Document and Entity Centric Analysis of Google+ Activity Data

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Computer Science and Engineering

Submitted By

Isha Arya
(801332009)

Under the supervision of:

Dr. V. P. Singh
Assistant Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

July 2015

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "*Document and Entity Centric Analysis of Goggle+ Activity Data*", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Computer Science and Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of **Dr. V. P. Singh** and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

(Isha Arya)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr. V. P. Singh)
Assistant Professor
Computer Science and Engineering
Department

Countersigned by

(Dr. Deepak Garg)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. S. Bhatia)
Dean (Academic Affairs)
Thapar University
Patiala

Acknowledgement

The successful completion of any task would be incomplete without acknowledging the people who made it possible and whose constant guidance and encouragement secured the success.

First of all I wish to acknowledge the benevolence of omnipotent God who gave me strength and courage to overcome all obstacles and showed me the silver lining in the dark clouds. With the profound sense of gratitude and heartiest regard, I express my sincere feelings of indebtedness to my guide **Dr. V. P. Singh** for their positive attitude, excellent guidance, constant encouragement, keen interest, invaluable cooperation, generous attitude and above all their blessings. They have been a source of inspiration for me.

I am grateful to **Dr. Deepak Garg**, Head of Department and **Dr. Ashutosh Mishra**, P.G. Coordinator, Computer Science and Engineering Department, Thapar University for the motivation and inspiration for the completion of this thesis. I will be failing in my duty if I do not express my gratitude to **Dr. S. S. Bhatia**, Senior Professor and Dean of Academics Affairs in the University, for making provisions of infrastructure such as library facilities, computer labs equipped with internet facility, immensely useful for the learners to equip themselves with latest in the field.

Last but not the least I would like to express my heartfelt thanks to my parents and my friends who with their thought provoking views, veracity and whole hearted cooperation helped me in doing this thesis.

Isha Arya

Abstract

Social media has become very popular way of communication among internet users in the recent years. Humans have an ingrained tendency to share their ideas, experiences and knowledge, which associate them with the rest of the world, so that they can be recognized and can also identify their importance and worth. They are eager to know about happenings around them, that is why they communicate in order to share their ideas, observations and queries. Social media is one such communication medium that made people to be heard and satisfy their curiosity to know about rest of the world. A huge amount of unstructured data is available for analysis on the social web. The data available on these sites have redundancies as users are free to enter the data according to their knowledge and interest. This data needs to be cleaned before doing any analysis due to the presence of various redundancies in it. In this research, Google+ activities data is extracted from Google+ API. This dataset is first cleaned by removing various HTML tags and stopwords present in the activities content. This human language data is queried using TF-IDF to find out the document of interest and similarity between there document is computed using Cosine Similarity metric and similarity of documents is visualized as a matrix diagram. The various collocations present in the activity data have been analyzed. Further the summary of each activity is extracted by using Luhn's document summarization algorithm. Various entities present in post/activities are also extracted and interactions between them are analyzed.

Table of Contents

List of Figures

List of Tables

Chatper1 Introduction

Now-a-days, social networking sites have become not only very popular but also a need for the new generation. Through these sites, the people can interact with others for their common interests, actions, attitude, thinking or awareness. These sites have eliminated the barriers of time, country, language, gender and money. The users are surfing these sites for various purposes like making new friends, hunting for new job, advertising their business and finding siblings *etc*. These sites provide a platform of the real world at home for the users. Every time, everywhere connectivity makes these sites at boom rapidly in the technocrats. Recently, Facebook, Twitter, LinkedIn, Google^{+} become very popular social websites. Facebook has more than 800 million active users. These users are increasing 83% annually since 2008. Hundreds of new websites are impending every day and attract users with various offers. Social networks are the central key to comprehend common phenomenon, envisage human deeds and examine social constitution. By studying the behaviour and concept of social networks, one can understand the social phenomena. The study of social networking also helps in optimization, execution and organization of problems in reality.

The current advances in information technology and increasing use of social media make it very easy to generate a bunch of data. Therefore the big challenges in front of us are to collect and integrate these bunches of huge data from the data sources which are universally distributed. If we have a look on the past 20 years, we can see the data has increased in a large scale. In 2011, IDC (International Data Corporation) submitted a report in which, the overall data volume which was created and copied in this world was 1.8 ZB and within 5 years, it is supposed to be increased by nine times. On YouTube, every minute videos of about 72 hour are uploaded. This Big Data can be understood as huge amount of unstructured data which needs a lot of real time analysis. Various Companies generates and process lot of such data such as Google which processes hundreds of Peta Byte (PB) of data, Facebook which generates nearly 10 PB per month of log data, a Chinese company Baidu which processes tens of PB of data and Taobao which processes data of tens of PB per day for online trading.

Figure 1.1 Online Social Network as a big source of Big Data

Today, the researchers are focusing to investigate the structures and associations of various social networks. The main research area is study and analysis of compactness, centrality and grouping with the promise of security and privacy of the users data in these networks. The social networks are created and knockdown by studying daily basis and continuous communication among users. These studies incorporate with individual's relationships with others in terms of different parameters of human behaviour. In this way, these networks are proven an important and critical tool to study the communal phenomenon, composition, relationships and behaviours of individuals and/or group. The researchers are aiming to investigate these networks for various issues like number of users, increment in number of users, various applications provided by the sites, user's applications, and interest due to the reputation of these online social networks, ease in graphical investigation, the accessibility of huge amount of chronicle data, business and marketing welfares *etc*. Moreover, these Social networks show attractive and motivating research confronts like finding out and toning of identical sub-graphs, neighbourhood relationship examination and representation, subscribers grouping, characterizing, cataloguing and information proliferation *etc*. Therefore, investigation of data analysis of these online websites of social networking has an immense prospective for researchers in an assortment of branches.

1.1 Social Web Analysis

Today, Internet performs an increasingly vital role and it has gradually infiltrated into every point of our lives because of its infinite and varied services. Social Web is a platform that permits people to post their updates and to communicate to other people on the network, for example users are playing online games, tagging each other, working and socializing online, revealing new techniques of participation and communication that were hardly imaginable just a short time ago. Social Web Analysis is becoming an important technique for researchers, but all the mandatory information is often available in a distributed environment. In social web analysis the most important task is usually about how to extract social network's data from different communication services. Facebook, Flicker, Google+, Hangout, Google+ and Twitter are the very famous social media on the internet. The social media provide their user a powerful means of sharing their posts, finding their friends and content of their interest, organising online events, *etc*. From these social media sites features of online social web graphs can be learned at a higher level. This is important to understand these graphs, for to plan new features and to amend existing systems of online social network.

Recently the increasing affinity of citizens to use these Social media websites such as Twitter, LinkedIn and Facebook have resulted in building different kinds of relationships and interactions which have led to the availability of an enormous amount of valuable data that has never been accessible before. Such enormous valuable data can be utilized in some new, eye-catching, varied and valuable research areas to lot of researchers. Although the area Social web analysis has received a enormous deal of consideration in last few years, yet a lot of problems related to social web analysis is still in its immaturity and needs some more methods to be developed in the future for additional improvements.

The intense reputation and fast progression of online social communities gives a exclusive chance to analyze, learn, and furnish their features.Various research issues associated with social web analytics such as predicting future trends, community detection, link prediction, recommender system, frequency analysis, mood analysis, opinion mining, influence propagation, text mining concepts such as Document Similarity and extraction of Document of Interest *etc*. are shown in Figure 1.2. These are the latest research areas which emerged from social web analysis. The digital text

data generated from these online social networks is noisy, vast and dynamic. Efficient text mining techniques are required to analyze such complex, large and rapidly changing social data.

Figure 1.2 Research Issues Associated with Social Web Analytics

1.2 Text Mining

Text mining is defined as a procedure of extracting non-retrieval and required information from unstructured documents or a corpus. Fundamentally, text mining translates text data into numbers that can be further used in some other analysis such as clustering, data mining task for prediction *etc*. [1]. However, text mining is also a very difficult job as it deals with inherently fuzzy and unstructured text data. The text here refers to human language that contains a lot of content from which useful knowledge can be discovered. For example, magazine articles, newspaper stories, manuals, fiction and nonfiction books, email, blogs and various online articles can be considered as texts. The quantity of text generated on the web is vast and continuously increasing. Text Mining is a the procedure of analyzing the text which is naturally happening for the intention of capturing and discovering lot of semantic information for that can be stored in KOS (Knowledge Organization Structure)

through the crucial objective of gaining knowledge discovery via either visual or textual access that can be utilized in a huge range of important applications. It is the procedure of fetching high-quality information hidden in the text by using statistical pattern learning. It includes the procedure of assigning structure to a text document like successive insertion into a database and parsing. It is a procedure that converts unstructured text into structured text and contains various quantitative methods which can be used in analyzing these structured objects [1].

