Cyber Bullying Detection using NLP Techniques

J COMPONENT PROJECT REPORT

Winter 2019-20

Submitted by

KUDRAT SETIA (17BCE0001)
CHRIS LAZAR (17BCE2160)
SIMRAN KOUL (17BCE2210)
MEYGA ANNE ALEXANDER (17BCE2218)

in partial fulfillment for the award of the degree of

B. Tech

in

Computer Science and Engineering

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Vellore-632014, Tamil Nadu, India

School of Computer Science and Engineering

May, 2020

Contents

Cyber Bullying Detection using NLP Techniques

1. ABSTRACT

Cyber bullying is the harassment or bullying executed through digital devices like computers, laptops, smartphones, and tablets. The platforms where cyber bullying can occur include social media, chat rooms, and gaming platforms where people can view and participate in the sharing of content. While social media offer great communication opportunities, they also increase the vulnerability of young people to threatening situations online. Recent studies report that cyberbullying constitutes a growing problem among youngsters. Successful prevention depends on the adequate detection of potentially harmful messages and the information overload on the Web requires intelligent systems to identify potential risks automatically.

In this project we have built a system in which we can extract the comments from any post on Facebook and detect the cyber bullying by doing sentiment analysis. This system will segregate the comments as positive, negative, abusive, neutral, facts, etc. The heart of this model is that we have mapped the comments of the user with reactions given by same user to make the model more precise.

This can be explained by an example, suppose a person posts a photo and some of his/her friend posts an abusive comments but along with a ‘love’ react on the post then that particular comment will not be considered as abusive or insulting because the friend have posted the comment with good intention. Furthermore, this model is also compatible of translating the given text in any language into English and then doing the Sentiment Analysis. Then we have visualized the results in an interactive manner such that even a naive user can understand the exact statistics.

