**Report on**

**Detection and Analysis of Cyber-bullying tweets based on Events**

**Vaishali Byloor**

Matrikelnummer: **78613**

**Maria Mamontova**

Matrikelnummer: **868678**

**July 2017**

Contents:

1. Motivation

2. Introduction

2.1 Sentiment Analysis

2.2 Twitter

3. Methodology

3.1 Classifiers

3.2 Datasets

3.2.1 Twitter Sentiment Corpus

3.2.2 Movie Review Dataset

3.3 Pre Processing

3.3.1 Hashtags

3.3.2 Handles

3.3.3 URLs

3.3.4 Emoticons

3.3.5 Punctuations

3.3.6 Repeating Characters

3.3.7 Removing Stopwords

3.4 Stemming Algorithms

3.4.1 Snowball Stemmer

3.4.2 Lemmatization

3.5 Features

3.5.1 N-grams

3.5.2 Unigrams

3.5.3 Bigrams

3.5.4 Trigrams

4. Architecture

4.1 Use case Diagram

4.2 Workflow

4.2.1 Hierarchical classifiers

5. Experimentation

5.1 Classifiers

5.1.1 Naive Bayes

5.1.3 NuSVC

5.1.4 Logistic Regression

5.2 Comparison of Accuracy of the Classifiers

5.3 Confusion Matrices of the Classifiers

5.4 Comparison of Precision and Recall of the classifiers

5.5 Comparison of F1 measures

5.6 10-fold cross validations

6. Future Work

7. Conclusion

**1. MOTIVATION**

The usage of social networks amongst children, adolescents and families is nowadays a common practise in our everyday lives. Social networking sites such as Twitter,Facebook,YouTube, blogs etc provide various ways for people to communicate with one another . With the rapid growth of social media, there is increasing number of users especially adolescents spending a significant amount of time on various social networking sites to connect with others, to share information, and to pursue common interests. While many beneficial effects definitely exist, the platform also has risk of exposing various offensive online contents against people which is termed as bullying.

Wikipedia defines cyberbullying as the use of information technology to repeatedly harm or harass other people in a deliberate manner. According to U.S. Legal Definitions, Cyberbullying could be limited to posting rumors or gossips about a person on the internet bringing about hatred in others minds; or it may go to the extent of personally identifying victims and publishing materials severely defaming and humiliating them.

A main reason individuals are targeted with bullying is perceived differences, i.e., any characteristic that makes an individual stand out differently from his or her peers. These include race, socioeconomic status, gender, sexuality, physical appearance, and behaviors.

Cyberbullying reflects a venue (other than face to face contact) through which verbal and relational forms can occur.This could have the negative impact on people, especially adolescents, which might lead to depression, anxiety, loneliness,aggression and also suicide. Recently, there are increasing cases of cyberbullying due to lack of preventing measures. Therefore, cyberbullying must be considered as a serious problem, which needs to be prevented from happening.

Existing detection techniques occur from the fields of natural language processing, computational linguistics, text mining, and a range from machine learning methods to rule-based methods. Machine learning methods involve training of models on specific collections of documents.

**2. INTRODUCTION:**

This work presents the discussion of the challenges, which arises during sentiment analysis and classification. In addition, this thesis includes the review of experiments and researches which were done to solve the similar problems. Going through the related work of what methods have been used and which of them are successful, we have discovered one main tendency: the performance of the specific method mainly depends on the dataset. It depends on the complexity of the data, if there is any positive, negative, neutral data in the datasets [6].

**2.1 SENTIMENT ANALYSIS**:

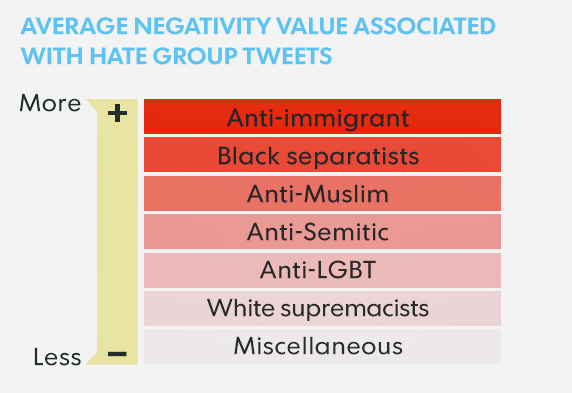
Sentiment analysis in reference to social media simple means the emotion behind a mention online. It is a way to measure the polarity of the conversation - is the person happy, sad or just indifferent. This is more important than the measurement of mentions alone since this could be misleading. If you were measuring mentions for a company’s new product, one might assume a strong increase in mentions meant it was being well received. After all, more mentions means more people talking about the product.

**2.2 TWITTER:**

Twitter is an information network made up of 140-character messages called Tweets. It's an easy way to discover the latest news related to subjects you care about. The maximum length of a Twitter message is 140 characters. This means that we can practically consider a tweet to be a single sentence, void of complex grammatical constructs. Because of this and the 140- character limit, language used in Tweets tend be shorthand, and filled with slang and misspellings. Data availability makes twitter a preferred social platform. With the Twitter API, it is easy to collect millions of tweets for training.

**2.3 EVENTS**

The two current world events selected are the refugee crisis in Europe and the Trump wall( hatred against islam and immigrants ). A [study released](https://www.safehome.org/resources/hate-on-social-media/) in February 2017 by by SafeHome.org, an organization of home security experts shows that social media engagement by groups labeled as hate organizations has been booming in the last two to three years and the largest shares of activity are focused on anti-immigrant and anti-Muslim sentiment.



Hashtags we used ,to collect data from Twitter

1 event -

2nd event -

**3. METHODOLOGY:**

**Figure:** illustrates the methodology of the project

We use different feature sets and machine learning classifiers to determine the best combination for detection of cyber bullying content of twitter. We also experiment with various pre-processing steps like - removing of punctuations, emoticons, twitter specific terms and stemming.. We finally train our classifier using various machine-learning algorithms - Naive Bayes, Decision Trees and Maximum Entropy.

