America. Both movies have more positive comments but Captain America has no neutral comments for this aspect so it shows people are clearer about their sentiments on Captain America.

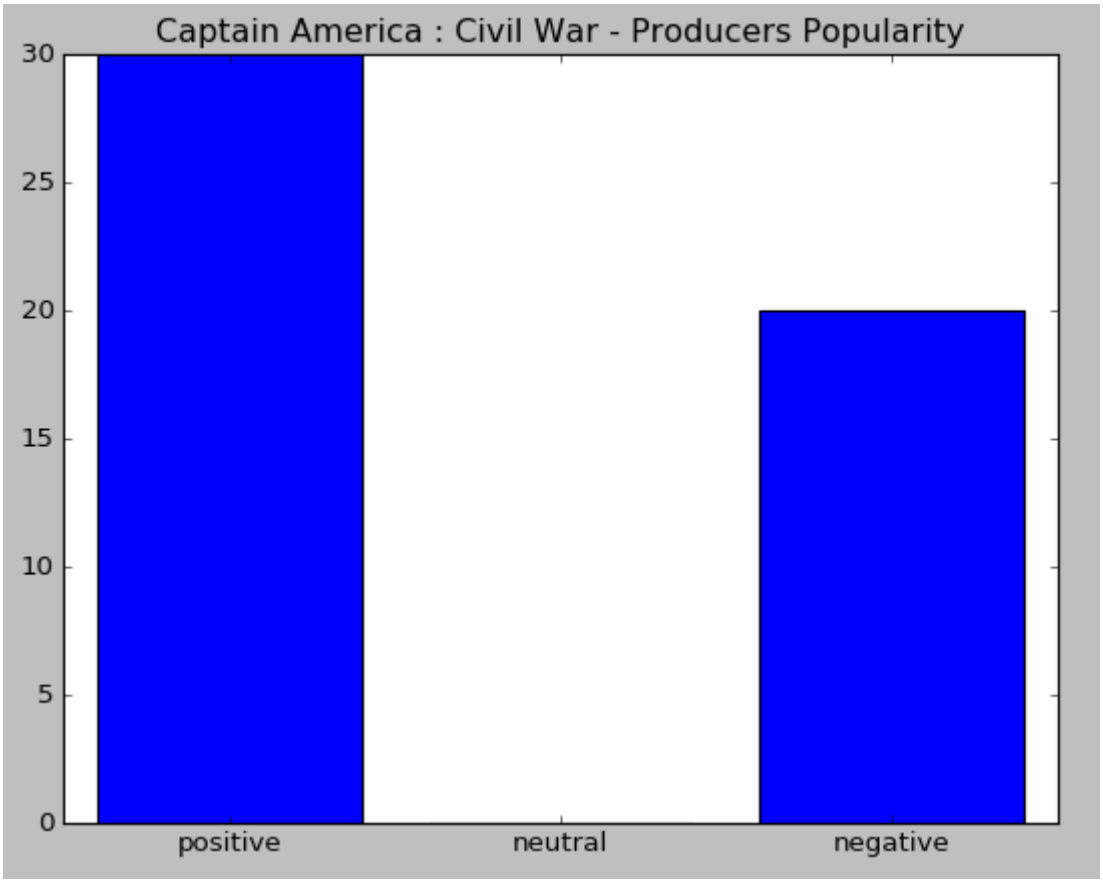
Batman v Superman:

```
dc_producers = ['dc']
Total Posts - 45
Total Comments - 18
Unique words - 220
Total Counts - 401
% positive - 44.4444444444
% negative - 11.1111111111
% neutral - 44.4444444444
```



Captain America:

```
mv_producers = ['marvel']
Total Posts - 65
Total Comments - 5
Unique words - 38
Total Counts - 57
% positive - 60.0
% negative - 40.0
% neutral - 0.0
```



8 Conclusions

- Comparing the overall popularities of the two movies, Batman v Superman: Dawn of Justice is more popular than Captain America: Civil War.
- There is a lot of buzz among the actors in the Batman v Superman: Dawn of Justice as compared to the actors in Captain America: Civil War.
- Director of the movie Batman v Superman: Dawn of Justice, Zack Snyder is more in the news that the director of Captain America: Civil War.
- The characters in Batman v Superman: Dawn of Justice are more commented than the characters in the movie Captain America: Civil War.
- Comics Production house DC is more commented than Marvel.