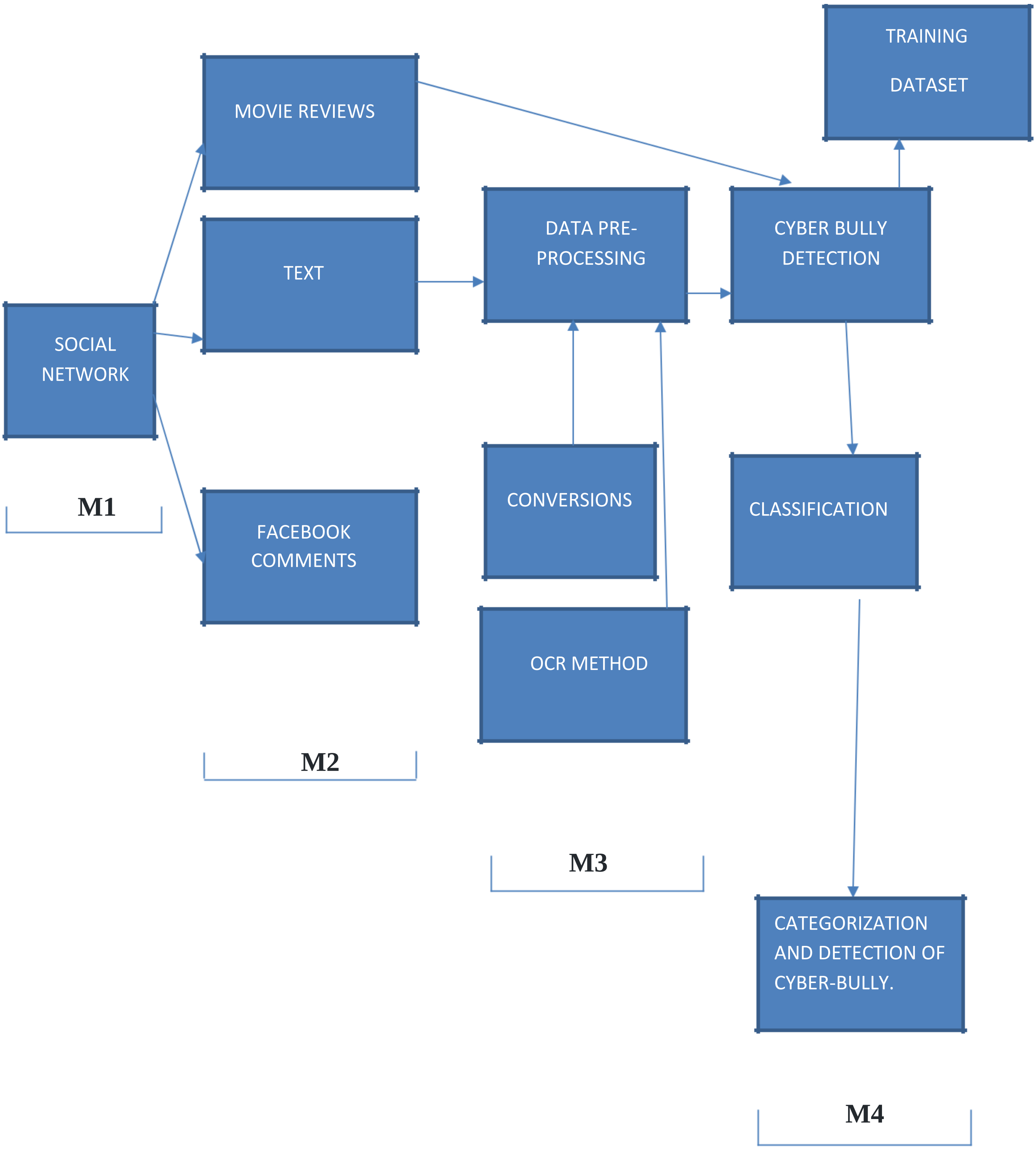
2. INTRODUCTION

[13] With the increasing use of social media like Facebook, cyberbullying behavior has received more and more attention. Bullying is defined as intentional aggression carried out repeatedly by one individual or a group of individuals towards a person who is unable to easily defend him or herself Cyberbullying includes sending, posting, or sharing negative, harmful, false, or mean content about someone else. It can include sharing personal or private information about someone else causing embarrassment or humiliation. Some cyberbullying crosses the line into unlawful or criminal behavior. The abundance of public discussion spaces on the Internet has in many ways changed how we communicate with others. These discussions can often be productive, but the anonymity that comes with hiding behind a username has allowed users to post insulting or inappropriate comments. These posts can often create a hostile or uncomfortable environment for other users, one that may even discourage them from visiting the site. In 2017, about 50% of young social media users reported being bullied online in various forms. Popular social media platforms like Twitter and Facebook are not immune, as racist and sexist attacks may even have caused potential buyers of Twitter to balk.

Cyberbullying may cause many serious and negative impacts on a person’s life and even lead to teen suicide. The detection of cyberbullying is often formulated as a classification problem. Techniques typically used for document classification, topic detection, and sentiment analysis can be used to detect electronic bullying using characteristics of messages, senders, and the recipients. To reduce and stop cyberbullying, one effective solution is to automatically detect bullying content based on appropriate machine learning and natural language processing techniques.

This project focuses on can extract the comments from any post on Facebook and detect the cyber bullying by doing sentiment analysis. Sentiment refers to the use of natural language processing to systematically identify, extract, quantify, and study affective states and subjective information. is the interpretation and classification of emotions (positive, negative and neutral) within text data using text analysis techniques.

3. ARCHITECTURE DIAGRAM



4. BACKGROUND STUDY

[13] Cyberbullying has grown as an important societal challenge nowadays. The Cyberbullying affects both in terms of psychological and emotional means of a person. So, there is a need to devise a method to detect and prevent cyberbullying in social networks. Most of the existing cyberbullying methods involves only text detection and few methods are available for analyzing the visual detection.

Booming of social networking leads to the extensive spread of cybercrimes such as cyberbullying, which is a quite severe problem for teenagers. Traditional studies of cyber-crime such as bullying stand more on a macroscopic view. Conducted by scientists and psychologists, those studies focus on the statistics of cyber-bullying and how to prevent them in a psychological way. As big social network service providers all APIs for academic research, instead of doing statistical study on limited sampled data, researchers are able to access to much larger corpus by using data crawling, which further drives the development of the computational study of cyberbullying based on machine learning and natural language processing techniques.

