User Interest Classification for Effective Marketing

Aviek Singh
Karthik Rumalla
Salim Zhulkhrni Shajahan
Dharanidharan Ramasamy Karuppanasamy

Syracuse University

Electrical Engineering & Computer Science EECS

Preface

This work has been carried out at Department of Electrical Engineering Computer science, Syracuse University, New York during the period March to April of 2018. This work has been carried out under the supervision of Mr. Martin Harrison, Assistant Professor at Department of Electrical Engineering .

Acknowledgements

We would like to thank our professor Mr. Martin Harrison for his support throughout the project for creating an opportunity for this project, for providing valuable suggestions on the field of work that could be researched on and his support and discussions on Sentiment Analysis and Classification problems. Finally, we thank our friends in the rich internet community and forums whose valuable suggestions had been of immense use to us.

Abstract

This project is to design a recommendation model that helps organization identify their target audience based on users interests and promote special offers and ads thereby enhancing the organizations marketing strategies. It is done by extracting data from social network (twitter in our project) and utilizing the captured data by processing it in order to make an analysis on users interests by using language processing and n grams for categorization of data. The data of users interests are collected from twitter in the form of tweets, retweets and favorites. The aim of this project to make a research in the field of natural language processing and n-grams in order to find and implement an effective marketing strategy by measuring real-time tweets and retweets made by user on any topic based on that. In collecting the data, we used tweepy API to extract tweets, retweets and favorites. In continuation to that the twitter entries were classified into relevant categories with an accuracy of 85% by using natural language processing algorithms where an investigation was made on developing a method to analyze positive sentiments of tweets, retweets and favorites resulting in a satisfying 90.45% accuracy for this mining application. The interests of a individual user are depicted in the form of pie chart using matplotlib and overall user interests are depicted in the form of bar graph using bokeh. The methods used in this project can be used for any twitter user account with public opinions.
Key Words: Natural Language Processing, Text categorization, Sentiment Analysis, Opinion mining, n-grams.

Contents

Chapter 1

Introduction

Web data mining generally refers to crawling through the web locating and fetching from pages containing desired valuable information mostly with the use of web crawlers. Web crawlers can be built to fetch information of desired target or in other words they can be made application specific. They find high applications in search engines to give up-to-date information. Nowadays social networks have covered hundreds of millions active and passive web users around the planet. The fast and exponential growth of social networking sites has proven undeniable, facilitating interconnection between users and high rate of information exchange. According to the Nielsen report in March 2009, Social Networking has been the global consumer phenomenon of 2008. Two-thirds of the worlds internet population visits a social network or blogging site and the sector now accounts for almost 10% of all internet time. With this amount of large user information exchange, social media have become a good platform for research and data mining. The valuable information retrieved from social networking sites can be utilized in many ways one of which can be to study, understand and give market strategies for specific user interests which is very essential to improve the growth of the respective company. Due to scrutiny certain social networking sites are continuously updating their user-dependent privacy policies for their users, which in turn are becoming a bit of a challenge for mining them. After collecting the desired information, the most important part would be to understand the contents of this information. This where natural language processing comes into play. NLP is a field of computer science and linguistics concerned with the interactions between computers and human (natural) languages. One specific application in NLP that can be used for this purpose is sentiment analysis. It can be used to identify and extract subjective information from the information source collected. With all these processes and methods, it is possible to build a system which can extract application dependent information, process it and produce data which can be used

for studying and deductions based on the information retrieved. After collecting the refined data by applying sentiment analysis to get the positive sentiments of tweets, retweets and favorites, we use n-grams to categorize the data and identify the top 10 most used words in 1-gram, 2-grams and 3-grams achieving a maximum of 30 words. From these, we achieve top 10 words by taking intersection of top frequencies of unigram, bigrams and trigrams. The output achieved is then depicted in the form of pie chart for individual users and in the bar graph we show overall interests from a set of users with the help of matplotlib and bokeh.