Text mining is suitably regarded as a feature of a wider area KDD (Knowledge Discovery from Data) that is described because it is the procedure of retrieving required facts from huge data by plotting unstructured data into a better form, more abstract forms and by detecting important configurations perfectly exists in data. It is usually performed on relational databases/structured data and includes process of data mining as its sub-procedure. The entire procedure of KDD consists of pattern detection and extraction, data cleansing, data storage and access and its interpretation, whereas data mining demotes closely to the specific stage that applies definite procedures for extracting information and detecting patterns from data.

Keyword search from web makes it difficult for network users to see the precise and accurate information from the query results. That is depends to the user to go through all document to extract the required and relevant data from search results. This task is a boring and unfeasible. For this problem text mining is a better solution as it connects all the extracted information in a group, keeps the appropriate information based on the query of a user and pushes all the unwanted information apart.

Difficulties that engage obtaining information for human consumption are beat first. For mining the simple text like assessing document similarity, text summarization, Information retrieval, text categorization and document retrieval.

1.2.1 Text Summarization

A text abstracter makes a squashed characterization of the input that defines human usage. It contains groups of documents called corpus or a specific document. Text Compression is as similar to this area but the result of text summarization is should be in a format that is easily readable by humans. The result of algorithms used for text compression is definitely not in the format understandable by humans and actions cannot be taken on the result. Only decompression is supported by the algorithm. It is

different from various other text mining approaches in which proficient abstractors are present for generating abstract [2].

1.2.2 Document Retrieval

In document retrieval the most appropriate documents are identified and sent back. Conventional libraries gives catalogues that permit various users in finding written records on the basis of metadata present in resources. Metadata is an extremely structured document for abstract, and successful approaches are there for retrieving metadata manually and extracting related records on the basis of this metadata. Metadata's extraction is a main application of text mining approaches. The motive is to catalogue each single word in a collection of documents [3].

1.2.3 Information Retrieval

Retrieval of information is considered as processing of documents to retrieve the specific information requested by a particular user. After retrieving a document, there is a task of summarizing the text which concentrates on the queries made by users, or an information extraction stage. The modularity of documents may be changed for every single subsection or paragraph, which consist its own units, so as to concentrate on outputs on distinct pieces of information instead of complete documents [4].

1.2.4 Measuring Similarity between Documents

One of text mining problem involves find similarities among various documents with in a corpus like group the set documents into ordinary clusters and allot them to pre-defined sets.

1.2.5 Text Categorization

Text categorization can be defined as the process of assigning documents written in human language to already described classes corresponding to content of the document. The set of classes here is a controlling parameter. Text categorization can be viewed as conventional method of information retrieval. It has so many useful applications, considering extracting the metadata from documents, organizing and maintaining huge amount of information stored in online resources, word meanings clarification by detecting the issues a document covers and indexing for information retrieval. For document classification, only the content of the document is necessary.

Few assignments restrict all the documents to only one class, while in some assignments one document may be present in more than one class. Sometimes probabilistic approach is used to label all the classes because our goal is to assign a class to a document based on its relevance [5].

1.3 Information Retrieval and Natural Language Processing

The primary objective of information retrieval is to make a system that gives relevant information according to the defined information requirement [6]. It is a method to search the matched information as user's requirement. Unmatched document may be retrieved simply because terms occurs unwillingly and in the another side, matched document may be dropped because no such term in the document occurs in the requested query. In information retrieval, locating a relevant documents in a corpus based on a user's request is a very usual problem, which is often few keywords defining an information need, even if it could also be an example relevant document. In these types of problem, individuals takes the initiative to drag the related information from the set of documents; this is most suitable if a user has a number of improvised information requirement, such as discovering information to buy an old car. While a user has a long-standing information requirement, a retrieval system may also take the steps to derive any recently arrived information entry to a user if the entry is determined as being useful to the user's need. This type of process of accessing information is known as information filtering, and the equivalent systems are usually called recommender systems or filtering systems. Since, as a technical perspective, however, filtering and searching share a lots of common method.

Electronic information on Web is a useful resource for users to obtain a variety of information. The retrieval performance of the information retrieval systems is largely dependent on similarity measures. Retrieving relevant text pages on a topic from a big corpus is a complex task. The information retrieval methods generally used are based on keywords. These methods use keyword lists to illustrate the content of document, but there is a problem with these types of lists that is, they do not understand the semantic relationships between keywords, meaning of words included in the content and phrases [7]. The significance of information retrieval has rapidly increased since the appearance of the World Wide Web (WWW) and its expansive volume of electronic documents. Information Retrieval has become a part of many people's daily lives. While ordinary users may not be familiar with the term

Information Retrieval, they are certainly well-familiarize with web-based search engines like Google, Yahoo, Ask, *etc*. Now days Information Retrieval is very famous, and it has become an exciting research field because of its prevalent presence in day-to-day life.

The main objective of every Information Retrieval method is to respond to their user's requested information by giving the reference documents that meet the requested information. Different aspects determine whether or not the information is relevant to a user's query.

There is difference between language processing and information retrieval. Language processing attempts to convey the exact meaning of the text but information retrieval goal tries to retrieve a particular document. Nevertheless, it is reasonable that retrieving a particular item of a certain subject requires all available related facts from the analysis of meaning of language understanding [8].

NLP (Natural Language Processing) is a branch of artificial intelligence that deals in processing text written in human language. Natural Language Processing Toolkit (NLTK) is used for performing effective and efficient functionality. NLTK also provides list of stopwords and this list is used to pull out the terms like *the, and, a,* and *is,* the text can contain such words that are avoided even we have finest stopwords lists. Although we can certainly customize a list of stopwords with domain knowledge, the metric termed as inverse document frequency is a computation which gives a general standardization measurement for a collection of documents. It works in the general case by accounting for the occurrence of most frequent terms in corpus by considering all the documents in which a requested term ever occurs [9].

1.3.1 Objectives of Natural Language Processing Toolkit

NLTK used for Natural Language Processing fulfils following primary objectives:

- **Simplicity:** To give a natural platform along with fundamental building blocks, providing users a practical awareness of NLP without getting bogged down in the boring maintenance usually connected with processing annotated language data. The NLP toolkit should structure the difficulties of structuring NLP systems, not cover them. Hence, every class is represented by the toolkit should be sufficient that a student could execute it by the time they finish preliminary course in computational linguistics.

- **Consistency:** NLTK provides consistency to give a sustained platform with reliable interfaces and data structures, and simply assumable method names.
- **Extensibility:** NLTK also provides extensibility to give a structure where new software components can be easily adjusted, including competing methods and alternative implementations to the same work. The toolkit should easily accept new software modules, whether those modules replicate the current functionality of the toolkit. This toolkit is structured in such a way that it is observable where latest extensions would accept by the infrastructure of the toolkit.
- **Modularity:** The toolkit provides modularity to give components those is used separately without need to learn entire toolkit. The communication among different components of the NLTK toolkit should be reserved to using easy, minimum, well-defined interfaces. Specifically, it should be feasible to complete individual projects using small parts of the toolkit is possible, without distressing about how they cooperate with the rest of the toolkit. This feature permits students to understand how to use the toolkit additionally during a course. To extend and change the toolkit is easier with modularity.

1.3.2 Operations Performed by NLP

Natural language Processing can perform lot of operations such as EOS detection, Tokenization, POS tagging, chunking, Extraction *etc*.

1.3.2.1 End-of-Sentence Detection

In this phase a text is broken in a group of valuable sentences. While sentences usually show analytical units of consideration, they incline to contain a composition that offers sound to advance analysis. For some different types of analysis the values of break down text into paragraphs or sections might be different, but it is improbable to help in the entire work of EOS detection.

1.3.2.2 Tokenization

Tokenization is a process where we splitting every particular sentence into tokens. Tokenization performed to do the same work as breaking on whitespace, with the exemption that it tokenized out EOS indicators accurately. As an informal note, some written languages, such as ones that use pictograms as incompatible to letters, do not

certainly even require whitespace to split the tokens in sentences and require the reader (or machine) to discriminate the boundaries.

1.3.2.3 Part-of-Speech (POS) Tagging

This step allocates POS information to all tokens. We do not spontaneously understand these tags, but they can easily illustrate POS information of tagging. For example, 'NNP' represent noun or noun phrases, 'JJ' represent an adjective and 'VBD' represent a verb which is in simple past tense. For example, with the use of POS tag, we will be capable to mass together nouns as part of noun phrases and then trying to cause regarding what kind of entities they might be (e.g., location, people, or institutes).