**3.1 Collect Data**

1. The first step is the registration of our app.In order to have access to Twitter data programmatically, we created an app that interacts with the Twitter API. In particular, we log-in to Twitter and register a new application →choose a name and a description for the app → we will receive a consumer key and a consumer secret and from the configuration page of your app, you can also get an access token and an access token secret. These four credentials must be kept private: they provide the application access to Twitter on behalf of our account.
2. Install Tweepy→ pip install tweepy==3.3.0
3. [TweetStream.py](https://gist.github.com/bonzanini/af0463b927433c73784d) - use the following python code to stream live tweets. We let this run for a period of time to have collected sufficient number of tweets. The tweets are stored in a json file.

|  |
| --- |
| import tweepy  from tweepy import OAuthHandler  from tweepy import Stream  from tweepy.streaming import StreamListener  consumer\_key = 'YOUR-CONSUMER-KEY'  consumer\_secret = 'YOUR-CONSUMER-SECRET'  access\_token = 'YOUR-ACCESS-TOKEN'  access\_secret = 'YOUR-ACCESS-SECRET'  auth = OAuthHandler(consumer\_key, consumer\_secret)  auth.set\_access\_token(access\_token, access\_secret)  api = tweepy.API(auth)  class MyListener(StreamListener):    def on\_data(self, data):  try:  with open('Corpus.json', 'a') as f:  f.write(data)  return True  except BaseException as e:  print("Error on\_data: %s" % str(e))  return True    def on\_error(self, status):  print(status)  return True    twitter\_stream = Stream(auth, MyListener())  twitter\_stream.filter(track=['#']) |

In machine learning, classification is an instance of supervised learning, i.e. learning where a training set of correctly identified observations is available. Other classifiers work by comparing observations to previous observations by means of a similarity or distance function. In our project, we used multiple classifiers to handle the datasets. Upon which we perform Training and Testing processes. We used few best classifiers namely Naive Bayes, Maximum Entropy, Logistic regression and NuSVC classifier.

**Jupyter Notebook Structure:**

|  |  |
| --- | --- |
| **Section Name** |  |
|  |  |
|  |  |
|  |  |
|  |  |

**DATA PRE-PROCESSING:**

**3.2.1 Twitter Corpus:**

The tweets were gathered in Json format, each line of the file is a json object of the following format:

The program is reading file line by line and casts each non empty line to the Json format. The json objects then are stored in the array.

After retrieving the jsons, the program starts preprocessing the data:

The

This is a collection of 5399 tweets collected for four different topics, namely, Apple, Google, Microsoft, Twitter It is collected and hand-classified by Sanders Analytics LLC. Each entry in the corpus contains Tweet id, Topic and a Sentiment label. We use Twitter-Python library to enrich this data by downloading data like Tweet text, Creation Date, Creator etc. for every Tweet id. Each Tweet is hand classified by an American male into the following four categories. For the purpose of our experiments, we consider Irrelevant and Neutral to be the same class. Positive For showing positive sentiment towards the topic Positive, for showing no or mixed or weak sentiments towards the topic Negative for showing negative sentiment towards the topic and Irrelevant for non-English text or off-topic comments.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Topic | Sentiment | TweetId | Creation Date | Tweet text |
| apple | positive | 1.264156146161  54E+017 | Tue Oct 18  21:53:25 +0000  2011 | un sind alle  [@Apple](about:blank) zu tun hat, ist Swype auf dem iPhone zu bekommen, und es wird knacken sein. Iphone, das ist |
| apple | neutral | 1.252468988304  59E+017 | Sat Oct 15  16:29:22 +0000  2011 | [@notleifgarrett](about:blank)  [@apple](about:blank) Sie, dass Villa einen Tag leisten können. |
| twitter | negative | 1.268645755083 | Thu Oct 20 | Ich brauche |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 82E+017 | 03:37:26 +0000  2011 | #twitter aussteigen |
| twitter | negative | 1.268645101946  84E+017 | Thu Oct 20  03:37:10 +0000  2011 | RT [@Prettynesz:](about:blank)  #twitter immer noch nicht meine fuckin #retweets showin. |

Table 1.0 showing parts of dataset in German

**3.2.2 Movie Review Dataset:**

This dataset consists of different movie reviews, which we used to find the subjectivity of the input tweets retrieved in real time. The dataset has 5000 subjective and 5000 objective processed sentences. Introduced in Pang/Lee ACL 2004. Released June 2004

**3.3 PREPROCESSING:**

Data preprocessing describes any type of processing performed on raw data to prepare it for another processing procedure. It is a data mining technique, which involves transforming raw data into an understandable format.

**Requirement of Data Preprocessing:**

Real-world data are often

● Incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data;

● Noisy: containing errors or outliers;

● Inconsistent: containing discrepancies in codes or names.

Data preprocessing is a proven method of resolving such issues. Moreover, user-generated content on the web is rarely present in a form usable for learning. It becomes important to normalize the text by applying a series of pre-processing steps. We have applied an extensive set of pre-processing steps to decrease the size of the feature set to make it suitable for learning algorithms.

The following figure illustrates various features seen in Micro-blogging; likely, Hashtags, Handles, Emoticons and Punctuations.