1.1 Motivation

With the rise of interconnectivity in our world with the different networks we have and with the amount of information shared, it is becoming highly important for harnessing this information on the web for various reasons. Based on the information collected applications such as market and stock predictions can be put into use. Especially this project focuses on its purpose in industries which are releasing their new products on the market will be eying on how the user interests are in order to improve the relation between them and their customers. The data can be analyzed to check the interests of users are, which then can be given as a feedback for the desired industry. It can open an area of research in solving the specified problem. There could be different approaches to it. The one and foremost most method which this project is trying to apply the use of artificial intelligence and machine learning techniques. Sentiment analysis and different clustering and categorizing algorithms such as Bayesian methods and n-grams methods are well established methods and widely used. The main purpose of this project is to collect data using a Tweepy API, which focuses on the compromise between data quantity and quality, along with different APIs and also process the data from twitter and different data by the use of the above-mentioned algorithms.

1.2 Limitations

During data collection one issue to be raised is making a compromise between large data extraction and low quality and lesser amount of data extracted with high quality. Writing a specific program for general data collection is quite challenging. One of the limitations of this project was the current model uses finite words in the categories to display the users interests. So, when a new word or category is encountered on

runtime, it will throw an error since the category list is finite. However, this can be automated using prediction and inference techniques of machine learning. Another limitation in this project was during the sentiment analysis phase where statements made by people are not always in correct grammar and with spelling errors. Also, there is an increase time consumption when the model is run for large datasets which is another limitation.

1.3 Related Work

This high rate of growth of information shared on the web has taken the attention of some researchers to use this information to analyze, predict situation based on the study. Some of the related researches are about analyzing the information for commercial purposes. One study done by Bo Pang, Lillian Lee and Shivakumar Vaithyanathan , focuses on classifying different movie reviews into three categories namely; positive, negative and neutral. They used different machine learning methods namely nave bayesian method, support vector machines and maximum entropy method. In their research they collected a set of proposed negative and positive words for a movie review from an audience and used the three methods to classify whether a review is in one of the three categories. The reviews were taken from internet movie database (IMDb). With all the methods by changing the size of the word list they were able to achieve a peak accuracy of 82.9%. The work done by Soo-Min Kim and Eduard Hovy in the paper Determining the sentiment of opinions [21], is related to our thesis work in the context of grammatical parsing of sentences for sentiment detection. We do proceed in a similar way during the sentiment analysis initial phase but do not follow the region-based scoring method. Several other attempts have been made on developing applications based upon sentiment analysis and can be found in the bibliography section. However we found many of the works to be focused on certain domains, few works are focused on relatively simple machine learning techniques which do not attempt to solve sentiment analysis to a depth, some work which are promising for grammatically correct sentences did not work on real time data. Hence we aim to develop a sentiment based analysis system that is used upon real time twitter data to show the output of users interests.

Chapter 2

Theory

The main driving idea behind the project is collecting data concerning different user interests and analyzing the captured data to see how these specific user interests are performing on the market. Basically we aim to create a much generalized data retrieval engine and inference system that can infer aggregate opinions of the public user interest. User interests can belong to any domain say Electronics, Sports, Movies, etc.. Public opinions are mined from Social Network like Twitter (that is used in this project). Inference we produce is based on the concept of Sentiment Analysis/Opinion Mining. Sentiment Analysis aims at making the system understand the Natural language expressed by people, fit a numerical score to the opinions in a range of positive/negative values. Tweets, retweets and favorites given by actual users are very important than the information you can get from reviews/advertisements of the Company itself. We capture this piece of valuable information, process it and give you the results, seeing which you will be in a comfortable position in making further decision about the user interest.

2.1 Tools

The main tasks to accomplish in this project are:

- Collecting data from social media site like twitter.

- Analyzing the data using sentiment analysis and categorizing the nouns into their respective category.

For the above broad tasks, we have used different tools and performed different modules for each subtask to come under them. Following are the major tools used in our project source code implementation work.

2.1.1 Tweepy API

Tweepy is a Python 2.6, 2.7, and 3.x library for accessing Twitter. It provides access to all Twitter RESTful API methods, including reading and posting of tweets. Tweepy supports OAuth authentication, as BasicAUTH is no longer supported by the Twitter API. The main difference between Basic and OAuth authentication are the consumer and access keys. With Basic Authentication, it was possible to provide a username and password and access the API, but since 2010 when the Twitter started requiring OAuth, the process is a bit more complicated. Both the Twitter search API and streaming API are available in tweepy. The browser advanced search function is separate and has been implemented in this github repository.