1.3.2.4 Chunking

In this phase all tagged tokens inside a sentence is analyzed and composite tokens which shows coherent concepts are collected. It is relatively a diverse method than finding collocations by statistics from these sentences. We can represent practice syntax through chunk of NLTK. In the next step NLTK reveals a function that collaborates chunking with named entity extraction.

1.3.2.5 Extraction

In this step we analyzing each and every chunk and then tagging them as named entities like people, places, organizations, *etc*. Some tokens are chunked together by it and also trying to categorize them to form specific types of entities.

1.4 Analyzing Bigrams

In the unstructured text processing there is an issue that is frequently overlooked as the huge quantity of information obtained when we are capable to consider large number of token at the same time, because numerous models we demonstrate are phrases [27]. For example, in a post most common terms are "friend", "forever", and "friendship" can we say that the text is most likely about "friend forever", "friendship forever", both, or neither? If we had a priori knowledge of the writer of the post or content of that post, then we could make a better assumption, but if we were depends entirely on a machine to attempt to categorize a document for collaboration software development, we required to again review the text to conclude which word generally

regularly appears after “forever” that is, we want to locate the collocations starting with the word “forever”.

An n-gram can be considered as a compact method of demonstrating every probable successive order of ‘n’ words of a sentence from a text record, and it gives underlying data organisation for calculating collocations. For every value of n, the number of possible bigrams is (n-1).We require a big sample of text for discover collocations, but expecting we had background auxiliary text, the next thing is to do the statistical analysis of the bigrams to decide which bigram corresponds to a collocation.

For the clustering of commonly co-occurring words, n-grams is a simple and powerful technique. If we evaluate all possible n-grams, we are probable to find that certain remarkable figures appear from the text [18].

1.5 Collocation Detection

A collocation is combination of randomly occurring and repetitive word or in simple words a sequence of words that occur together is called Collocation. Collocations can be used as indexing units, in addition to single-word tokens, since the frequency of a collocation is correlated with the importance of the collocation in the text. Collocation detection is one of many tasks in statistical NLP, and it involves finding interesting word combinations, collocations, in large corpora.

To determine the different meanings of a word or discover a suitable word to fulfil in the perspective, collocation knowledge can be very useful. So that collocation information is broadly used in various natural language processing applications, particularly for Word Sense Disambiguation, Information Retrieval, Machine Translation and Natural Language Generation. We specify collocation as a persistent and predictable appearance that holds syntactic and semantic associations. This is required that a collocation must consists more than two words in which at least two are content words.

Natural languages occupy a number of recurrent mixtures of frequent occurring words than assumable by collocations, chance and that correspond to random word usages. Various methods have been recommended to retrieve different types of collocations from the analysis of big set of textual data. These methods automatically generate large numbers of collocations through statistical figures calculated to reflect the importance of the links. However, none of these methods

gives a functional detail along with the collocation. Also, the results generated often included inappropriate word relations reflecting some specious aspect of the training a set of documents that did not arise for true collocations [10].

Collocations vary massively in the number of words included, in the syntactic group of the words, in the syntactic relations among the words and in how strictly the individual words are used together.

1.6 Document Similarity

It plays a vital job in various exploration tasks and document retrieval. Several techniques and methods have been introduced to define a graded sequence of similar documents to the document which is given by the user as a query. Documents classification according to contents of a given document is important to get necessary document efficiently. To attain good document classification similarity estimation is a key technique. In area information retrieval and knowledge discovery, text classification, document clustering, topic detection, topic tracking, questions generation, short answer scoring, machine translation, document summarization and identification of document similarity is a very important problem [12].

Classification is implemented based on the document similarity. In NLP "bag-of-words" model is used to mine features from documents and term appearance frequency based value is used as a score of every features. However, the term score approach usually uses predefined models and has some limitations. The score is calculated by "TF*IDF", where TF means "term frequency" and IDF means "inverse document frequency" [12].

1.7 Document Summarization

A summary is an abstract generated from a large document which contains a significant piece of information in the real document is no longer than half of the primary document. The tools and technology for summarizing a document is evolving and it provides a key for the problem of information overload [13].

Document summarization has concerned much consideration since the actual work by Luhn, which has established wide-ranging applications particularly with the explosion of documents on the web. Besides its major role of serving users to grab the important facts of a long document with less effort, it is also helpful as a pre-processing step for some text mining tasks such as document classification.

Luhn's algorithm is one of the best algorithms that can be used for finding summary of a document [14]. The first step of this algorithm is to select a sensible value for *N* and select the topmost '*N'* number of words for analysis. The hidden assumption which is considered in this algorithm is that the topmost *N* numbers of words are sufficiently expressive to differentiate the documents by their nature. The sentence which contains plenty of such words can become part of the abstract. The next step is to apply an empirical approach to every sentence and drag out certain divisions of sentences for being a part of the summary/abstract of a document. The algorithm uses a threshold distance metrics to collected words for scoring every sentence and scoring each collection. The absolute score for every sentence is equivalent to the maximum score obtained for several collections occurring in sentence of the document. A collection is represented as a words chain considering more than one term, where all significant terms lie in the range of scores obtained by distance threshold of its closest neighbor. The next step is to find out the sentence which can be a part of abstract. This is done by using two methods, the first one uses a numerical threshold to drag out sentences by calculation the mean value and standard deviation for the scores attained, and the second approach merely gives the top *N* sentences as a summary [15].

The important information in a text is often distributed throughout the different parts of the text. That is, some sentences contain more important information than others. Therefore, the key task in document summarization is to identify the key sentences in a text using an applicable method.

1.8 GooglePlus: Google's Social Network

Anyone with a Gmail account can create a Google+ account and start communicates with friends by posts, comments, likes, tagging *etc*. Google+ has evolved quickly and used some of the most powerful features of existing social websites such as Twitter, LinkedIn, Facebook *etc*., in carving out its own set of specific capabilities. The API documentation that's available online is always the exhaustive source of guidance, but a brief overview may be helpful to get thinking about how Google+ compares to another platform such as Twitter or Facebook. Various phrases used in Google+ are:

1.8.1 People

People are Google+ users. Programmatically, discover users by using the search API look them up by a personalized URL if they are a celebrity type or strip their Google+ IDs out of the URLs that appear in web browser and use them for exploring their profiles.

1.8.2 Activities

The things a Google+ user can do in their account are called activities. Basically, an activity is a note and can be as long or short as the writer likes: it can be as long as a blog post or it can be devoid of any real textual meaning. Given a Google+ user, we can easily get a list of that user's activities. Like a tweet, an activity includes a lot of interesting metadata, such as the number of times the activity or post has been re-shared.

1.8.3 Comments

Comment is the way of interaction with one another in every social media. Simple statistical analysis of comments on Google+ could be very interesting and potentially shows a lot of insights into a person's social circles. For example, which other Google+ users most frequently comment on activities? Which activities have the highest numbers of comments (and why)?

1.8.4 Moments

Moments are relatively recent modifications to Google+ and define a way of expressing interactions among a user and a Google+ application. Moments are equivalent to Facebook's social graph stories in which they are originate to capture and provide a chance for user interaction with an application that can be shown on a timeline. For example, if we were upload a picture, buy a product, or watch a YouTube video, it could be expressed as a moment (something we did in time) and presented in a history of our actions or shared with friends in an activity stream by the application.

The main focus is to collecting and analyzing Google+ activity data that is in textual form and deliberate to convey the similar kind of meaning that we might experienced in a tweet, blog post, or Facebook status update. In other words, we will be trying to analyze human language data. Google+ is mostly similar to Twitter in that

it gives a “following” model where we can add people to one of our Google+ circles and keep up with their happenings without any permissions on their side, but the Google+ platform also provides rich integration with other Google web traits, support a powerful property for videoconferencing with hangout, and has an API similar to Facebook’s in the way that users share posts and interact with each other.

1.9 Reflections on Analyzing Human Language Data

There are various reflections that may be helpful in synthesizing the process of analyzing human language data:

- **Context drives meaning:** Although TF-IDF is a robust technique which is easy to implement, but definite execution of it has some vital restrictions that we have appropriately ignored. The primary limitations is that it serves document by means of “collection of tokens”, that signifies the sequence of words in both the text document and the text given by the user as a query.