**Figure:Tweet** Shows a tweet with various features. [10]

In this section, we mention how we processed the tweets using regular expressions

**3.3.1 Hashtags:**

Hashtag is a word or an un-spaced phrase prefixed with the hash symbol ‘#’. These are used to

both naming subjects and phrases that are currently in trending topics. For example,

#USdebate, #Killerclowns Regular Expression: #(\w+) Replace Expression: HASH\_

**3.3.2 Handles:**

Every Twitter user has a unique username. Anything directed towards that user can be indicated be writing their username preceded by ‘@’. Thus, these are like proper nouns. For example, [@Intel](about:blank) [@Microsoft](about:blank)

Regular Expression: @(\w+) Replace Expression:\_\_HNDL\_

**3.3.3 URLs:**

Users often share hyperlinks in their tweets. Twitter shortens them using its in-house URL shortening service, like [http://t.co/](https://t.co/)wyXCv - Shortened links allow you to share long URLs in a Tweet while maintaining the maximum number of characters for your message. From the point of view of text classification, a particular URL is not important. However, presence of a URL can be an important feature. Regular expression for detecting a URL is complex because of different types of URLs that can be there, but because of Twitter’s shortening service, we can use a relatively simple regular expression.

Regular Expression: (http|https|ftp)://[a-zA-Z0-9\\./] Replace Expression: URL\_

**3.3.4 Emoticons:**

Use of emoticons is very common throughout the web, particularly on micro- blogging sites. We identify the some of the following emoticons and replace them with a single word. Table:emotic lists some of the emoticons we are currently detecting and replacing.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Emoticons** | **Examples** | | | | |
| EMOT\_SMILEY | 🙂 | � | � | � | � |
| EMOT\_LAUGH | � | � | � | � |  |
| EMOT\_SAD | 😔 | 😕 | 🙁 | ☹ | 😟 |
| EMOT\_CRY | 😖 | 😫 | � | 😭 | � |
| EMOT\_ANGRY | � | � | � |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| EMOT\_SHOCK | 😳 | 😱 | � | 😦 | 😵 |
| EMOT\_WINK | 😜 | (; | � | ;-D |  |
| EMOT\_KISS | 😘 | 💏 | 😗 | 😙 | 😚 |
| EMOT\_LOVE | 😍 | 💖 | 💘 | ❤ |  |

**Table:Emoticon** shows some of the emoticons we are detecting and replacing

**3.3.5 Punctuations:**

Although not all Punctuations are important from the point of view of classification but some of these, like question mark, exclamation mark can also provide information about the sentiments of the text. We replace every word boundary by a list of relevant punctuations present at that

point. Table: Punctuation shows the punctuations, which are currently identified.

|  |  |  |
| --- | --- | --- |
| Punctuatuions | Examples | |
| PUNC\_EXCL | ! | i |
| PUNC\_QUES | ? | ¿ |
| PUNC\_ELLP | **...** | … |

**Table:Punctuation** lists the punctuations currently identified.

**3.3.6 Repeating Characters:**

People often use repeating characters while using colloquial language, like "I’m happyyyyyy", "We won, omggggg!" As our final pre-processing step, we replace characters repeating more than twice as two characters.

Regular Expression: (.)\1{1,} Replace Expression: \1\1

**3.4 Stemming Algorithms:**

“Stemming is the process for reducing inflected words to their word stem (base form).”

For example: banks, banking become bank, and investing, invested become invest.

The classifier does not understand that the verbs investing and invested are the same, and treats them as different words with different frequencies. By stemming them, it groups the frequencies of different inflection to just one term — in this case, invest.

All stemming algorithms are of the following major types – affix removing, statistical and mixed. The first kind, Affix removal stemmer, is the most basic one. These apply a set of transformation rules to each word in an attempt to cut off commonly known prefixes and / or suffixes. A trivial stemming algorithm would be to truncate words at N-th symbol. But this obviously is not well

suited for practical purposes. We used snowball-stemmer, which we describe in the next section.

**3.4.1 Snowball Stemmer:**

Snowball algorithm handles two main issues: One is the lack of readily available stemming algorithms for languages other than English. The other is the consciousness of a certain failure on my part in promoting exact implementations of the stemming algorithm described in (Porter

1980), which has come to be called the Porter stemming algorithm. The first point needs some qualification: a great deal of work has been done on stemmers in a wide range of natural languages, both in their development and evaluation. As we deal with multiple languages

(English, German and French), we made use of Snowball stemmer.

|  |  |
| --- | --- |
| 1. | Gets rid of plurals and -ed or -ing suffixes. |
| 2. | Turns terminal y to i when there is another vowel in the stem￼. |
| 3. | Maps double suffixes to single ones: -ization, -ational, etc. |
| 4. | Deals with suffixes, -full, -ness etc. |
| 5. | Takes off -ant, -ence, etc. |
| 6. | Removes a final –e |

**Table: Snowball** describes the steps involved in stemming

**3.4.2 Lemmatization:**

Lemmatization is the process of normalizing a word rather than just finding its stem. In the process, a suffix may not only be removed, but may also be substituted with a different one. It may also involve first determining the part-of-speech for a word and then applying normalization rules. It might also involve dictionary look-up. For example, verb ‘saw’ would be lemmatized to

‘see’ and the noun ‘saw’ will remain ‘saw’. For our purpose of classifying text, stemming should be suffice.

**3.5 FEATURES:**

In machine learning and pattern recognition, a feature is an individual measurable property of a phenomenon being observed. The most widely used and basic feature set is word n-grams. An *n*-gram of size 1 is referred to as a "unigram"; size 2 is a "[bigram"](https://en.wikipedia.org/wiki/Bigram) (or, less commonly, a "digram"); size 3 is a "[trigram"](https://en.wikipedia.org/wiki/Trigram). In addition, We have experimented with these sets of features.

**3.5.1 N-GRAMS:**

N-gram refers to an n-long sequence of words. Probabilistic Language Models based on Unigrams, Bigrams and Trigrams can be successfully used to predict the next word given a current context of words. In the domain of sentiment analysis, the performance of N-grams is unclear. Some researchers report that unigrams alone are better than bigrams for classification

movie reviews, while some others report that bigrams and trigrams yield better product-review polarity classification [1]. As the order of the n-grams increases, they tend to be more and sparser. Figure: shows an example of n-grams.