2.1.2 Nltk.sentiment.vader

Sentiment analysis is simply the process of working out (statistically) whether a piece of text is positive, negative or neutral. The majority of sentiment analysis approaches take one of two forms: polarity-based, where pieces of texts are classified as either positive or negative, or valence-based, where the intensity of the sentiment is taken into account. VADER belongs to a type of sentiment analysis that is based on lexicons of sentiment-related words. In this approach, each of the words in the lexicon is rated as to whether it is positive or negative, and in many cases, how positive or negative. Below you can see an excerpt from VADERs lexicon, where more positive words have higher positive ratings and more negative words have lower negative ratings.

2.1.3 Nltk.corpus.stopwords

The process of converting data to something a computer can understand is referred to as **pre-processing**. One of the major forms of pre-processing is to filter out useless data. In natural language processing, useless words (data), are referred to as stop words. A stop word is a commonly used word (such as the, a, an, in) that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query. We would not want these words taking up space in our database or taking up valuable processing time. For this, we can remove them easily, by storing a list of words that you consider to be stop words.

NLTK (Natural Language Toolkit) in python has a list of stopwords stored in 16 different languages. You can find them in the nltk_data directory.

2.1.4 Nltk.tokenize

Tokenization is a way to split text into tokens. These tokens could be paragraphs, sentences, or individual words. NLTK provides a number of tokenizers in the tokenize module. The text is first tokenized into sentences using the PunktSentenceTokenizer. Then each sentence is tokenized into words using 4 different word tokenizers: TreebankWordTokenizer, WordPunctTokenizer, PunctWordTokenizer, WhitespaceTokenizer. The pattern tokenizer does its own sentence and word tokenization, and is included to show how this library tokenizes text before further parsing.

2.1.5 Nltk.ngrams

Ngrams (generalized way to say unigrams, bigrams, trigrams, so on) is an n-tuple of items taken from the text, usually n contiguous words. Most common use of ngrams is to find occurrence count of ngrams in corpus (set of documents) and then use their relative occurrence count. Google Ngrams obtain this data from the books that it has digitized. The datasets available have counts by year, and within the year they have number of occurrences and number of distinct books that included the ngram.

2.1.6 Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-orientedAPI for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged.[2] SciPy makes use of matplotlib. It was originally written by John D. Hunter, has an active development community, and is distributed under a BSD-style license. Michael Droettboom was nominated as matplotlib's lead developer shortly before John Hunter's death in 2012, and further joined by Thomas Caswell.

2.1.7 Tkinter

Tkinter is a Python wrapper for Tcl/Tk providing a cross-platform GUI toolkit. On Windows, it comes bundled with Python; on other operating systems, it can be installed. The set of available widgets is smaller than in some other toolkits, but

since Tkinter widgets are extensible, many of the missing compound widgets can be created using the extensibility, such as combo box and scrolling pane. IDLE, Python's Integrated Development and Learning Environment, is written using Tkinter and is often distributed with Python.

2.1.8 Bokeh

Bokeh is an interactive visualization library that targets modern web browsers for presentation. Its goal is to provide elegant, concise construction of versatile graphics, and to extend this capability with high-performance interactivity over very large or streaming datasets. Bokeh can help anyone who would like to quickly and easily create interactive plots, dashboards, and data applications.

Chapter 3

Experiment

For sake of research and experimental purposes we collect data on users intersts from his/her tweets, retweets and favorites. All the data about each product are collected from twitter by using tweepy API for collecting data are discussed in following sections. The reason behind all the test interests happen to be users interests is because ample data is available on the same. In fact, with the following proposed methods and techniques its possible to evaluate any user interests trends on the web. All the data are stored in a JSON format for further use in the sentiment analysis phase. On the collected data, we perform two levels of sentiment analysis, sentiment level and document level. As it should be obvious from the table, sentiment level analysis is carried out on the tweets, retweets and favorites, and document level is based on contents of a website (which is not taken as a reference to this project). After performing the sentiment analysis, we arrive at the text categorization phase. Here numerous methods may be used to categorize the words fetched from the tweets, such as machine learning algorithms, classification, etc. But here, we are using n-grams data structure to get top words from the list of nouns and depicting it to its particular group. Inferences produced are given a user interests,

- What are the important features of a user interests and their respective numerical scores in the range -1 to +1. Negative score implies the particular feature is disliked by the public that is to be removed from the category and positive score implies it is appreciated by the public. This inference is produced by two methods of sentiment analysis.
- What are the nouns used from the tweets of a particular user.
- What are the negative tweets to be removed from the list of tweets.