  In performing an analysis of n-grams to find the collocations present in the text and word ordering, we are still facing fundamental problem that TF-IDF understands that all words with the equal value of text denotes the similar object. A homonym is a word that has same spellings and pronunciations to different word but whose meaning are driven entirely by context.
- **Human language is overloaded with context:** There can be a set of unwanted data that have to be controlled in analyzing unorganized data, and this data is to be quite valuable for competitive executions. For example, comparisons between strings are sensitive to the case of alphabets, so that it is necessary to moderate words in a way to compute the frequencies as correctly as probable.
- **Parsing context from human language is not easy:** Another concern that is embedded lot in specific execution in comparison to universal feature of TF-IDF is that our using partition to divide the text into words may leave diminishing punctuation on various tokens which can influence organizing frequencies. Now in this case the trailing period on the token influences both the TF and the IDF computations. Something as apparently easy as a period indicating the end of a sentence is context that our brain processes trivially, but it is very difficult for a machine generate the same result.

Chapter 2 Literature Survey

2.1 Social Web Analysis

A social network analysis explores the framework of social interactions within a group to expose the informal relations among users. These interactions are often ones of communication, awareness, trust, and decision-making. Social web analysis believes that relationships are essential. As an approach to looking at these connections, social web analysis has been around a long time. Social web analysis is a technique with growing application in the online social network and has been applied in field as diverse as health, electronic communications psychology and business organization. In network theory, social networks are discussed in terms of ties and node. Nodes are users or members and ties are relationship within networks.

Mining and analyzing the social web has been an emerging area of research in the recent era mainly due to the enormous increase in the popularity and usage of such social networks. However, the vast amount of resources presented in these online Social Networks poses a big confront for researchers to analyze these social networks. Also, the data produced from these social networks is dynamic and requires intelligent mining to analyze this data. However, since data generated from these social networks is noisy, vast, dynamic and distributed, this requires suitable data mining methods to analyze such complex, large, and rapidly changing social data. Research is being carried out associated with various research issues in analyzing the social web such as, expert finding, influence propagation, link prediction, recommender systems, opinion mining, community detection, mood analysis, *etc*.

In social web analysis, the associations and interaction with one user with other users in the network is more significant than the features of users. To define various real-world phenomena, this method is useful but provides less space in favour of private enterprise, the capability of individuals, as most of it resides inside the network layout [20].

Social webs is used to inspect how online enterprises communicates with one another, connections and associations among individual users at different enterprises, as well as categorizing the many unstructured connections that relate subscribers together. For example, the extent to which an individual person within a network is at the centre of many connections than actual job title derives the power in an

organization. Contributions of Social web in hiring, in job performance and in business success are well known. Network provides many features for companies to collude in setting policies or prices collect data and deter competition.

M. Jamali and *H. Abolhassani* [21] reviewed social networks, formal techniques and properties of social web. For addressing many aspects of social structure Social web analysis techniques provides some useful tools. They had illustrated a particular link structure for Weblogs that includes only comments. The Semantic Web is a promising concept that gives a proposal of having data on Internet defined and connected in a way that it can be used by users and executed by machines.

2.2 Text Mining

Before understanding the area of text mining or information retrieval one must get some fundamental idea of vector space models. It can be described as the procedure of analyzing text to extract appropriate information that is required for specific purposes. Evaluated with the kind of data stored in databases, text is informal and complex to deal with. For text mining there is no need to understand the text from the document context to extract necessary information [22].

V. Gupta and G. S. Lehal [23] concluded that, Text mining can also be considered as KDT (Knowledge-Discovery from Text) or Text Data Mining. It denotes the procedure of fetching information that is required as well as important from informal document. Text mining is a current integrative area that represents on machine learning, statistics linguistics, information retrieval and data mining *etc*. As almost all information is collected as text, text mining is supposed to possess a lofty commercial prospective worth. Various sources of information and unstructured texts stay readily as source of knowledge.

J. H. Kroeze et al. [24] investigated intelligent text mining with real text data mining. The main principle involved in text mining is to find or make new facts using a corpus. These facts can be the innovation of latest unknown patterns in known data or it may integrate artificial intelligence capabilities to understand the patterns and offer more advanced abilities such as hypotheses idea. Natural language processing reproduces individual potential required for intelligent text mining. The need of natural language processing to assist intelligent text mining should be researched advance. The parameters of novel analysis, non-novel, and semi-novel were used to

distinguish among standard text mining, intelligent text mining and full-text information retrieval, and. The same factors were also used to distinguish between associated procedure for text metadata and numerical data. These differences may be used as a road map in the developing areas of information and data retrieval, knowledge detection and the formation of new facts.

D. Sanchez et al. [25] introduced the design of TKM (text knowledge mining) as a specific event of knowledge mining. Obtaining previously unknown, potentially useful and non-trivial information from information repositories are the main purpose of TKM. Researcher considers the term "knowledge mining" was used by many researchers in different fields, though they do not include this design in the present literature. After this proposal many new possibilities in the field of text mining came into existence. The application of non-inductive inference methods extend to different kind of text mining applications that can take benefits of the extensive existing literature in natural language processing, knowledge demonstration, and reasoning. In this paper they have briefly described some existing text mining methods that, in their belief, can be considered as TKM techniques.

2.3 Information Retrieval

The information retrieval system searches for information or data contained in a corpus by the user's query. Searches are based on full-text indexing or metadata. With the fast growth of present IT, Web has become the huge and most extensively used system in the world that can gives a large environment or interface for knowledge management and information sharing. Internet users can extract required information any time and from anywhere with information retrieval tools and method. Though the conventional keyword-based search techniques disregard the semantic information included in the key words, it has lower recall factor and pertinence factor and cannot accommodate to the demand of information retrieval.

Information retrieval is used in the large datasets to make a small summary of those large documents set, when a client fires a query to ask for the required information on the web or on their personal system.

Natural Language Toolkit (NLTK) provides list of stopwords that can be used to filter words such *as a, and, is* and *the*, except some words which can avoid almost all stopwords lists but still they are pretty frequent. The inverse document frequency is an estimation that offers a general normalizing metric for a collection of documents

even when we can modify a list of stopwords with domain knowledge. It gives a higher value if a word is repeated many times in a set of documents [26].

In order to build context based semantically information retrieval system, both semantic as well as context information was used by *T. Cioara et al.* [27]. They used three types of information retrieval system: information retrieval, constraint context information and user context information.

Z. Mei [28] reviewed the main modules of the grid information retrieval system, and the main methodologies are analyzed in depth, an easy and practical model of the grid information retrieval system is intended and executed. The experiment defines that the system recognizes the information retrieval function based on semantic grid which can understand the accurate semantic matching according to the specific context of the concept. The retrieval recall factor and pertinency factor can be enhanced significantly, and the query request of the users can be fulfilled.

L. Shuang and H. Zhu [29] presented that distributed information retrieval methodology for information retrieval research that builds the retrieval effectiveness greatly. It can also efficiently resolve problems of interaction among distributed servers. Protection of data or information is obvious in case of Distributed technology. A number of database retrieval points can be attained by the amount of information and search effectiveness.

N. J. Belkin [30] presented an analysis of information retrieval as an information-seeking activity, following interactions among people with text. From their analysis of the situation which guide to the information retrieval situation, they found that information retrieval is most appropriately considered as a form of information-seeking behaviour, in which the interaction between users with text is the central phenomenon, to which the information retrieval system must respond, and which the information retrieval system must support. The clear consequences of this view are the motive of the information retrieval system is to maintain the user in entire range of information seeking behaviours; the user must be considered the central component of the information retrieval system. From this follows, that the role of the demonstration and comparison processes in information retrieval are in support of communication, and, that control of the information retrieval interaction must be mixed among the participants.

Y. Huang et al. [31] introduced rough ontology into semantic information retrieval to fulfil user need to extreme amount. Firstly, semantic information retrieval

is defined and its advantages are analyzed in detail. Secondly, rough ontology is employed to extend precise ontology. Thirdly, rough arrangement based the semantic information retrieval system and its semantic similarity computational method is given by them. Finally, a rough ontology based semantic information retrieval system called as ROSRS is designed. The experimental outputs illustrate that system ROSRS can retrieval information semantically not only from defined ontology but also from rough ontology, and the implicit information can be gotten. Moreover, the recall ratio and correctness ratio of retrieval outputs can be enhanced.

2.4 Inverse Document Frequency or Term Frequency

The Term frequency for a particular word can be defined as a number of times that word present in the content document upon total words present in that document.