Figure: n-grams

**3.5.2 UNI-GRAM:**

Unigrams are the simplest features that can be used for text classification. A Tweet can be represented by a multi-set of words present in it. We, however, have used the presence of unigrams in a tweet as a feature set. Presence of a word is more important than how many times it is repeated. The presence of unigrams yields better results than repetition [4]. This also helps us to avoid having to scale the data, which can considerably decrease training time [5]. Figure 3 illustrated the cumulative distribution of words in our dataset.

Figure: Cumulative Frequency Plot for 50 Most Frequent Unigrams. [10]

**3.5.3 BI-GRAMS:**

Figure: Cumulative Frequency Plot for 50 Most Frequent bigrams. [10]

**3.5.4 TRI-GRAMS**

Figure: Cumulative Frequency Plot for 50 Most Frequent trigrams. [10]

**4. ARCHITECTURE:**

An Architecture is usually a model that analyzes and defines the difficulties and design of the system/project. In this Architecture section, we have provided pictorial representations of our project, in the following ways:

**4.1 Use case diagram:**

We provide a use case diagram for a simplest representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

**Figure: use case** shows the use case diagram of Twitter sentiment analysis

13

**4.2 Workflow:**

**4.2.1 Hierarchical Classifier:**

We have use a hierarchy of classifiers for the sentiment analysis process. The picture below illustrates how the hierarchical classifier works in the project.

**Figure: Hierarchy** shows the process involved with hierarchical classification.

**5. EXPERIMENT:**

We use four classifiers for the classification. Firstly, we used the Naive Bayes to determine whether the tweet is polar or not. Next, if the tweet is polar then it is passed through four classifiers. This is done because there is no one best classifier and combining them improves the results drastically. Since, another classifier can handle the shortcomings of one classifier.

**5.1 CLASSIFIERS:**

**5.1.1 Naive Bayes:**

In [machine learning,](https://en.wikipedia.org/wiki/Machine_learning) **naive Bayes classifiers** are a family of simple [probabilistic classifiers](https://en.wikipedia.org/wiki/Probabilistic_classifier) based on applying [Bayes' theorem w](https://en.wikipedia.org/wiki/Bayes%27_theorem)ith strong (naive) [independence a](https://en.wikipedia.org/wiki/Statistical_independence)ssumptions between the features. It is used mostly as a baseline method in text classification [1]. One advantage of this dataset is that it only requires a small size dataset for classification.

**5.1.2 Maximum Entropy**:

The principle of Maximum Entropy states that if you take precisely stated prior data or testable information about a probability distribution function. Consider the set of all trial probability distributions that would encode the prior data. According to this principle, the distribution with maximal [information entropy is](https://en.wikipedia.org/wiki/Information_entropy) the proper one [1].

**5.1.3 NuSVC:**

SVMs are linear classifiers, which assume that your data are linearly separable [1]. In the simplified two-dimensional case, imagine that your data are plotted along the x- and y-axes. The algorithm attempts to find an optimal vector (in this case, a line) that separates them. Indeed, the algorithm would like to separate the instances by a *maximum margin.* This is so that unseen examples are more likely to be classified correctly [2].

**5.1.4 Logistic Regression:**

It is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). In our case the outcomes are positive or negative.[3]

Using these classifiers with our best bigrams and unigram features, we got a significant increase in accuracy as compared to using only the classifiers with our unigrams. In addition, we noticed there was no significant improvement when they were combined with our unigrams bigrams and trigrams. Below are graphs showing how they performed for our English, German and French Datasets.

**5.2 Comparison of the Accuracy of classifiers:**

We used Column charts to compare the accuracies of different classifiers with respect to the languages, likely English, German and French. Each graph that belongs to a particular language pictures the accuracy of Unigrams without feature selection, Unigrams with feature selection, Unigrams Bigrams with feature selection, Unigrams Bigrams Trigrams with feature selection with respect to the different classifiers namely Naive Bayes, Maximum Entropy, NuSVC and Logistic regression.

In order to find the highest information features, we had to determine the rate of information gain for each term whether it is a unigram, bigram or a trigram. The Information gain for classification is a measure of how common a feature is in a particular class compared to how common it is in all other classes.

A word that occurs primarily in positive movie reviews and rarely in negative reviews is high information. For example, the presence of the word “magnificent” in a movie review is a strong indicator that the review is positive. That makes “magnificent” a high information word.

One of the best metrics for information gain is chi square. NLTK includes this in the BigramAssocMeasures and the TrigramAssocMeasures class in the metrics package. To use it, first we need to calculate a few frequencies for each word: its overall frequency and its frequency within each class. This is done with a FreqDist for overall frequency of words, and a ConditionalFreqDist where the conditions are the class labels. Once we have those numbers, we can score words with the BigramAssocMeasures.chi\_sq function, then sort the words by score and take the top 1500 for the polarity classifier and 10000 for the subjectivity classifier since its dataset has a larger dataset.

We then put these words into a set, and use a set membership test in our feature selection function to select only those words that appear in the set. Each tweet is classified based on the presence of these high information words. [12]

In following we will mention analyze how the feature selection and the use of unigram, bigram trigrams affected the classifiers

For English language, we got higher accuracy with Logistic regression classifier for Unigram without feature selection. However, there is no huge difference with accuracies when compared to other classifiers. In addition, for Unigrams with feature selection, Max entropy classifier has the best result and it’s noticeable. Followed by Unigrams Bigrams with feature selection, Maximum entropy classifier has the best accuracy, but it’s lesser than the accuracy we got with Unigram with feature selection. Finally, with Unigrams Bigrams Trigrams with feature selection we got the best accuracy with Naive Bayes classifier which is significantly accurate than the rest. Overall, Unigram with feature selection has the best accuracy with 89.05 provided by Maximum entropy classifier.