[2] Sentiment analysis of short texts such as single sentences and Twitter messages is challenging because of the limited contextual information that they normally contain. Effectively solving this task requires strategies that combine the small text content with prior knowledge and use more than just bag-of-words. One introductory work has been presented in one of the papers, in which several NLP models such asBoW, Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA) have been applied to detect the signals associated with. Bullying in social media. Their outcomes have already been verified the possibility of automatic cyberbullying detection. Dinakar et.al used Lin-ear Discriminative Analysis to learn label specific features and combine them with BoW features to train a classifier. The length of label-specific features is limited to be less than the class numbers, which creates problems for the performance boost. Nahar et.al magnified the weights corresponding to bullying words by a double. This work shares a similar motivation with the construction of bullying features in our model that bullying features should be enhanced. However, they did not consider the words’ semantics and the scaling operation was quite arbitrary. In addition, Nahar et.al [15] also adopted topic models including Probabilistic Latent Semantic Analysis (PLSA) and Latent Dirichlet Allocation (LDA) to learn

topics and performed feature selection, which is conducted over topics that feature that comes under bullying-like topics are preserved. However, the determination of bullying-like topics lacks a general theory basis.

5. METHODOLOGY

- We have created a corpus of commonly used negative phrases, words and facts, by using Selenium library to scrape the web and extract them. For training of our model, we have used the movie reviews dataset which is available in the NLTK corpus. This dataset has actual review text annotated along with their associated sentiments. Now, we have to generate the testing data. This can be done by using SocioFi tool which is a third party application which is used to extract real-time Facebook comments in reaction to posts on the social media platform. This data is not labeled and our model will predict the sentiment label based on the text in the comments.

- First, we generate the corpus of the negative and abusive phrases or words and factual negative phrases using import and use Selenium driver. The driver will scrape the web and create a corpus of these words and phrases which we will further use as the abusive word corpus.

- We then import the NLTK corpus from the TextBlob library, which includes the movie reviews dataset with labeled reactions that will be used as our training dataset. After that, we will create a Naïve Bayes classifier, which will train on the reviews text available in the movie reviews using the abusive word corpus. This will further generate a polarity and subjectivity score for each of these texts and a final label as abusive, negative, factual negative, neutral, positive, etc. is predicted. Now, based on the actual label and the predicted label, the model learns.

- We will similarly create a Support Vector Machine classifier that trains on the same movie reviews dataset using cyber bullying corpus. For test data generation, we will use Sociofy that is a third party application to scrape Facebook and retrieve comments on posts. This will extract all comments and reactions from the Facebook along with the user id that are posted. The models are independently tested on these extracted Facebook comments. The models analysis the sentiment and subjectivity of each and every comment and set certain parameters according to the user beyond which the negative sentiment must not be tolerated.

- When the comments are predicted negative by the system, the abusive words in that comment are found and mapped to the abusive word corpus. These comments are then categorized into labels like abusive, negative, factual negative, neutral and positive based on the model’s prediction only after they have generated the polarity and subjectivity scores for each of these comments.

- After the whole process is completed, the user ids are stored which are mapped with the reaction user ids. If any user has posted any abusive content on the post but the reaction posted seems to be positive, then it will not be booked for cyber bullying because it maybe some sarcastic comment according to the admin of the post.

- In the end, we plot a bar graph for the percentage of comments in each category predicted by both the models separately.