IDF can be computed by considering the presence of identical words in a corpus of documents by considering total count of documents present in the corpus in which a term given as query always appears. The perception beyond calculating IDF is that it generates the greater score if a word is quite rare across the collection of documents than if that word is frequent, which helps in solving the problem occurring with various stopwords because they are commonly present in each document. Mathematically, the one and only gradation of attention for the IDF computation is that a logarithm function is required to diminish the coming solution into a condensed scope, since its standard use is in multiplying IDF with TF for scaling purposes

TF-IDF is one of the most common approaches for information retrieval from a set of text documents. Using the qualified value of words or terms in the document, user can query a collection of documents. Mathematically,

$$tf\_idf = tf * idf,$$

Where *tf* shows significance of any particular document, and *idf* shows significance of a word or term relative to the whole set of document. Product of these gives a score that combines both factors.

TF-IDF is a very common weighting technique used to define documents in the Vector Space Model, particularly in Information Retrieval problems. TF-IDF combines two quantities: the term frequency (TF) the number of documents in which a specified word appears and the inverse document frequency. It gives more attention to rare terms (Hapaxes) than to common words. Combining TF and IDF, therefore, increases the score of the common terms in a document, while at the same time

increasing the score of rare terms in the collection. IDF reduces as the number of documents that contain a specified word or term increases.TF-IDF is used to query a collection of documents by computing normalized scores that demonstrate the comparative value of terms in the documents. The tf-idf value is rises in accordance to the times a word occurs in the document, but is balanced by the frequency of the word in the document, which is useful to adjust for the detail that some terms appear more repeatedly in general [33].

N. Oren [34] aimed to examine the functionality of genetic programming to the formation of *tf.idf like* document surveyors. These surveyors can then be used to produce document demonstration vectors, which are in turn consumed in similarity computations to create a document list ordered by supposed significance to a request of a user. This method proposed the opportunity of constructing well-tuned information retrieval tools for particular corpus, as well as the invention of surveyor superior to fundamental *tf.idf* while used to common collections. Discovering an enhanced document survey or permitted one to control currently available higher level methods such as significance feedback to further advance the quality of the information retrieval system. Firstly the extra documents are added to the corpus after training has completed. Secondly, training is done on the entire dataset, with implementation taking place on a dissimilar dataset. The first expansion to their work was to replicate the situation where a document collection housing associated documents is customized as time progresses. While the documents are relevant, the quality of the surveyor is expected to fall among the results described, and those contains when applying this technique to produce general purpose surveyors. The second extension they were done is to consider the ability of this method to function as a general purpose surveyor.

W. Zhang et al. [35] used an English and Chinese corpus to correspondingly examine the three approaches in information retrieval and text categorization. From past results he concluded that LSI performs better than other approaches of document collections. Their multi-word extraction is an easy method. They were validated whether or not an enhancement in multiword extraction would generate an enhancement in indexing using multi-word. They discussed two distinct features related to text illustration as word weighting and index word selection carefully. Individual effects on demonstration were different from each other.

2.5 Vector Space Model

A document in vector space model is defined as a vector. In the vector space model, the relevance of a document to a request is represented by the mathematical similarity of their corresponding term vectors. In Vector space model the distance between any two vectors represents similarity of the corresponding documents by providing a large multidimensional space. In this model a query is defined as a vector to discover similar documents. Documents with shorter vector distance are most appropriate. Cosine similarity is used to measure similarity between vectors. The similarity among two vectors is also not intrinsic in the model [36]. Cosine of the angle between vectors is used to measure the similarity among the documents represented by vectors. The cosine value for orthogonal vectors is 0.0 and 1.0 for overlapping vectors. Similarity measure is calculated by the dot-product of two vectors. If the cosine value of the angle between two vectors is equal to their dot product then all the vectors must be of unit length. A vector can logically be defined as a line segment drawn among the origin and a point in an *N*-dimensional space.

Below in Figure 2.1, an example of a document vector plotted in 3-dimensional space is shown.

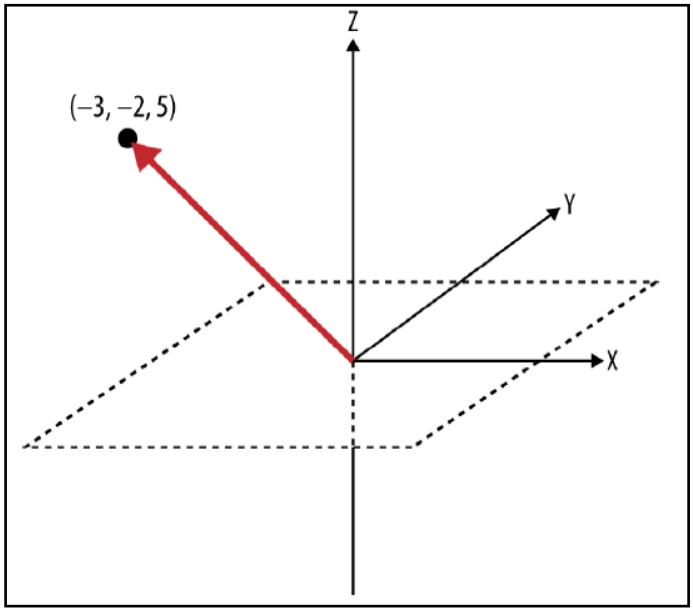

Figure 2.1 A document vector plotted in 3D space

The fundamental theory behind a vector space model is that it provides a large multidimensional space that includes one vector for each document, and the distance among any two vectors specifies the similarity of the equivalent documents..

L. Liu et al. [37] proposed an approach for measuring the similarity of words using Pattern vector space model. They purposed that similar words fundamental idea is that a similar word tends occur in similar context. They used TF*IDF, related pattern vectors for all word and cosine similarity measure for two different pattern vectors. And, Chinese Miller-Charles data set produced an output that tells that approach better than other two baseline models. This testing shows that the process is influenced by training data obviously.

A. Wibowo et al. [38] proposed the use of WordNet and Indonesian vocabulary to construct relations among terms and also described a way to give relation scores among these terms. They also proposed the use of dictionary to link relations among terms from different languages.

X. Xiaoping and Y. Junhu [39] improved the classical Vector Space Model (VSM), with solved the problem of vocabulary by centre vector. Centre vector can be visually defined as a vocabulary dictionary. By means of looking up this dictionary, the most appropriate vocabulary can be originated. It overcomes the limitations of the previous model, enhances recall rate and accuracy rate and makes the compatibility of vector space model better. The improved vector space model also has some dissatisfaction.

2.6 Cosine Similarity

In Cosine similarity metric the similarity among two vectors by computing the angle among them is measured. For documents, the cosine similarity measure is used to find the angle between two document-term vectors. The angle between these vectors shows how similar these documents are. The range of values for this measure is between 0 and 1, where 0 means that the two vectors are orthogonal and 1 means that the two vectors point to the same direction. Considering two vectors a and b, the cosine similarity between them is represented as:

$$\cos(a, b) = \frac{a.b}{\|a\|.\|b\|}$$

Where, ||a|| is the Euclidean norm of the vector a.

Cosine similarity is the general techniques to differentiate the documents in a corpus with each other, which is the fundamental nature of discovering a similar document from a given data set. It calculates similarity among two vectors that computes the cosine of the angle among them. The cosine value of the angle among two vectors thus defines whether two vectors are pointing in approximately the similar direction [40].

The cosine of the angle among any two vectors is a suitable metric for distinguish them and is called the cosine similarity of these vectors. The cosine similarity of documents shown as term vector is a very efficient metric. Similarity among two vectors is measured by the cosine of the angle among any two vectors and it is similar to the dot product of their unit vectors. Following equations demonstrate this concept:

$\overrightarrow{\boldsymbol{doc1}} . \overrightarrow{\boldsymbol{doc2}} = \|\boldsymbol{doc1}\|.\|\boldsymbol{doc2}\| \cos \boldsymbol{\theta}$	**Given (by trigonometry)**
$\frac{\overrightarrow{\boldsymbol{doc1}} . \overrightarrow{\boldsymbol{doc2}}}{\|\boldsymbol{doc1}\|\|\boldsymbol{doc2}\|} = \cos \boldsymbol{\theta}$	**By division**
$\hat{\boldsymbol{doc1}} . \hat{\boldsymbol{doc2}} = \cos \boldsymbol{\theta}$	**By definition of "unit vector"**
$\hat{\boldsymbol{doc1}} . \hat{\boldsymbol{doc2}} = \boldsymbol{similarity}(\boldsymbol{doc1}, \boldsymbol{doc2})$	**By substitution**

S. Tata and J. M. Patel [41] discussed the issue of approximating the selectivity of cosine similarity predicates. They examined why calculating the selectivity of cosine similarity predicates is a much complex problem, and designed a solution based on suspicious experimental interpretations regarding the allocation of the dot product of distinctive queries. They represented that the method is space capable (small size summaries) and time efficient. They also described that their method has logically better exactness in practice.