For French language, we got higher accuracy with Logistic regression classifier for Unigram without feature selection. However, there is no huge difference with accuracies when compared to other classifiers. In addition, for Unigrams with feature selection, Naive Bayes classifier has the best result and it is noticeable. Followed by Unigrams Bigrams with feature selection, Naive Bayes classifier has the best accuracy, and it is greater than the accuracy we got with Unigram with feature selection. Finally, with Unigrams Bigrams Trigrams with feature selection we got the best accuracy with Max Entropy classifier, which is less accurate than the Unigram Bigram. Overall, Unigram Bigram with feature selection has the best accuracy with 86.54 provided by Maximum entropy classifier.

17

For German language, we got higher accuracy with Naive Bayes classifier for Unigram without feature selection. Moreover, all the classifiers in this language with Unigram without feature selection have better accuracy compared to the other languages. Furthermore, for Unigrams with feature selection, Naive Bayes classifier has the best result and it’s noticeable. Followed by Unigrams Bigrams with feature selection, Maximum entropy classifier has the best accuracy, but it’s not more accurate than the Naive Bayes classifier. Finally, with Unigrams Bigrams Trigrams with feature selection we got the best accuracy with both Naive Bayes and Max Entropy classifier that is equal to 85.4. Overall, Unigram Bigram with feature selection has the best accuracy with 85.76 provided by Maximum entropy classifier.

**5.3 Confusion matrices of the classifiers:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Languages** | **N-grams** | **Confusion Matrix** | | |
| **English** | **Unigrams without**  **Feature-selection** |  | **Predicted pol** | **Predicted neu** |
| Actual Pol | **n=2868** | 1340 | 281 |
| Actual neu | 1202 | 1247 |
| **Unigrams with**  **Feature-selection** |  | **Predicted pol** | **Predicted neu** |
| Actual Pol | **n =**  **2868** | 1440 | 181 |
| Actual neu | 45 | 1202 |
| **Unigrams**  **Bigrams with**  **Feature-selection** |  | **Predicted pol** | **Predicted neu** |
| Actual pol | **n=2868** | 1456 | 165 |
| Actual neu | 47 | 1200 |
| **Unigrams**  **Bigrams**  **Trigrams with**  **Feature-selection** |  | **Predicted pol** | **Predicted neu** |
| Actual pol |  | 1474 | 147 |

18

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Actual neu | **n=2868** | 51 | 1196 |
| **GERMAN** | **Unigrams without**  **Feature-selection** |  | **Predicted pol** | **Predicted neu** |
| Actual pol | **n=2868** | 1370 | 251 |
| Actual neu | 11 | 1236 |
| **Unigrams with**  **Feature-selection** |  | **Predicted pol** | **Predicted neu** |
| Actual pol | **n=2868** | 1483 | 138 |
| Actual neu | 23 | 1224 |
| **Unigrams**  **Bigrams with**  **Feature-selection** |  | **Predicted pol** | **Predicted neu** |
| Actual pol | **n=2868** | 1504 | 117 |
| Actual neu | 26 | 1221 |
| **Unigrams**  **Bigrams**  **Trigrams with**  **Feature-selection** |  | **Predicted pol** | **Predicted neu** |
| Actual pol | **n=2868** | 1491 | 130 |
| Actual neu | 25 | 1222 |
| **FRENCH** | **Unigrams without**  **Feature-selection** |  | **Predicted pol** | **Predicted neu** |
| Actual pol | **n=2868** | 1279 | 342 |
| Actual neu | 36 | 1211 |
| **Unigrams with**  **Feature-selection** |  | **Predicted pol** | **Predicted neu** |
| Actual pol | **n=2868** | 1400 | 221 |
| Actual neu | 54 | 1193 |

19

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Unigrams**  **Bigrams with**  **Feature-selection** |  | **Predicted pol** | **Predicted neu** |
| Actual pol | **n=2868** | 1389 | 232 |
| Actual neu | 37 | 1210 |
| **Unigrams**  **Bigrams**  **Trigrams with**  **Feature-selection** |  | **Predicted pol** | **Predicted neu** |
| Actual pol | **n=2868** | 1381 | 240 |
| Actual neu | 42 | 1205 |

Table: Confusion Matrix for subjectivity

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Languages** | **N-grams** | **Classifier s** | **Confusion Matrix** | | |
| **English** | **Unigrams without Feature- selection** | **Naive**  **Bayes** |  | **Predicted pos** | **Predicted neg** |
| Actual Pos | **n=274** | 102 | 29 |
| Actual Neg | 19 | 124 |
| Actual Pos | **Max**  **Entropy** | **n = 274** | 110 | 21 |
| Actual Neg | 23 | 120 |
| Actual pos | **NuSVC** | **n=274** | 107 | 24 |
| Actual neg | 21 | 122 |
| Actual pos | **Logistic Regressi on** | **n=274** | 107 | 24 |
| Actual neg | 20 | 123 |

20

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Unigrams with Feature- selection** |  |  | **Predicted pos** | **Predicted neg** |
| Actual Pos | **Naive**  **Bayes** | **n=274** | 110 | 21 |
| Actual Neg | 12 | 131 |
| Actual Pos | **Max**  **Entropy** | **n = 274** | 115 | 16 |
| Actual Neg | 14 | 129 |
| Actual pos | **NuSVC** | **n=274** | 106 | 25 |
| Actual neg | 15 | 128 |
| Actual pos |  | **n=274** | 106 | 25 |
| Actual neg |  | 17 | 126 |
| **Unigrams Bigrams with Feature- selection** | **Naive**  **Bayes** | **n=274** | **Predicted pos** | **Predicted neg** |
| Actual Pos | 110 | 21 |
| Actual Neg |  | 12 | 131 |
| Actual Pos | **Max**  **Entropy** | **n = 274** | 115 | 16 |
| Actual Neg | 14 | 129 |
| Actual pos | **NuSVC** | **n=274** | 106 | 25 |
| Actual neg | 15 | 128 |
| Actual pos | **Logistic Regressi on** | **n=274** | 106 | 25 |
| Actual neg | 17 | 126 |
|  | **Unigrams Bigrams Trigrams with** |  |  | **Predicted pos** | **Predicted neg** |