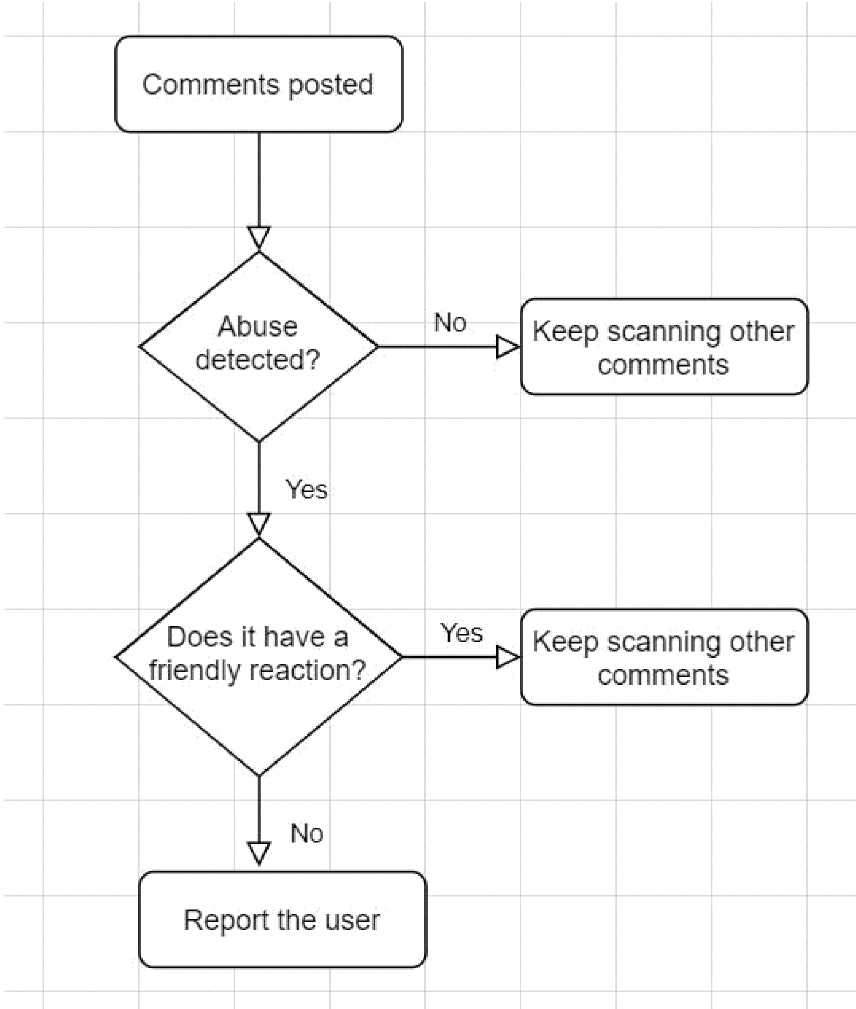
6. PROPOSED MODEL

In this project we have intended to build a system in which we can extract the comments from any post on Facebook and detect the cyber bullying by doing sentiment analysis. The system will then segregate the comments as positive, negative, abusive, neutral, facts, etc. the main function of this model is that we have mapped the comments of the user with the reactions given by the same user to make the model more accurate.

For example, a boy posts a photo and some of his friend posts abusive comments but along with a ‘friendly’ react on the post then that particular comment will not be considered as abusive or insulting because the friend has posted this comment with a good intention.

Additionally, this model is also compatible of translating the given text in any language into English and further doing the Sentiment Analysis. By this procedure, we have visualized the results in an interactive manner such that even a naïve user will be able to understand the exact statistics.

For example, if you consider the below Facebook post with comments:



Now, this post is receiving abusive comments that will be detected and first checked if they are having a friendly reaction to it or not. As we can see, the comments don't have a friendly tone to it and are harsh. Thus, the system will report the user and display a prompt message upon the comment.