S. S. Nyein [42] proposed an algorithm that extracts the key data from the web pages. They used an algorithm that supports CST (Content Structure Tree). The developed algorithm first employ HTML Parser to create Document Object Model (DOM) tree that is further used to create Content Structure Tree (CST).Content Structure Tree (CST) simply divides the key content chunks apart from the other chunks. Then the algorithm establishes cosine similarity to predict which branch of the CST tree demonstrates the less significant and which branch illustrate the more significant of the page. Their algorithm describes the similarity values, ranking of documents and picks up the highest ranked documents as among the most appropriate to the request. The appropriate documents can be extracted from the web using cosine similarity metric and developed algorithm that employs CST.

M. Rezvani and S. M. Hashemi [43] introduced a new framework based on web graph and content similarity is existed in order to get better the correctness of PageRank. This framework is executed using Jaccard index and cosine similarity measures, and as a result of their experimental analysis, they shall show that putting page content similarity in action increases the correctness of web ranking in some user ranking algorithms. In addition, time complexity and implementation problems are discussed to get a practical result.

Y. Dong et al. [44] established and designed an algorithm for originality recognition in real time by using real valued negative selection approach. They focussed on developing an effective algorithm that is inspired by immunology for monitoring various systems available online. The method used by them completes the computation in real time and domain. The important benefit of this algorithm is how to consume the metric of cosine similarity.

L. Muflikhah and B. Baharudin [45] reviewed that the document clustering can be applied with the use of concept cosine similarity and space. It had made the important reduction of term-document matrix dimension with demote to *k*-rank. Also their common presentation is very high with f-measure regarding 0.91 and entropy regarding 0.51. It is an important enhancement when applied in massive data amount until more than 50% increasing.

2.7 Document Summarization

Document summarization is the procedure of diminishing a text document with an algorithm in order to produce a summary that keeps the most significant points of the original document. A better document summary should consist of the only those information which are very important, from a document in a form consistent to the user of the information. Being capable to execute practically better sentence recognition as portion of an NLP method for mining formless data may allow certain great text mining abilities, like simple but very logical attempts at document abstraction. Document summarization has concerned much concentration since the original work by Luhn, which has originated wide-ranging applications particularly with the detonation of documents on the Web.

Luhn's document summarization algorithm can be used for finding out summary of the various posts, which describes that the significant sentences that are present in a text document are which include commonly appearing terms. All frequently occurring words are not important such as stop words. First all the stop words must be filtered out. The next task in this approach is to select a sensible value for *n* and then n topmost words are selected for further analysis [46]. After that most "significant terms" in the document is identified to use a heuristic to every single sentence and drag out certain subdivision of various sentences that can be used an abstract. To score every sentence, one distance threshold metric is used to cluster various words and assigning scores to each obtained cluster corresponding to the succeeding expression:

$$\frac{\boldsymbol{important\ words\ present\ in\ cluster^2}}{\boldsymbol{number\ of\ words\ in\ a\ cluster}}$$

The absolute score for every sentence is equivalent to the maximum score obtained for several collections occurring in sentence of the document. A collection is represented as a words chain considering more than one term, where all significant terms lie in the range of scores obtained by distance threshold of its closest neighbor. The next step is to find out the sentence which can be a part of abstract. This is done by using two methods, the first one uses a numerical threshold to drag out sentences by calculation the mean value and standard deviation for the scores attained, and the second approach merely gives the top *N* sentences as a summary.

2.8 Entity Centric Analysis

One explanation is being proficient to identify the entities within corpus and then for the analysis those entities are used, as opposite to doing document-centric analysis including searching for various keywords or explaining a search input as a special kind of entity and customize the outputs correspondingly. While we may not have idea regarding it in those terms, this is accurately what promising expertise. Apart from of the internal method that is used to complete this end, the resultant user knowledge is considerably very dominant because the outputs be conventional to a layout that more closely convince the user's requirements.

By using NLP pipeline it is easily pull out all the phrases and noun from the given document (Google+ posts) and indicating them as entities occurring in the documents. The significant fundamental supposition is being that phrases and nouns succeeded as entities of interest. This is actually a reasonable guesses to make and a fine initial point for entity-centric analysis.

M. Pennacchiotti and P. Pantel [47] generalized the combining of sources and properties in an environment called Ensemble Semantics. They defined very big improvements in entity extraction by merging state-of-the-art distributional and pattern based systems with a great set of properties from a webcrawl, Wikipedia, and requested query logs. Experimental results on a web scale extraction of actors, sportspersons and musicians illustrate expressively higher mean average accuracy scores equated with the present state of the art.

W. Wang et al. [49] studied the problem of dictionary-based entity extraction with edit distance constraints in this paper. By capturing small faults that are possibly to be missed by existing methods it can increase the recall of the system. The matching problem is practically difficult as current methods based on q-gram filtering have low performance because of the presence of numerous small entities in the dictionary. Their designed solution is based on an enhancement neighbourhood production filtering technique. They had successfully diminish the size of the neighbourhood that necessary to be generated and indexed from $O(m^{\tau})$ to $O(lp^{\tau} 2)$. In addition, they proposed an effective query processing algorithm that evades examining query section and entity sets that are not in the final matching results. They also improved the algorithm to share calculation and avoid unwanted variant record.

Chapter 3 Research Problem

3.1 Problem Statement

Social media has become very popular communication tool among internet users in the recent years. Humans have an ingrained tendency to share their ideas, experiences and knowledge, which associate them with the rest of the world, so that they can be recognized and can also identify their importance and worth. They are eager to know about happenings around them, that is why they communicate in order to share their ideas, observations and queries. Social media is one such communication medium that made people to be heard and satisfy their curiosity to know about rest of the world. The rapidly escalating coverage of the social media user-bases and user engagement provide the social media an immense potential to study the logic under the users social behaviour and organization, and finally to transfer this knowledge to real time business profit.

Social media can be considered as one of the important sources of real time human language data. A large unstructured data is available for analysis on the social web. The data available in the form of text on these sites contains various HTML tags, lot of stopwords which are not so important for analysis and lot of redundancies as users are free to enter the data according to their knowledge and interest. This data needs to be pre-processed before doing any analysis to make the data suitable for analysis.Analyzing this human language data presents in huge social datasets plays an important role in business and academics. By mining this social data, a lot of useful information about our social contacts can be analyzed and visualized, which cannot be discovered manually. For example, which users are interacting with each other? , what they posts on their profile? , is there any relevant information present in their post? , Searching for the similar posts, about whom a particular user is talking about *etc*., are some of the useful information that can be analyze by mining the social data and this extracted knowledge can be used in various decision making purposes.

In social networking sites, in addition to friendship relationships, individuals can have lot of family ties with other individuals, join several social groups, be associated with a variety of organizations, or belong to diverse geographical sub-networks. By considering the complex composition of these social networks rather than restraining the analysis to a single relationship type or entity, richer models has been build that endow with better indulgent of the dynamics in these websites and

thus help in providing extremely accurate predictions about upcoming events. These days, the main focus of analysts is to observe the interactions between various users of social networking sites and analyze the human language data generated from their interaction. To study and analyze compactness, centrality and grouping with the promise of security and privacy of the users data in these networks is the main research area in social web analytics. Various research issues associated with social web analytics are predicting future trends, community detection, link prediction, recommender system, frequency analysis, mood analysis, retrieval of document of interest, computing similarities between posts, Finding summary of posts, opinion mining, influence propagation *etc*. Use of efficient data mining techniques, machine learning approaches, text mining and information retrieval fundamentals, various statistical tools and network analysis concepts made the social data analysis, a challenging and more effective goal.

3.2 Research Gaps

A lot of research is being carried out in the area of social web analytics. The social networks are created and knockdown by studying daily basis and continuous communication among users. These studies incorporate with individual's relationships with others in terms of different parameters of human behaviour. In this way, these networks are proven an important and critical tool to study the communal phenomenon, composition, relationships and behaviours of individuals and/ or group. The pre-processing steps such as removing HTML markup and various stopwords present in human language data are carried out to make the social data set suitable for analysis. A lot of researches are going in the area of social web analysis and text mining but there is a lack of effective research to analyze the social behaviour of human data, by applying text mining and information retrieval techniques on huge amount of human language data available on social web.

There are some special web platforms such as Google+, which allow us to post unlimited text but there is lack of effective research work has been carried out in order to find out a precise abstract of large post which will help us to analyze gist of the post without going through the whole post. In case of similar posts of trending topics posted by single user or multiple users, those posts can be clustered by finding the similarity score between all posts.