- What are the most spoken topics from a user to segregate to their respective categories.

Chapter 4

Modules

4.1 Data Collection

4.1.1 Twitter Data Collection

Twitter is a popular social media place to look for information about literally any product. Also, the nature of tweets in twitter that its length cannot be greater than 140 characters interests us because we deal with sentence level sentiment detection in the following sections. Twitter provides an open source library to access the tweets programmatically, there are many wrappers developed and in our case we use the tweepy. The API (application program interface) allows to input a search query along with various other preferences like

- Location of the tweet
- Language of the tweet, we are interested in only English tweets since we do not focus on detecting sentiments of other languages
- Username of the person who tweeted.
- Tweets between certain period, or tweets since a date or tweets until a date. Twitter does not provide any data older than a week hence it is necessary to run the data retrieval program continuously in order to obtain updated tweets.
- Re-tweets, we can fetch the replies to a tweet also.
- Attitudes. Twitter has inbuilt program that classifies a tweet into positive, negative and neutral attitudes, pretty much the same we are attempting to do. But this feature was not made available in the tweepy library we use and could be accessed only when searched manually through a browser. With all

the above features we can collect tweets of our choice. It was possible to collect the comments with all the features except attitude using our program. It was possible to collect 200 comments per run and managed to collect more comments in a real time basis, different amounts for each product. Using twitter data collector program we organize the collected data into database with respect to their user interests. Now that data is on hand we move to sentiment analysis module to associate the data with numerical score.

4.1.2 Data Cleansing

Data cleansing or data cleaning is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data. Data cleansing may be performed interactively with data wrangling tools, or as batch processing through scripting. After cleansing, a data set should be consistent with other similar data sets in the system. The inconsistencies detected or removed may have been originally caused by user entry errors, by corruption in transmission or storage, or by different data dictionary definitions of similar entities in different stores. Data cleaning differs from data validation in that validation almost invariably means data is rejected from the system at entry and is performed at the time of entry, rather than on batches of data. Here data cleansing is done by removing the http links present with the tweets and almost all the stop words present in them.

4.2 Sentiment Analysis

The aim of sentiment analysis is to detect the sentiment of a comment/sentence. A sentence can speak something positive, negative or can imply a neutral opinion about a feature. To be more exact, a sentence can imply positive opinion on some features, be neutral on some features and express a negative opinion on some features. So, it is desirable to identify the features in a sentence, identify the opinions, and also identify which opinions are targeted to which of the features. Before going into the core pseudo code that generates the score we shall take a look at few concepts that are used in the algorithm.

4.2.1 Avoid Stop Words

When working with text mining applications, we often hear of the term stop words or stop word list or even stop list. Stop words are just a set of commonly used words in any language. Stop words are commonly eliminated from many text processing applications because these words can be distracting, non-informative (or non-discriminative) and are additional memory overhead.For example, in the context of a search engine, let us assume that your search query is how to implement the BM25 retrieval formula. If the search engine tries to find web pages that contain the terms how, to implement, BM25 , retrieval, formula the search engine is going to find a lot more pages that contain the terms how, to and the than pages that contain information about implementing the BM25 formula because the terms how, to and the are so commonly used in the English language.

4.2.2 Sentiment Rules

Given the sentence we apply certain rules in the initial stage of sentiment analysis. These rules when applied produce a result on elements of the sentence. These rules are followed from a previous work on Sentiment detection. These scores are used further in the algorithm. Rule 1: Given a sentence assign +1 for positive opinion words, -1 for negative opinion words and 0 for context dependent words. Identifying opinion words are described earlier. Context dependent words are those related to the product in picture. For example, words like camera, battery, design, app, keypad, music etc. are context dependent words of any mobile phone. Rule 2: This rule handles negation in a sentence. The word not negates the meaning conveyed by the word succeeding it, and the word in succession is most probable to be an opinion word. In such a case it is important to handle negation of the opinion terms. For example consider the sentence, Samsung is not a good smart phone. The not negates the opinion term good, good has a score of +1 from rule1, and now after applying negation rule good obtains a score of -1. Rule 3: This rule handles but clauses in a sentence. Part of sentence preceding and succeeding the but clause are usually oriented opposite in meaning to each other. For example in the sentence Samsung wave looks awesome but its price is costly, style is spoken good and price is spoken bad. Since we know awesome is a positive opinion term we 13 can now infer that some negative is spoken about the context dependent word price. Similarly is the case if we find an opinion term after the but clause. Rule 4: This rule handles comparative

sentences. Comparative sentences are those which uses than, better than, higher than etc. to express positiveness about a user interest.