7. RESULTS AND DISCUSSION

➢ Using pattern analyzer

Importing the libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib notebook
from textblob import TextBlob as tb
```

Importing the Comments dataset

In [2]:
```
df=pd.read_csv("Desktop/comments_609026426288720 (2).csv")
df
```

Out[2]:

	Comment Id	Date	Time	form_id	From	Message	Likes	Comments
0	609026476288715_609026876288675	7/23/2019	10:22	6.090000e+14	Bhai bhai	hi	0	0
1	609026476288715_609026892955340	7/23/2019	10:22	6.090000e+14	Bhai bhai	bye	0	1
2	609026476288715_609029069621789	7/23/2019	10:29	2.460000e+15	Harsh Khandelwal	wow very nice post..	0	2
3	609026476288715_609029662955063	7/23/2019	10:30	2.460000e+15	Harsh Khandelwal	very funny	0	0
4	609026476288715_609029732955056	7/23/2019	10:31	6.090000e+14	Bhai bhai	thankyou so much dear	0	0
5	609026476288715_609253272932702	7/23/2019	17:37	6.090000e+14	Bhai bhai	Shut up????	0	0
6	609026476288715_609264796264883	7/23/2019	18:03	6.090000e+14	Bhai bhai	you dumb ass	0	0
7	609026476288715_609264796264889	7/23/2019	18:03	4.090000e+14	Vidit	Very Bad post	0	0
8	609026476288715_609264796264890	7/23/2019	18:04	4.090000e+14	Vidit	वाह तुम कमाल की लग रही हो	0	0
9	609026476288715_609029662955063	7/23/2019	10:30:00	2.460000e+15	Harsh Khandelwal	سبی جدا	0	0

Importing the Reactions dataset

In [3]:
```
df1=pd.read_csv("Desktop/reactions_609026426288720(1).csv")
df1
```

Out[3]:

	form_id	From	Type
0	6.090000e+14	Bhai bhai	LIKE
1	2.460000e+15	Harsh Khandelwal	HAHA
2	4.090000e+14	Vidit	Angry

Importing the Corpus(Vulgar words)

In [4]:
```
data=pd.read_csv("Desktop/data.csv")
```

In [5]:
```
df["form_id"]  #Column used to merge the comments and the reactions
```

Out[5]:
```
0    6.090000e+14
1    6.090000e+14
2    2.460000e+15
3    2.460000e+15
4    6.090000e+14
5    6.090000e+14
6    6.090000e+14
7    4.090000e+14
8    4.090000e+14
9    2.460000e+15
Name: form_id, dtype: f
```

Merging the Dataset

In [6]:
```
merged = df.merge(df1, on='form_id')
del merged["From_y"]
merged.to_csv("output.csv", index=False)
merged
```

Out[6]:

	Comment Id	Date	Time	form_id	From_x	Message	Likes	Comments	Type
0	609026476288715_609026876288675	7/23/2019	10:22	6.090000e+14	Bhai bhai	hi	0	0	LIKE
1	609026476288715_609026892955340	7/23/2019	10:22	6.090000e+14	Bhai bhai	bye	0	1	LIKE
2	609026476288715_609029732955056	7/23/2019	10:31	6.090000e+14	Bhai bhai	thankyou so much dear	0	0	LIKE
3	609026476288715_609253272932702	7/23/2019	17:37	6.090000e+14	Bhai bhai	Shut up????	0	0	LIKE
4	609026476288715_609264796264883	7/23/2019	18:03	6.090000e+14	Bhai bhai	you dumb ass	0	0	LIKE
5	609026476288715_609029069621789	7/23/2019	10:29	2.460000e+15	Harsh Khandelwal	wow very nice post..	0	2	HAHA
6	609026476288715_609029662955063	7/23/2019	10:30	2.460000e+15	Harsh Khandelwal	very funny	0	0	HAHA
7	609026476288715_609029662955063	7/23/2019	10:30:00	2.460000e+15	Harsh Khandelwal	سبی جدا	0	0	HAHA
8	609026476288715_609264796264889	7/23/2019	18:03	4.090000e+14	Vidit	Very Bad post	0	0	Angry
9	609026476288715_609264796264890	7/23/2019	18:04	4.090000e+14	Vidit	वाह तुम कमाल की लग रही हो	0	0	Angry

Defining lists for differnet categories of the comments

In [7]:
```
positive=[]
negative=[]
neutral=[]
facts=[]
abusive=[]
others=[]
```

Training the Model

```
In [8]: from __future__ import absolute_import
from collections import namedtuple
import nltk
from textblob.en import sentiment as pattern_sentiment
from textblob.tokenizers import word_tokenize
from textblob.decorators import requires_nltk_corpus
from textblob.base import BaseSentimentAnalyzer, DISCRETE, CONTINUOUS


class PatternAnalyzer(BaseSentimentAnalyzer):

    kind = CONTINUOUS
    RETURN_TYPE = namedtuple('Sentiment', ['polarity', 'subjectivity'])

    def analyze(self, text, keep_assessments=False):

        if keep_assessments:
            Sentiment = namedtuple('Sentiment', ['polarity', 'subjectivity', 'assessments'])
            assessments = pattern_sentiment(text).assessments
            polarity, subjectivity = pattern_sentiment(text)
            return Sentiment(polarity, subjectivity, assessments)

        else:
            Sentiment = namedtuple('Sentiment', ['polarity', 'subjectivity'])
            return Sentiment(*pattern_sentiment(text))



def _default_feature_extractor(words):
    return dict(((word, True) for word in words))
```

Categorizing the comment

```
In [9]: comm=merged["Message"] +" "+ merged["Type"]  # Mapping comment with the reaction

reac=merged["Type"]
comm_id=merged["Comment Id"]