A lot of research work is going on in the area of document centric analysis but there is limited work that is done in field of entity centric analysis of social posts made by the user. The posts can be analyzed to extract entities present in the posts. This will help us in discovering "To whom posts are addressing?" The interactions between various entities can also be analyzed so as to find out "How Entities are communicating?" Our research work focuses on Document centric as well as Entity centric analysis of Google+ data.

3.3 Research Objectives

In the light of above discussed research gaps following objectives have been formulated.

- To study various research issues associated with social web analysis and text mining techniques for analyzing the social data.
- To perform document centric analysis such as finding most frequent words, group similar posts in a cluster, extracting posts according to our interests, finding abstract/summary of the post and finding out the various collocations.
- To perform entity centric analysis by extracting various entities present in a Google+ post and analyzing the interaction between those entities.

3.4 Research Methodology

In this research work python language is used, which is an excellent scripting language for manipulating text. The dataset required for analysis will be extracted using Google+ API. Google+ uses OAuth 2.0 authorization mechanism to provide its data for development purposes. Python "google-api-python-client" package and authorization credentials (OAuth 2.0) obtained by registering for an application on Google+ development environment, will be used to explore the Google+ API users and their activities. The "clean_html" function used *i.e.* provided by Python library to remove various HTML markup present in the text and then a Python package "BeautifulSoup" is used to convert html tags into plain text. The python package "matplotlib" is used to plot the graph generated as a frequency distribution of words according to Zipf's law. For finding out Similarity between various posts, each document is represented as a vector and "cosine similarity" metric is then used to measure the similarity between two posts. To make a summary of an entire post Luhn's algorithm is used. There are two approaches that are applied to get the

summary; one uses a statistical threshold to find the sentences by calculating the mean and standard deviation for the scores and another is simply returns the top N sentences. The metric "Jaccard Index" is used to find out various collocations present in the text by analyzing each possible bigram. To analyze social dataset the python NLTK module is used for text analytics.

Chapter 4 Data Extraction, Pre-processing and Analysis

In this section Google+ activity data is analyzed *i.e.* stored in the form of text and proposed to express the same kind of sense that might see in posts of Google. In other words, to analyze human language data that is stored on the web in the unstructured from. Anyone that has Gmail account can easily create a Google+ account and starts freely communicate with their friends. Google+ has grows rapidly and used some of the most effective properties of existing social media such as Twitter, LinkedIn and Facebook in represent its own set of specific capabilities and characteristics. Google+ has emerged tried-and-true properties of existing social media, such as maintaining users profile according to personalized user privacy settings and marking the content for posts with hashtags, with additional creativity such as a new approach of content sharing called *circles*, video chats called *hangouts*, and huge integration with other Google facilities for instance Gmail contacts.

While exploring Google+ the important thing take in mind is that it has a specific set of characteristics and is a quite difficult to compare directly to other social network features. It is similar to Twitter in that it provides a "following" strategy where we add particular user to one of user's Google+ circles and keep up with their activities without their permission, but the Google+ platform also provides strong integration with other Google web features, supports a powerful property for videoconferencing (hangouts), and has an API similar to Facebook's in the way that users share their posts and other content and communicate.

Using Google+ API, the interaction among users is formed in terms activities, comments, people and moments.

Table 4.1Phrases used in Goggle+ with their description

Phrases	Description
People	People are the various users on Google+.
Activities	Activities are the various things that people do on Google+.
Comments	The users on Google+ interact with each other by leaving comments.
Moments	Moments represents a way of capturing interactions between Google+ Application and a user.

4.1 Implementation Methodology

The flow chart of the implementation methodology is described well in Figure 4.1. In the first step is to get the Goole+ API is from the Google's developer page and then obtains the dataset from the particular Goole+ API in the JSON file. In the next step entire HTML tags are removed such as
tag and escaped HTML from the activity contents with the help of *cleanHtml* function. In the next step for the better analysis we will ignore the various stopwords with the help of Natural Language processing Toolkit (NLTK).Then this dataset is analyzed by performing frequency analysis to find out number of sentences, tokens, unique words, Hapaxes, most frequent words *etc*. Also the frequency of word is graphed with their corresponding rank according to Zipf's law.

After frequency analysis, the next phase is to retrieve the relevant data from the document by calculating score for each document using TF-IDF. Now we will use the Cosine Similarity Metric for finding the similarity among various posts or documents. After that similarity between various documents are computed by representing each document as a vector in N dimensional space and using cosine similarity metric. Then similarity between documents is visualized with a matrix diagram representing the adjacency matrix, where every cell 'ij' illustrate an edge from vertex *i* to vertex *j*. Here, vertices signify documents in the corpus, while edges represent similarity in the corpus.

Then collocation detection and analyzing bigrams has been done. Collocations are extracted from all posts. Collocation can be defined as a sequence of words that regularly occurs together. In the next Step the analysis is based on entity Centric to identify the entities present in given document and to utilize those entities for further analysis. Finally, summary of the various posts is generated using Luhn's document summarization algorithm. At last various entities from each post are extracted and interactions between various entities present in the posts are analyzed.

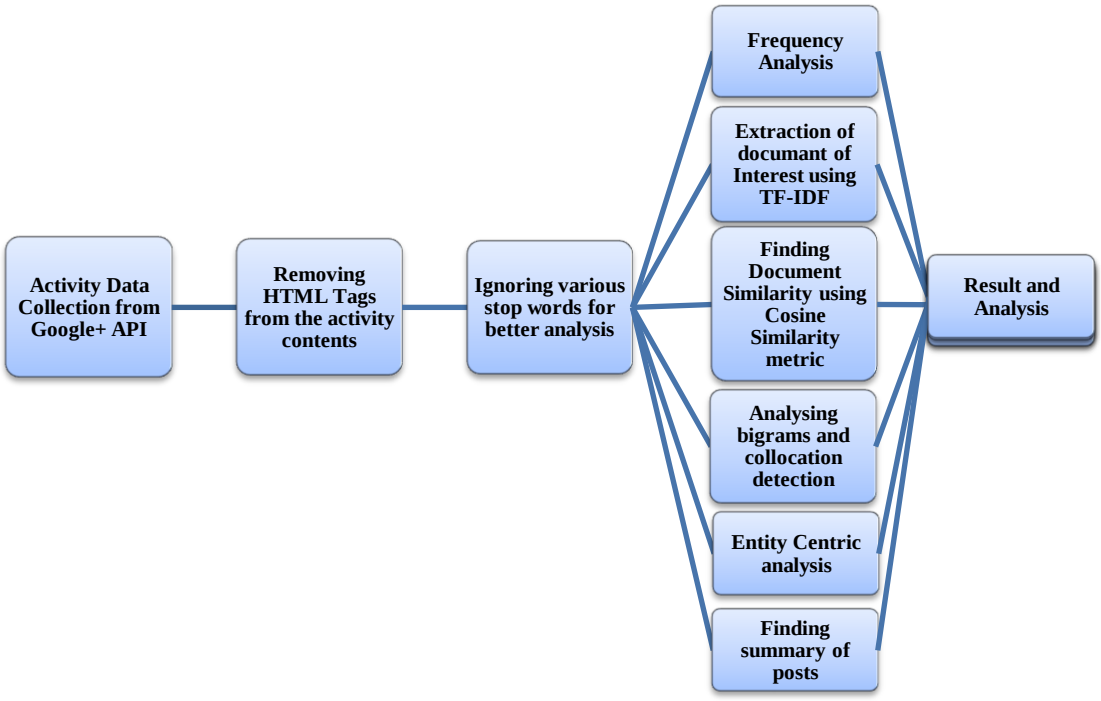

Figure 4.1 Flowchart of Methodology

Various packages in python are available for analysis and visualization of Google+ data. It is easy to download and install this packages using pip. Table 4.1 illustrates various python packages used in this implantation phase and their functions.

Table 4.2Python Packages used for Analysis of Google+

Python packages	Description
google-api-python-client	Python wrapper that wraps the Google API.
httplib2	A comprehensive HTTP client library.
Json	Encode and decode tweets into JSON format.
Math	It provides access to the mathematical functions defined by the C standard.
clean_html	Clean the html Tags.
BeautifulSoup	Convert the plain text into Html format.
Nltk	Python package used for Natural Language Processing.
Numpy	Array processing for strings, numbers, objects and records.
Display	Various display related classes are present in this module.
IFrame	Demonstrate the use of iframes and nbviewer to embed IPython notebooks in WordPress.
Matplotlib	library which produces publication quality figures
Operator	Provide functions corresponds to intrinsic operators

4.2 Making Google+ API Request

The Google API Console gives a means of registering an application to get OAuth credentials but also provides an API key that is used for "simple API access". This API key is used to programmatically access the Google+ platform and just about all other Google service. Figure 4.3 display various authorization keys which are accessed after filling the application form at Google+ developer platform.