21

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Feature- selection** |  |  |  |  |
| Actual Pos | **Naive**  **Bayes** | **n=274** | 110 | 21 |
| Actual Neg | 12 | 131 |
| Actual Pos | **Max**  **Entropy** | **n = 274** | 115 | 16 |
| Actual Neg | 14 | 129 |
| Actual pos | **NuSVC** | **n=274** | 106 | 25 |
| Actual neg | 15 | 128 |
| Actual pos | **Logistic Regressi on** | **n=274** | 106 | 25 |
| Actual neg | 17 | 126 |
| **Languages** | **N-grams** | **Classifier s** | **Confusion Matrix** | | |
| **German** | **Unigrams without Feature- selection** | **Naive**  **Bayes** |  | **Predicted pos** | **Predicted neg** |
| Actual Pos | **n=274** | 102 | 29 |
| Actual Neg | 19 | 124 |
| Actual Pos | **Max**  **Entropy** | **n = 274** | 110 | 21 |
| Actual Neg | 23 | 120 |
| Actual pos | **NuSVC** | **n=274** | 107 | 24 |
| Actual neg | 21 | 122 |
| Actual pos | **Logistic Regressi on** | **n=274** | 107 | 24 |
| Actual neg | 20 | 123 |
| **Unigrams with Feature- selection** |  |  | **Predicted pos** | **Predicted neg** |

22

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Actual Pos | **Naive**  **Bayes** | **n=274** | 110 | 21 |
| Actual Neg | 12 | 131 |
| Actual Pos | **Max**  **Entropy** | **n = 274** | 115 | 16 |
| Actual Neg | 14 | 129 |
| Actual pos | **NuSVC** | **n=274** | 106 | 25 |
| Actual neg | 15 | 128 |
| Actual pos |  | **n=274** | 106 | 25 |
| Actual neg |  | 17 | 126 |
| **Unigrams Bigrams with Feature- selection** | **Naive**  **Bayes** | **n=274** | **Predicted pos** | **Predicted neg** |
| Actual Pos | 110 | 21 |
| Actual Neg |  | 12 | 131 |
| Actual Pos | **Max**  **Entropy** | **n = 274** | 115 | 16 |
| Actual Neg | 14 | 129 |
| Actual pos | **NuSVC** | **n=274** | 106 | 25 |
| Actual neg | 15 | 128 |
| Actual pos | **Logistic Regressi on** | **n=274** | 106 | 25 |
| Actual neg | 17 | 126 |
|  | **Unigrams Bigrams Trigrams with Feature- selection** |  |  | **Predicted pos** | **Predicted neg** |

23

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Actual Pos | **Naive**  **Bayes** | **n=274** | 110 | 21 |
| Actual Neg | 12 | 131 |
| Actual Pos | **Max**  **Entropy** | **n = 274** | 115 | 16 |
| Actual Neg | 14 | 129 |
| Actual pos | **NuSVC** | **n=274** | 106 | 25 |
| Actual neg | 15 | 128 |
| Actual pos | **Logistic Regressi on** | **n=274** | 106 | 25 |
| Actual neg | 17 | 126 |
| **Languages** | **N-grams** | **Classifier s** | **Confusion Matrix** | | |
| **French** | **Unigrams without Feature- selection** | **Naive**  **Bayes** |  | **Predicted pos** | **Predicted neg** |
| Actual Pos | **n=274** | 102 | 29 |
| Actual Neg | 19 | 124 |
| Actual Pos | **Max**  **Entropy** | **n = 274** | 110 | 21 |
| Actual Neg | 23 | 120 |
| Actual pos | **NuSVC** | **n=274** | 107 | 24 |
| Actual neg | 21 | 122 |
| Actual pos | **Logistic Regressi on** | **n=274** | 107 | 24 |
| Actual neg | 20 | 123 |
| **Unigrams with Feature- selection** |  |  | **Predicted pos** | **Predicted neg** |
| Actual Pos |  | **n=274** | 110 | 21 |

24

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Actual Neg | **Naive**  **Bayes** |  | 12 | 131 |
| Actual Pos | **Max**  **Entropy** | **n = 274** | 115 | 16 |
| Actual Neg | 14 | 129 |
| Actual pos | **NuSVC** | **n=274** | 106 | 25 |
| Actual neg | 15 | 128 |
| Actual pos |  | **n=274** | 106 | 25 |
| Actual neg |  | 17 | 126 |
| **Unigrams Bigrams with Feature- selection** | **Naive**  **Bayes** | **n=274** | **Predicted pos** | **Predicted neg** |
| Actual Pos | 110 | 21 |
| Actual Neg |  | 12 | 131 |
| Actual Pos | **Max**  **Entropy** | **n = 274** | 115 | 16 |
| Actual Neg | 14 | 129 |
| Actual pos | **NuSVC** | **n=274** | 106 | 25 |
| Actual neg | 15 | 128 |
| Actual pos | **Logistic Regressi on** | **n=274** | 106 | 25 |
| Actual neg | 17 | 126 |
|  | **Unigrams Bigrams Trigrams with Feature- selection** |  |  | **Predicted pos** | **Predicted neg** |
| Actual Pos |  | **n=274** | 110 | 21 |

25

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Actual Neg | **Naive**  **Bayes** |  | 12 | 131 |
| Actual Pos | **Max**  **Entropy** | **n = 274** | 115 | 16 |
| Actual Neg | 14 | 129 |
| Actual pos | **NuSVC** | **n=274** | 106 | 25 |
| Actual neg | 15 | 128 |
| Actual pos | **Logistic Regressi on** | **n=274** | 106 | 25 |
| Actual neg | 17 | 126 |

Table: Confusion matrix for Polarity:

**5.4 Comparison of Precision and Recall of the classifiers:**

In this section, we are comparing the Precision positives and negatives, Recall positives and negatives with respect to different languages.