9 Code

```
import json
import requests
import facebook
from textblob import TextBlob
import re
import matplotlib.pyplot as plt
```

```
ACCESS_TOKEN='CAACEdEose0cBAKZCcGu1gHvxzFmq822MRrryKsl6OYPR9yWev3KYGnpTONVZAAaZACcBv0
hAZAdDSaqrsn6D8s6tKQWeuUMBaIBW219FQAQzHVGXYN1mCCZCDlKAJZBrsz8gPKf4v7XgDDdacVYKmOZA6Cc
4KZA44hds8hdZBsHg1hK10gTWF1PZBZBASEAtNAOiUFbqgqGaFTRsQZDZD'

pepsi_id = 'batmanvsuperman'
#cacw = 'captainamericacw'
i = 0

j = 0
pol_list =[]
g = facebook.GraphAPI(ACCESS_TOKEN)

dc_actors = ['ben', 'affleck', 'henry', 'cavill']
mv_actors = ['robert', 'downey', 'chris', 'evans']

dc_director = ['zack', 'snyder']
mv_director = ['joe', 'russo', 'anthony']

dc_characters = ['superman', 'batman', 'wonder','knight', 'clark', 'kent', 'bruce',
'wayne']
mv_characters = ['ironman', 'captain', 'america', 'tony', 'stark', 'rogers']

dc_producers = ['dc']
mv_producers = ['marvel']


#d = [two
PolDict = {"positive": 0, "negative": 0, "neutral": 0}
comment = 0
wcount=0
new_dict = dict()
for one in g.get_connections(pepsi_id,"posts", limit=2500)['data']:
#for one in g.get_object('/walmart/' + 'posts', since='2013-01-01', until='2014-01-
10', limit=500)['data']:
    j = j + 1
    print("Post # "+str(j))
    for two in one['comments']['data']:
_________nohttp = two['message'].replace('\n', '')
_________nohttp = re.sub(r'http://\b.*',' site ',nohttp)
_________nohttp = re.sub(r'https://\b.*',' site ', nohttp)
_________nohttp = re.sub(r'[^A-Za-z0-9\s\.\!\?\'\-]+', '', nohttp)
_________#if("Marvel" not in nohttp):
```

```
        #    continue
        if(any(substr in nohttp.lower() for substr in dc_director)):
            comment = comment + 1
        else:
            pass
            #continue

        i = i + 1
        print("New Comment #"+str(i)+"-->  "+nohttp[:500])
        print("Old Comment #"+str(i)+"-->  "+two['message'].replace('\n',''))
        #print("\n")
        tweet = TextBlob(nohttp)
        #print(tweet.sentiment.polarity)
        #print(tweet.word_counts['pepsi'])
        for w in tweet.word_counts:
            #print(w+" = "+str(tweet.word_counts[w]))
            if(new_dict.get(w)):
                new_dict[w] = new_dict[w] + tweet.word_counts[w]
            else:
                new_dict[w] = tweet.word_counts[w]



        if(tweet.sentiment.polarity < 0):
            PolDict["negative"] +=1
        elif(tweet.sentiment.polarity == 0):
            PolDict["neutral"] +=1
        else:
            PolDict["positive"] +=1
#         wcount = wcount + int(tweet.word_counts())
    print("############## Total Comments - ",str(i))

pos,neg,neu = 0,0,0
print("Number of words:", wcount)
print(PolDict.items())
#print(new_dict)
#plt.bar(range(len(PolDict)), PolDict.values(), align='center')
#plt.xticks(range(len(PolDict)), PolDict.keys())
title("Batman v Superman : Dawn of Justice - Overall Popularity")


wiki = TextBlob("Python is a high-level, general-purpose programming language.")
#wiki.tags
pd = PolDict
summ = sum(pd.values())
```

```
print("Total Posts - "+str(j))
print("Total Comments - "+str(i))
print("Unique words - "+str(len(new_dict)))
print("Total Counts - "+str(sum((new_dict.values()))))
#print("% positive - "+str(float(PolDict["positive"])/summ*100.0))
#print("% negative - "+str(float(PolDict["negative"])/summ*100.0))
#print("% neutral - "+str(float(PolDict["neutral"])/summ*100.0))

#piedict = {'Jones': 30, 'Jack': 50, 'Jill': 20}

#plt.pie(piedict.values, labels=piedict.keys,autopct='%1.1f%%', shadow=True)
plt.show()
per_rate = []

per_rate.append(float(PolDict["positive"])/summ*100.0)
per_rate.append(float(PolDict["negative"])/summ*100.0)
per_rate.append(float(PolDict["neutral"])/summ*100.0)

labels = 'Positive', 'Negative', 'Neutral'
sizes = per_rate
colors = ['yellowgreen', 'gold', 'lightskyblue']
explode = (0, 0.1, 0)  # only "explode" the 2nd slice (i.e. 'Hogs')
print(PolDict.keys())
plt.pie(sizes, explode=explode, labels=labels, colors=colors, autopct='%1.1f%%',
shadow=True, startangle=90)
# Set aspect ratio to be equal so that pie is drawn as a circle.
plt.axis('equal')

fig = plt.figure()
ax = fig.gca()



plt.show()'
```