# Categorizing the comments
for i in range(len(merged)):
    b=comm[i]
    flag=int(0)
    blob = tb(b)
    p=blob.detect_language()
    print(b)
    print(p)
    if(p!='en'):
        b=blob.translate(from_lang=p, to ='en')
        if(b.sentiment.polarity>0.30 and b.sentiment.subjectivity!=1):
            positive.append(comm_id[i])

        elif(b.sentiment.polarity==0 and b.sentiment.subjectivity!=1):
            neutral.append(comm_id[i])

        elif(b.sentiment.polarity<-0.25 and b.sentiment.subjectivity!=1):
            for row in data:
                if(row.lower() in b):
                    abusive.append(b)
                    flag=1;
                    break;
            if(flag==0):
                negative.append(b)

        elif(b.sentiment.subjectivity>=0.75):
            facts.append(comm_id[i])
        else:
```

```
        else:
            others.append(comm_id[i])

        print(b.sentiment)


    else:
        if(tb(b).sentiment.polarity>0.30 and tb(b).sentiment.subjectivity!=1):
            positive.append(comm_id[i])

        elif(tb(b).sentiment.polarity==0 and tb(b).sentiment.subjectivity!=1):
            neutral.append(comm_id[i])

        elif(tb(b).sentiment.polarity<-0.25 and tb(b).sentiment.subjectivity!=1):
            for row in data:
                if(row.lower() in b):
                    abusive.append(b)
                    flag=1;
                    break;
            if(flag==0):
                negative.append(b)

        elif(tb(b).sentiment.subjectivity>=0.75):
            facts.append(comm_id[i])
        else:
            others.append(comm_id[i])

        print(tb(b).sentiment)
```

```
hi LIKE
en
Sentiment(polarity=0.0, subjectivity=0.0)
bye LIKE
en
Sentiment(polarity=0.0, subjectivity=0.0)
thankyou so much dear LIKE
en
```

```
hi LIKE
en
Sentiment(polarity=0.0, subjectivity=0.0)
bye LIKE
en
Sentiment(polarity=0.0, subjectivity=0.0)
thankyou so much dear LIKE
en
Sentiment(polarity=0.2, subjectivity=0.2)
Shut up???? LIKE
en
Sentiment(polarity=0.0, subjectivity=0.0)
you dumb ass LIKE
en
Sentiment(polarity=-0.375, subjectivity=0.5)
wow very nice post.. HAHA
en
Sentiment(polarity=0.36000000000000004, subjectivity=0.7666666666666666)
very funny HAHA
en
Sentiment(polarity=0.2625, subjectivity=0.65)
سيئ جدا HAHA
ar
Sentiment(polarity=-0.24999999999999992, subjectivity=0.48333333333333333)
Very Bad post Angry
en
Sentiment(polarity=-0.7049999999999998, subjectivity=0.9333333333333333)
वाह तुम कमाल की लग रही हो Angry
hi
Sentiment(polarity=0.35000000000000003, subjectivity=0.95)
```

In [10]:
```
neutral
```
Out[10]:
```
['609026476288715_609026876288675',
 '609026476288715_609026892955340',
 '609026476288715_609253272932702']
```

In [11]:
```
facts
```
Out[11]:
```
[]
```

In [12]:
```
positive
```
Out[12]:
```
['609026476288715_609029069621789', '609026476288715_609264796264890']
```

In [13]:
```
tb("you dumb ass LIKE").sentiment
```
Out[13]:
```
Sentiment(polarity=-0.375, subjectivity=0.5)
```

In [14]:
```
abusive
```
Out[14]:
```
['you dumb ass LIKE']
```

In [15]:
```
neg=np.array(negative)
np.unique(neg)
```
Out[15]:
```
array(['Very Bad post Angry'], dtype='<U19')
```

Calculating the percentage of each category

In [16]:
```
p_positive = (len(positive)/len(merged))*100
p_negative = (np.unique(neg).size/len(merged))*100
p_neutral  = (len(neutral)/len(merged))*100
p_facts    = (len(facts)/len(merged))*100
```

Calculating the percentage of each category

In [16]:
```
p_positive = (len(positive)/len(merged))*100
p_negative = (np.unique(neg).size/len(merged))*100
p_neutral  = (len(neutral)/len(merged))*100
p_facts    = (len(facts)/len(merged))*100
p_abusive  = (len(abusive)/len(merged))*100
p_others  = (len(others)/len(merged))*100
```

In [17]:
```
#Printing the percentage

print(str(p_positive) + "% people's comment was positive")
print(str(p_negative) + "% people's comment was negative")
print(str(p_neutral) + "% people's comment was neutral")
print(str(p_facts) + "% people's comment was facts")
print(str(p_abusive) + "% people's comment was abusive")
print(str(p_others) + "% people's comment was others(funny,sad...)")
```