**Comparison of Precision Positives and Negatives, Recall Positives and Negatives for**

**English language.**

**Comparison of Precision Positives and Negatives, Recall Positives and Negatives for**

**French language.**

**Comparison of Precision Positives and Negatives, Recall Positives and Negatives for**

**German language.**

**5.5 Comparison of F1 measures:**

**Showing F1 measure for Naive Bayes**

**Showing F1 measure for NuSVC**

**Showing F1 measure for MaxEntropy**

**5.6 10 - Fold Cross Validation:**

Cross-validation is a technique to evaluate predictive models by partitioning the original sample into a training set to train the model, and a test set to evaluate it.

In k-fold cross-validation, the original sample is randomly partitioned into k equal size subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k-1 subsamples are used as training data. The cross-validation process is then repeated k times (the folds), with each of the k subsamples used exactly once as the validation data. The k results from the folds can then be averaged (or otherwise combined) to produce a single estimation. The advantage of this method is that all observations are used for both training and validation, and each observation is used for validation exactly once.

We provide few cross validation tables below, from the cross validations we performed using multiple classifiers on English, German and French respectively.

**Best Unigram-Naïve Bayes positive**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fold number** | **Accuracy** | **Precision** | **Recall** | **F-Score** |
| 0 | 0.903 | 0.902 | 0.902 | 0.902 |
| 1 | 0.855 | 0.805 | 0.852 | 0.828 |
| 2 | 0.841 | 0.882 | 0.769 | 0.821 |
| 3 | 0.841 | 0.810 | 0.833 | 0.821 |
| 4 | 0.841 | 0.81 | 0.833 | 0.821 |
| 5 | 0.902 | 0.9 | 0.9 | 0.899 |
| 6 | 0.878 | 0.921 | 0.833 | 0.875 |
| 7 | 0.865 | 0.829 | 0.894 | 0.86 |
| 8 | 0.817 | 0.833 | 0.813 | 0.823 |
| 9 | 0.865 | 0.897 | 0.833 | 0.864 |

Table: Unipos shows the 10fold cross validation of Best Unigram positive.

**Best Unigram-Naïve Bayes negative**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fold number** | **Accuracy** | **Precision** | **Recall** | **F-Score** |
| 0 | 0.903 | 0.904 | 0.904 | 0.904 |
| 1 | 0.855 | 0.893 | 0.857 | 0.875 |
| 2 | 0.841 | 0.812 | 0.906 | 0.857 |
| 3 | 0.841 | 0.866 | 0.847 | 0.857 |
| 4 | 0.841 | 0.904 | 0.904 | 0.904 |
| 5 | 0.902 | 0.840 | 0.925 | 0.880 |
| 6 | 0.878 | 0.902 | 0.840 | 0.870 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7 | 0.865 | 0.880 | 0.840 | 0.860 |
| 8 | 0.817 | 0.80 | 0.820 | 0.810 |
| 9 | 0.865 | 0.837 | 0.90 | 0.867 |

Table: Unineg shows the 10fold cross validation of Best Unigram negative.

**Best Unigram Bigram-Max Entropy positive**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fold number** | **Accuracy** | **Precision** | **Recall** | **F-Score** |
| 0 | 0.831 | 0.704 | 0.968 | 0.815 |
| 1 | 0.915 | 0.891 | 0.916 | 0.904 |
| 2 | 0.780 | 0.729 | 0.875 | 0.795 |
| 3 | 0.853 | 0.864 | 0.820 | 0.842 |
| 4 | 0.865 | 0.880 | 0.860 | 0.870 |
| 5 | 0.817 | 0.826 | 0.844 | 0.794 |
| 6 | 0.756 | 0.695 | 0.842 | 0.761 |
| 7 | 0.756 | 0.764 | 0.684 | 0.722 |
| 8 | 0.804 | 0.787 | 0.860 | 0.822 |
| 9 | 0.890 | 0.857 | 0.923 | 0.888 |

Table: UniBipos shows the 10fold cross validation of Best Unigram Bigram positive.

**Best Unigram Bigram-Max Entropy negative**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fold number** | **Accuracy** | **Precision** | **Recall** | **F-Score** |
| 0 | 0.831 | 0.974 | 0.745 | 0.844 |
| 1 | 0.915 | 0.934 | 0.914 | 0.924 |
| 2 | 0.780 | 0.852 | 0.690 | 0.763 |
| 3 | 0.853 | 0.844 | 0.883 | 0.863 |
| 4 | 0.865 | 0.850 | 0.871 | 0.860 |
| 5 | 0.817 | 0.805 | 0.783 | 0.794 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6 | 0.756 | 0.833 | 0.681 | 0.749 |
| 7 | 0.756 | 0.750 | 0.818 | 0.782 |
| 8 | 0.804 | 0.828 | 0.743 | 0.783 |
| 9 | 0.890 | 0.925 | 0.860 | 0.891 |

Table: UniBineg shows the 10fold cross validation of Best Unigram Bigram negative.

**Best Unigram Bigram Trigram-nuSVC positive**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fold number** | **Accuracy** | **Precision** | **Recall** | **F-Score** |
| 0 | 0.839 | 0.878 | 0.770 | 0.821 |
| 1 | 0.843 | 0.878 | 0.763 | 0.816 |
| 2 | 0.807 | 0.812 | 0.722 | 0.764 |
| 3 | 0.768 | 0.793 | 0.638 | 0.707 |
| 4 | 0.865 | 0.857 | 0.878 | 0.867 |
| 5 | 0.804 | 0.783 | 0.783 | 0.783 |
| 6 | 0.853 | 0.829 | 0.871 | 0.850 |
| 7 | 0.890 | 0.950 | 0.680 | 0.891 |
| 8 | 0.756 | 0.833 | 0.681 | 0.749 |
| 9 | 0.756 | 0.780 | 0.744 | 0.761 |

Table: UniBiTripos shows the 10fold cross validation of Best Unigram Bigram Trigram Positive.