4.3 Text categorization

Text categorization (a.k.a. text classification) is the task of assigning predefined categories to free-text documents. It can provide conceptual views of document collections and has important applications in the real world. For example, news stories are typically organized by subject categories (topics) or geographical codes; academic papers are often classified by technical domains and sub-domains; patient reports in health-care organizations are often indexed from multiple aspects, using taxonomies of disease categories, types of surgical procedures, insurance reimbursement codes and so on. Another widespread application of text categorization is spam filtering, where email messages are classified into the two categories of spam and non-spam, respectively. To apply text categorization, n-grams are used in this project where a series of unigram, bigrams, trigrams are taken into consideration. Top 10 frequency words are taken from each of unigram, bigrams and trigrams and searches in each category present and displays the number of words present in each category for a particular user.

N-Grams: : In the fields of computational linguistics and probability, an n-gram is a contiguous sequence of n items from a given sample of text or speech. The items can be phonemes, syllables, letters, words or base pairs according to the application. The n-grams typically are collected from a text or speech corpus. When the items are words, n-grams may also be called shingles. Using Latin numerical prefixes, an n-gram of size 1 is referred to as a "unigram"; size 2 is a "bigram" (or, less commonly, a "digram"); size 3 is a "trigram". English cardinal numbers are sometimes used, e.g., "four-gram", "five-gram", and so on. In computational biology, a polymer or oligomer of a known size is called a k-mer instead of an n-gram, with specific names using Greek numerical prefixes such as "monomer", "dimer", "trimer", "tetramer", "pentamer", etc., or English cardinal numbers, "one-mer", "two-mer", "three-mer", etc. An n-gram model models sequences, notably natural languages, using the statistical properties of n-grams. This idea can be traced to an experiment by Claude Shannon's work in information theory. Shannon posed the question: given a sequence of letters (for example, the sequence "for ex"), what is the likelihoodof the next letter? From training data, one can derive a probability distribution for the next letter given a history of size n: a = 0.4, b = 0.00001, c = 0, ...; where the probabilities of

all possible "next-letters" sum to 1.0. More concisely, an n-gram model predicts xi based on xi-(n-1),,xi-1. In probability terms, this is P(xi — xi-(n-1),,xi-1) . When used for language modeling, independence assumptions are made so that each word depends only on the last n 1 words. This Markov model is used as an approximation of the true underlying language. This assumption is important because it massively simplifies the problem of estimating the language model from data. In addition, because of the open nature of language, it is common to group words unknown to the language model together. Note that in a simple n-gram language model, the probability of a word, conditioned on some number of previous words (one word in a bigram model, two words in a trigram model, etc.) can be described as following a categorical distribution (often imprecisely called a "multinomial distribution"). In practice, the probability distributions are smoothed by assigning non-zero probabilities to unseen words or n-grams; see smoothing techniques. n-grams find use in several areas of computer science, computational linguistics, and applied mathematics. They have been used to:

- design kernels that allow machine learning algorithms such as support vector machines to learn from string data
- find likely candidates for the correct spelling of a misspelled word
- improve compression in compression algorithms where a small area of data requires n-grams of greater length
- assess the probability of a given word sequence appearing in text of a language of interest in pattern recognition systems, speech recognition, OCR (optical character recognition), Intelligent Character Recognition (ICR), machine translation and similar applications
- improve retrieval in information retrieval systems when it is hoped to find similar "documents" (a term for which the conventional meaning is sometimes stretched, depending on the data set) given a single query document and a database of reference documents
- improve retrieval performance in genetic sequence analysis as in the BLAST family of programs
- identify the language a text is in or the species a small sequence of DNA was taken from

- predict letters or words at random in order to create text, as in the dissociated press algorithm.