```
20.0% people's comment was positive
10.0% people's comment was negative
30.0% people's comment was neutral
0.0% people's comment was facts
10.0% people's comment was abusive
30.0% people's comment was others(funny,sad...)
```

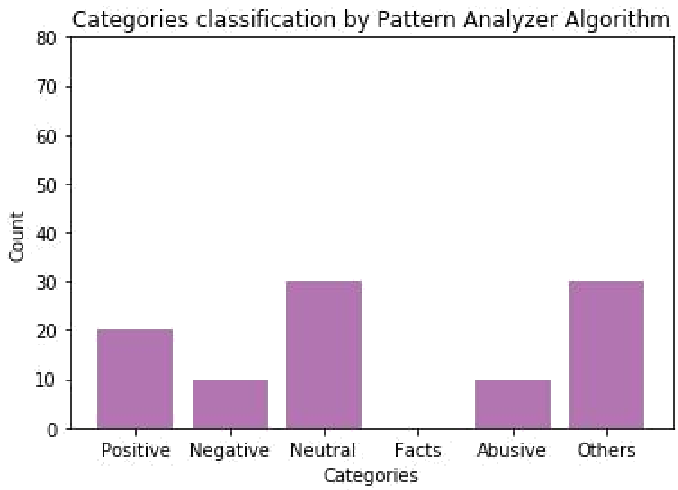
Plotting the Bar graph

In [18]:
```
height = [p_positive, p_negative, p_neutral, p_facts,p_abusive,p_others]
bars = ('Positive', 'Negative', 'Neutral', 'Facts', 'Abusive', 'Others')
y_pos = np.arange(len(bars))

# Create bars and choose color
plt.bar(y_pos, height, color = (0.5,0.1,0.5,0.6))
```

Plotting the Bar graph

In [18]:
```
height = [p_positive, p_negative, p_neutral, p_facts,p_abusive,p_others]
bars = ('Positive', 'Negative', 'Neutral', 'Facts', 'Abusive', 'Others')
y_pos = np.arange(len(bars))

# Create bars and choose color
plt.bar(y_pos, height, color = (0.5,0.1,0.5,0.6))

# Add title and axis names
plt.title('Categories classification by Pattern Analyzer Algorithm')
plt.xlabel('Categories')
plt.ylabel('Count')

# Limits for the Y axis
plt.ylim(0,80)

# Create names
plt.xticks(y_pos, bars)

# Show graphic
plt.show()
```

➢ Result using pattern analyzer



➢ Analysis between Naïve Bayes and Pattern Analyzer

Importing the Libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib notebook
from textblob import TextBlob as tb
```

Categories obtained from Pattern Analyzer

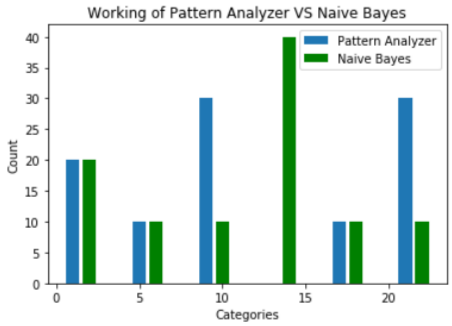
In [2]:
```
p_positive1 = 20.0
p_negative1 = 10.0
p_neutral1  = 30.0
p_facts1    = 0.0
p_abusive1  = 10.0
p_others1  =  30.0
```