**Best Unigram Bigram Trigram-nuSVC negative**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fold number** | **Accuracy** | **Precision** | **Recall** | **F-Score** |
| 0 | 0.839 | 0.820 | 0.911 | 0.863 |
| 1 | 0.843 | 0.803 | 0.872 | 0.836 |
| 2 | 0.807 | 0.754 | 0.869 | 0.808 |
| 3 | 0.768 | 0.843 | 0.934 | 0.886 |
| 4 | 0.865 | 0.874 | 0.853 | 0.864 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5 | 0.804 | 0.822 | 0.822 | 0.822 |
| 6 | 0.853 | 0.878 | 0.837 | 0.857 |
| 7 | 0.890 | 0.857 | 0.923 | 0.888 |
| 8 | 0.756 | 0.695 | 0.842 | 0.761 |
| 9 | 0.756 | 0.731 | 0.769 | 0.750 |

Table: UniBiTrineg shows the 10fold cross validation of Best Unigram Trigram negative.

**6. Future work:**

Sentiment Analysis for Sarcasm will be the next step in sentiment analysis. In this project, we only dealt with positive and negative sentiments. As sarcasm is a trend, it will be the next step. Upgrading datasets, which is a pertinent work that can be done in the future. As the vocabulary keeps updating and many of the vocabularies are not in use any more, also Emoticons are very common nowadays; updating datasets on a regular basis will have a huge impact on the sentiment analysis. “Web crawler scraper” is a method that can be used for this process. Generalizing Datasets**,** which will be the next important feature to notice. In our project, we used datasets of Movie reviews and technologies. In order to get more accurate results, generalizing the datasets will be the solution. Implementation of language translation, which is the other better way to deal with multilingual sentiment analysis. Instead of using datasets in different languages, one can translate the testing data from different languages to English and pass it to the classifiers, as the English language is easy to handle and has a better result.

**7. Conclusion:**

Sentiment analysis is an evolving field with a variety of use applications. Although sentiment analysis tasks are challenging due to their natural language processing origins, much progress has been made over the last few years due to the high demand for it. Not only do companies want to know how their products and services are perceived by consumers (and compare to competitors), but consumers want to know the opinions of others before making buying decisions. The growing need for product insights – and the technical challenges currently facing the field –will keep sentiment analysis and opinion mining relevant for the near future. Next- generation opinion mining systems need a deeper bind between complete knowledge bases with reasoning methods inspired by human thought and psychology. This will lead to a better understanding of natural language opinions and will more efficiently bridge the gap between unstructured information in the form of human thoughts and structured data that can be analyzed and processed by a machine. The result: intelligent opinion mining systems capable of handling semantic knowledge, making analogies, continuous learning, and detecting emotions

— leading to highly efficient sentiment analysis [11].

**References:**

[1] Naive Bayes. (2016). Naive Bayes - Wikipedia, the free encyclopedia. Retrieved from [https://en.wikipedia.org/wiki/Naive\_Bayes\_classifier ([](https://en.wikipedia.org/wiki/Naive_Bayes_classifier)Online; accessed 12.10.2016])

[2] Akihiro Matsukawa (2014). Support Vector Machines - Quora. Retrieved from <https://www.quora.com/What-does-support-vector-machine-SVM-mean-in-laymans-terms> ([Online; accessed 13.10.2016])

[3] Logistic Regression (2013). Logistic Regression – Quora. Retrieved from [https://www.medcalc.org/manual/logistic\_regression.php (](https://www.medcalc.org/manual/logistic_regression.php)[Online; accessed 14.10.2016])

[4] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2): pages 1-135, 2008.

[5] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *Processing*, pages 1-6, 2009.

[6] Natalia Vyrva. Sentiment Analysis in Social Media. Pages 1-2, 2016

[7] Kate Dunham (2014). The Beginner’s Guide to Social Media Metrics: Sentiment – Hootsuite. Retrieved from [https://blog.hootsuite.com/beginners-guide-sentiment/ ([](https://blog.hootsuite.com/beginners-guide-sentiment/)Online; accessed

14.10.2016])

[8] Twitter Help Center (2016). Getting started with Twitter – Twitter. Retrieved from [https://support.twitter.com/articles/215585 ([](https://support.twitter.com/articles/215585)Online; accessed 14.10.2016])

[10] Yogesh Garg (2011). Sentiment Analysis of Twitter Feeds – GitHub. Retrieved from [https://github.com/yogeshg/Twitter-Sentiment ([O](https://github.com/yogeshg/Twitter-Sentiment)nline; accessed 13.10.2016])

[11] Ashish Katrekar (2016). An Introduction to Sentiment Analysis. Pages 3-4, 2016.

Retrieved from [https://www.globallogic.com/wp-content/uploads/2014/10/Introduction-to-](https://www.globallogic.com/wp-content/uploads/2014/10/Introduction-to-Sentiment-Analysis.pdf) [Sentiment-Analysis.pdf ([O](https://www.globallogic.com/wp-content/uploads/2014/10/Introduction-to-Sentiment-Analysis.pdf)nline; accessed 14.10.2016])

[12] Jacob Perkins (2010). Text Classification for Sentiment Analysis.

Retrieved from [http://streamhacker.com/2010/06/16/text-classification-sentiment-analysis-](http://streamhacker.com/2010/06/16/text-classification-sentiment-analysis-eliminate-low-information-features/) [eliminate-low-information-features/ ([O](http://streamhacker.com/2010/06/16/text-classification-sentiment-analysis-eliminate-low-information-features/)nline; accessed 05.07.2016])