1. Here we extract top 10 words in unigram. Code for it is given below
   # for single words
   $\text{freq}_s ingle = Counter(nouns_t ext)$
   $for token, countin freq_s ingle.most_c ommon(10):$
   $all_s ingle.append(token)$

2. Here we extract top 10 words in bigram. Code for it is given below
   # for bigrams
   $\text{bigrams} = \text{list}(\text{ngrams}(\text{nouns}_t ext, 2))$
   $freq_b i = Counter(bigrams)$
   $for token, countin freq_b i.most_c ommon(10):$
   $all_b igrams.append(list(token))$

3. Here we extract top 10 words in bigram. Code for it is given below
   #for trigrams
   $\text{trigrams} = \text{list}(\text{ngrams}(\text{nouns}_t ext, 3))$
   $freq_t ri = Counter(trigrams)$
   $for token, countin freq_t ri.most_c ommon(10):$
   $all_t rigrams.append(list(token))$

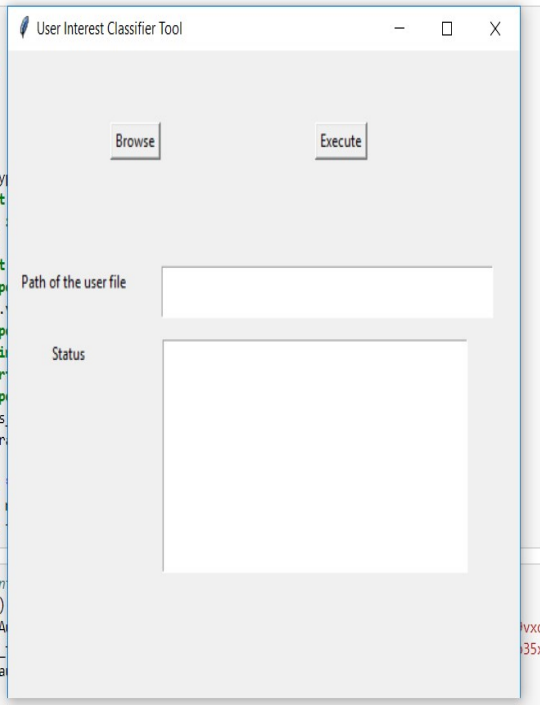
4.4 Data Visualization

Data visualization is viewed by many disciplines as a modern equivalent of visual communication. It involves the creation and study of the visual representation of data, meaning "information that has been abstracted in some schematic form, including attributes or variables for the units of information". A primary goal of data visualization is to communicate information clearly and efficiently via statistical graphics, plots and information graphics. Numerical data may be encoded using dots, lines, or bars, to visually communicate a quantitative message.[2] Effective visualization helps users analyze and reason about data and evidence. It makes complex data more accessible, understandable and usable. Users may have particular analytical tasks, such as making comparisons or understanding causality, and the design

principle of the graphic (i.e., showing comparisons or showing causality) follows the task. Tables are generally used where users will look up a specific measurement, while charts of various types are used to show patterns or relationships in the data for one or more variables. Data visualization is both an art and a science. It is viewed as a branch of descriptive statistics by some, but also as a grounded theory development tool by others. Increased amounts of data created by Internet activity and an expanding number of sensors in the environment are referred to as "big data" or Internet of things. Processing, analyzing and communicating this data present ethical and analytical challenges for data visualization. The field of data science and practitioners called data scientists help address this challenge. Data visualization refers to the techniques used to communicate data or information by encoding it as visual objects (e.g., points, lines or bars) contained in graphics. The goal is to communicate information clearly and efficiently to users. It is one of the steps in data analysis or data science. Data visualization is closely related to information graphics, information visualization, scientific visualization, exploratory data analysis and statistical graphics. In the new millennium, data visualization has become an active area of research, teaching and development. According to Post et al. (2002), it has united scientific and information visualization. The picture above refers to how users can provide his input by clicking on the browse button and the path of that file appears in the text box. Status in this picture illustrates about the execution that is carried on at the back-end side. When we click on execute button, the code is run at the background to generate multiple pie charts and a bar chart.

Here, the bar graph we show is the overall interests from a set of users which is generated with the help of bokeh.