Categories obtained from Naive Bayes

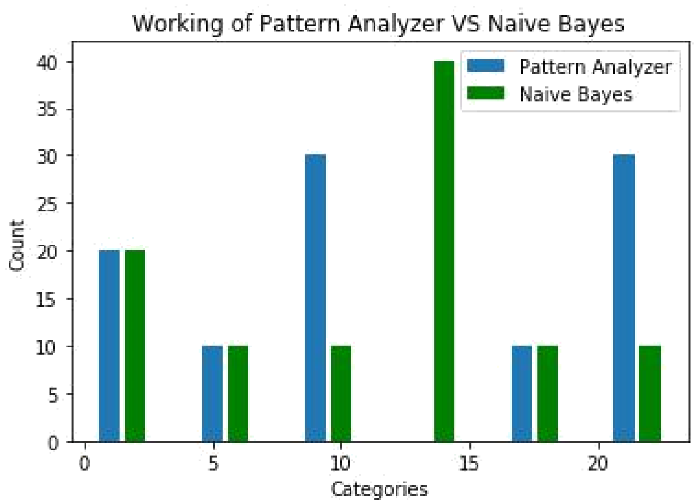
In [3]:
```
p_positive2 = 20.0
p_negative2 = 10.0
p_neutral2  = 10.0
p_facts2    = 40.0
p_abusive2  = 10.0
p_others2  =  10.0
```

Visualizing the categories

```
In [4]: plt.bar([1,5,9,13,17,21],[p_positive1, p_negative1, p_neutral1, p_facts1,p_abusive1,p_others1], label="Pattern Analyzer")
        plt.bar([2,6,10,14,18,22],[p_positive2, p_negative2, p_neutral2, p_facts2,p_abusive2,p_others2], label="Naive Bayes", color='g')
        plt.legend()
        plt.xlabel('Categories')
        plt.ylabel('Count')
        plt.title('Working of Pattern Analyzer VS Naive Bayes')
        plt.show()
```



➢ Comparison of Pattern Analyzer and Naïve Bayes



8. CONCLUSION

We have used Sentiment analysis for detecting cyber bullying. We created a Naïve Bayes classifier, that trains on the review text available in the movie reviews, using the abusive word corpus, which generates a polarity and subjectivity score, for each of these texts, and a final label as abusive, negative, factual negative, neutral, positive. This system will then segregate the comments as positive, negative, abusive, neutral, facts thus helping us detect comments that come under cyberbullying.

9. REFERENCES

[1] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang. Abusive Language Detection in Online User Content. In WWW, 2016.,6

[2] D. Santos, C. Nogueira, and M. Gatti. Deep Convolutional Neural Network for Sentiment Analysis of Short Texts. COLING, 2014.,13

[3] Divyashree, H. Vinutha, N. S. Deepashree. An Effective Approach for Cyberbullying Detection and Avoidance. International Journal of Innovative Research in Computer and Communication Engineering, 2016.,14

[5] G. E. Hine, J. Onaolapo, E. De Cristofaro, N. Kourtellis, I. Leontiadis, R. Samaras, G. Stringhini, and J. Blackburn. A Measurement Study of 4Chan’s Politically Incorrect Forum and its effort on the web. In ICWSM, 2017.,3

[6] H. Hosseinmardi, S. A. Mattson, R. I. Rafiq, R. Han, Q. Lv, and S. Mishra. Analyzing Labelled Cyberbullying Incidents on the Instagram Social Network. InSocInfo, 2015,1

[7] Huang, Minlie, Y. Cao, and C. Dong. Modelling Rich Contexts for Sentiment Classification with LSTM. arXiv preprint arXiv: 1605.01478, 2016.,12

[8] I. Kayes, N. Kourtellis, D. Quercia, A. Iamnitchi, and F. Bonchi. The Social World of Content Abuser in Community Question Answering. In WWW, 2015.,5

[9] J. M. Xu, X. Zhu, A. Bellmore. Learning from Bullying Traces in Social Media. University of Wisconsin-Madison, 2016.,22 [10] J. M. Xu, X. Zhu, and A. Bellmore. Fast Learning for Sentiment Analysis on Bullying. In WISDOM, 2012.,10

[11] K. Dinakar, R. Reichart, and H. Lieberman. Modelling the Detection of Textual Cyberbullying. The Social Mobile Web, 11, 2011.,7

[12] L. Engman. Automatic Detection of Cyberbullying on Social Media. UMEA UNIVERSITY, 2016.,15

[13] Pradheep. T, Yogeshwaran.T, J.I.Sheeba, S. Pradeep Devaneyan. AUTOMATIC MULTIMODEL CYBERBULLYING DETECTION FROM SOCIAL NETWORKS, 2016.,17

[14] V. Nahar, S. Tankard, X. Li, and C. Pang. Sentiment Analysis for Effective Detection of Cyberbullying. In APWeb, 2012.,9