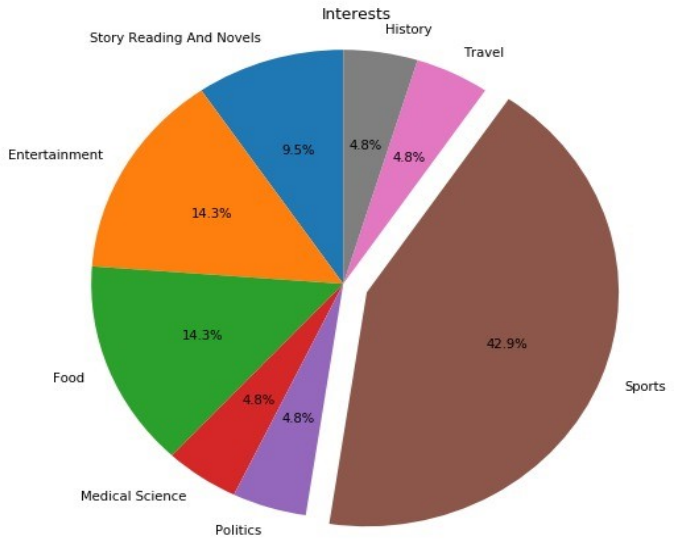
In [1]:
```
import tweepy
import json
import pandas as pd
import copy
import nltk
import numpy as np
import re
import matplotlib.py
from bokeh.io import
from bokeh.plotting
from math import pi
from operator import
from collections imp
from nltk.sentiment.
from nltk.corpus imp
from nltk.tokenize i
from nltk.stem impor
from collections imp
from nltk import pos
from nltk import ngr
import tkinter
from tkinter import
from tkinter import
from tkinter import
```

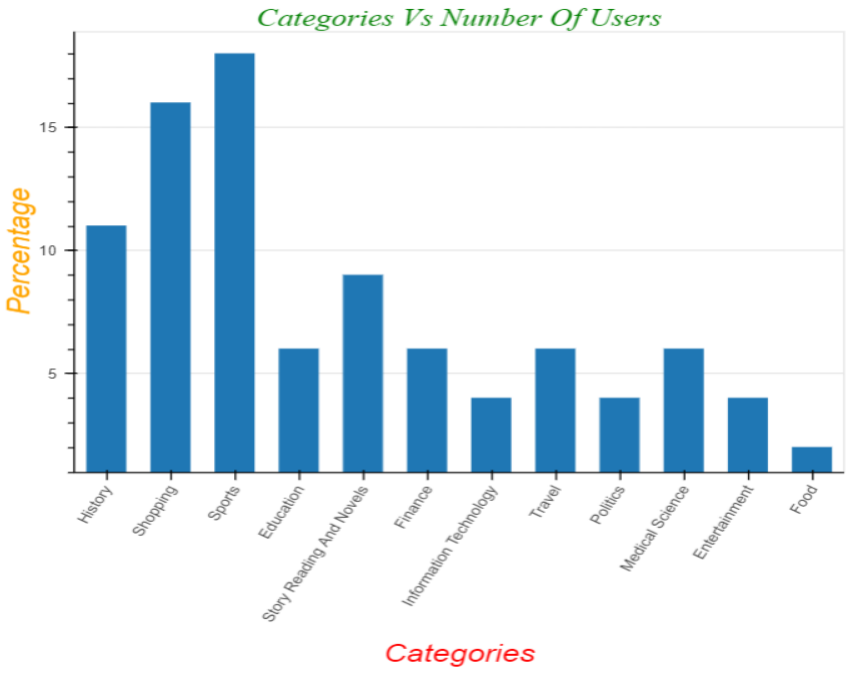


In [2]:
```
# function to authen
def authentication()
    auth = tweepy.OA
    auth.set_access_
    api=tweepy.API(a
    return api
```

```
vxo8RqmlZfzUmkREC')
35xKWiSEr5KeMvoa1e2IYMq8e79j21pg')
```

	Category	Words	Percentage
0	Story Reading And Novels	2	9
1	Entertainment	3	14
2	Food	3	14
3	Medical Science	1	4
4	Politics	1	4
5	Sports	9	42
6	Travel	1	4
7	History	1	4




Categories Vs Number Of Users
Percentage
15
10
5
History
Shopping
Sports
Education
Story Reading And Novels
Finance
Information Technology
Travel
Politics
Medical Science
Entertainment
Food
Categories

Chapter 5

Conclusion

In this project work we developed an application that provides subtle interests of a user based on public opinions. We used twitter API which is the tweepy API to collect data, though it was not the prime goal of a project our tweepy API manages to retrieve data efficiently. We process the data using Natural Language Processing and data cleansing with nouns to make the system understand the opinions expressed by people and associate them with values. The system manages to process accurately the sentences which are and not grammatically correct. We dealt with feature extraction in a sentence, opinion words identification, grammatical parsing of the sentence to understand the relationship between features and opinion terms, score the features in a sentence and finally made some statistics to produce results to the user. We collected the revised positive data and categorized into their specific categories by using n grams architecture and collecting top frequency words among the common data from unigram, bigram, and trigram thus in output achieved is then depicted in the form of pie chart for individual users and in the bar graph we show overall interests from a set of users with the help of matplotlib and bokeh. This project as a whole aims to assist the organizations in making better marketing decisions, by identifying the target audiences. The model can be enhanced by dynamically adding categorical data to improve accuracy. There was a tradeoff between Speed vs Accuracy we had to decide on right from the start of this project work. We chose to focus on Accuracy rather than speed since we thought there is a workaround for speed. We can produce a NLP score without these parsers but that is not an accurate inference, not close to the results we have now at all. Hence, we resorted to produce better results compromising speed/time. There were limitations in the system and were discussed in the end of previous section. On the merits side, we managed to develop a system that can infer about a user belonging to twitter domain, the scores generated by NLP module were good for the available data so as to retrieve the positive texts from a variety of texts.

And at last we can say that even though it might not only be these methods to use to solve the problems we proposed, our methods and algorithms have shown quite satisfactory results.

Chapter 6

Future Work

There is a lot of scope for extension and future work on our project. The following are the possible areas we can work in future:

- Improving the performance and quality of extracted data: Twitter data forms the basis of our project work since the entire module of sentiment detection depends on the data fed to it. One way to improve the performance of twitter data is to initiate the series of multiple tokens so that the large twitter request can handled at a shorter time. Higher the volume of data better is the accuracy of result. It is possible to fetch data from various other social media sites similar to twitter, e.g.: Facebook, Orkut etc. thereby collecting more data.

- Selection of Text Categorization tools: We use the N-grams to a large extent in our project for categorization. It is quite slow in processing sentences especially when sentences are long. Accuracy of the parser is not an issue since it recognizes well-formed sentences very precisely and categorizes efficiently. Hence, in our scope of future work, we can experiment on using machine learning algorithms such as using inference and prediction techniques as well as using count vectorization for larger improvement.

- Inclusion of context into our project framework might be very useful to analyze the sentences more accurately. Context means knowledge of sentences suppose for example if its a question, if the sentence is previous statements reply etc. can help immensely while giving any subject whether it is positive or negative.

- Ambiguity resolution of certain words can be done to enhance the accuracy of sentiment scoring to extract positive tweets and categorize them.

- Another interesting place we can extend our work is to extract the timeline of comments and record in a csv file. If the time of comment data is available, we can infer about the opinion on the user interests in various periods of time. This area is known as trend analysis and is considered widely by organizations to know how their users popularity towards their interest is trending from time to time.

Chapter 7

References

1. http://en.wikipedia.org/wiki/Natural_language_processing
2. https://www.pythoncentral.io/introduction-to-tweepy-twitter-for-python
3. https://www.oreilly.com/learning/how-can-i-tokenize-a-sentence-with-python
4. http://blog.alejandronolla.com/2013/05/20/n-gram-based-text-categorization-c
5. http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html
6. http://scipy-cookbook.readthedocs.io/items/Matplotlib_Interactive_Plotting.html
7. http://t-redactyl.io/blog/2017/04/using-vader-to-handle-sentiment-analysis-w html
8. https://pythonspot.com/nltk-stop-words/
9. Global Faces and Networked Places, A Neilsen report on Social Networking's New Global Footprint, March 2009. Neilsen company
10. Bing Liu. Sentiment Analysis and Subjectivity. Handbook of Natural Language Processing,2010