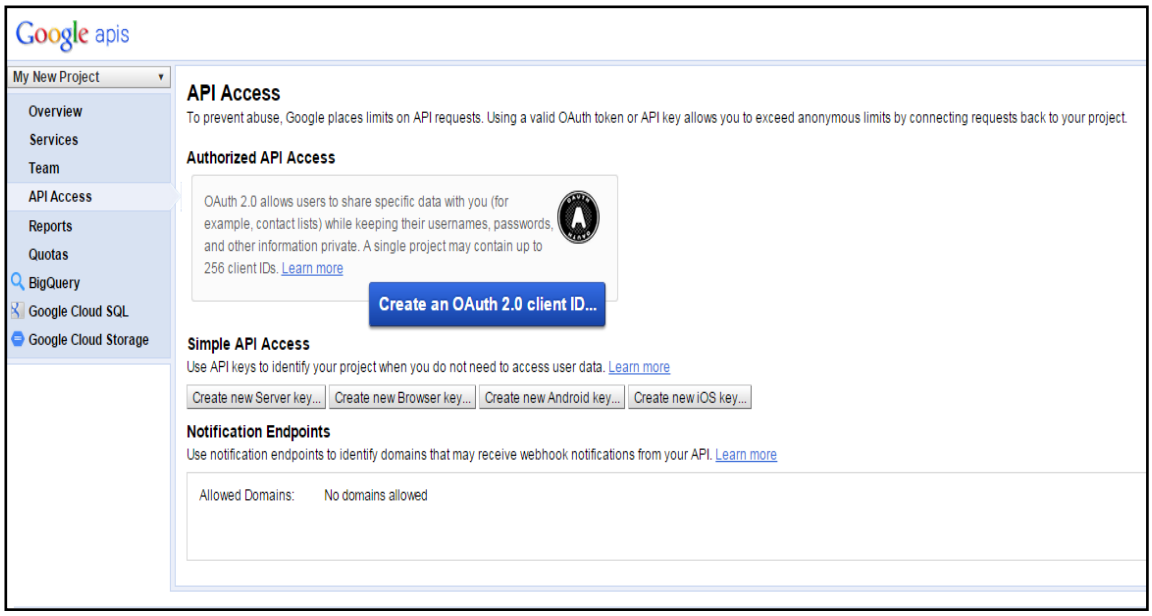

Figure 4.2 Getting OAuth from Google API console

After creating an application it is also need to particularly enable it to use Google+ as a deferent step. Figure 4.4 illustrate a snapshot of the Google+ API Console. Python package called google-api-python-client is installed to access Google's API by installing install google-api-python-client through pip.

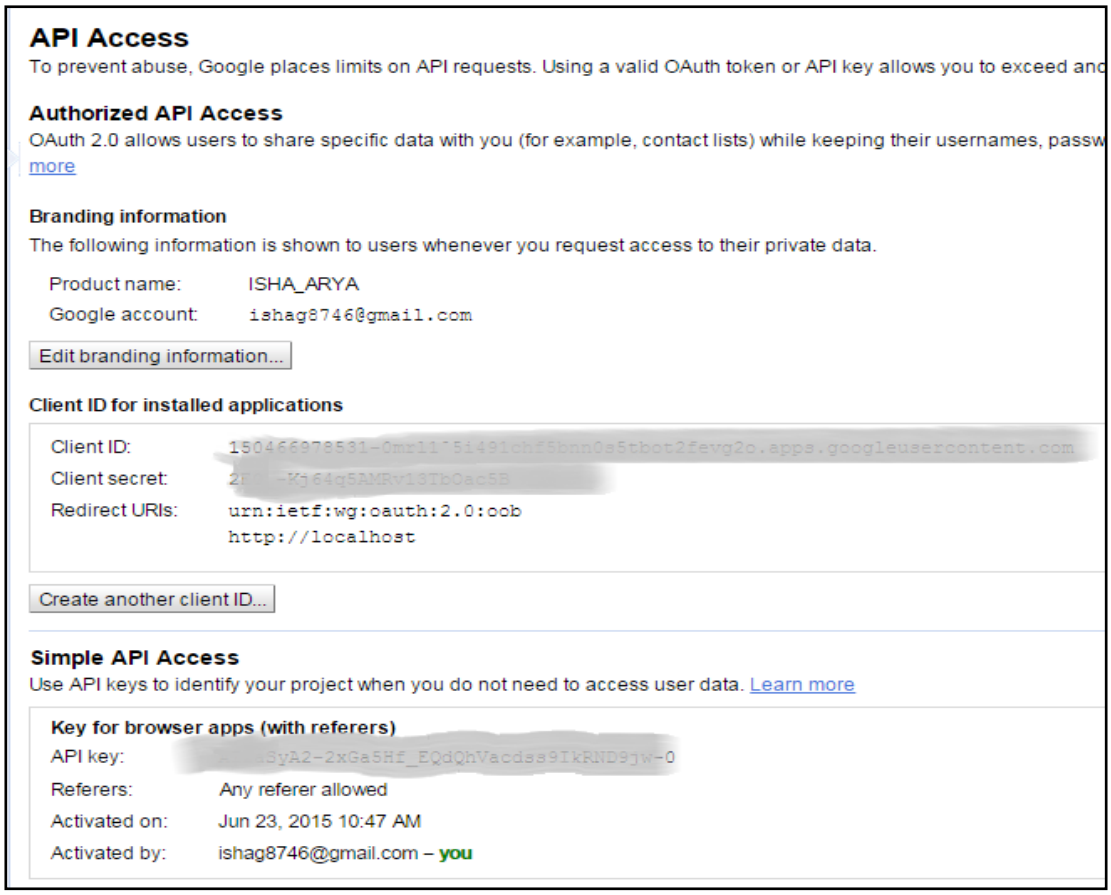

Figure 4.3 Registering an Application with the Google API Console

It is necessary to enable Google+ API access as one of the service options as shown in Figure 4.5.

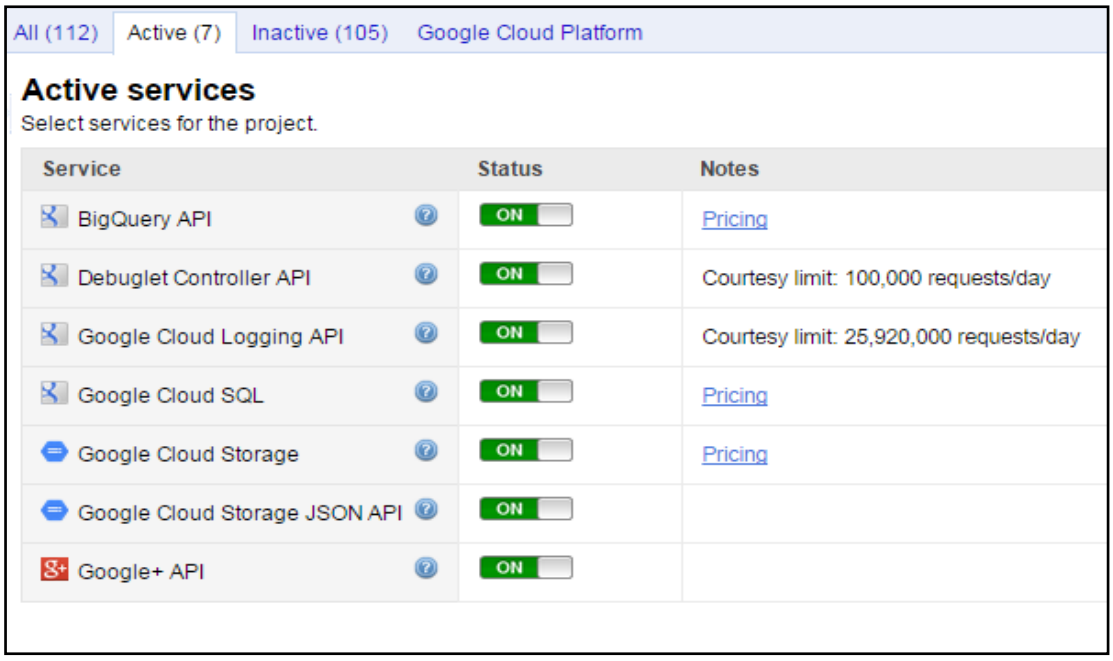

Figure 4.4 Enabling Google+ API Access as one of the Service Options

After collecting all the information, this data is downloaded in to a file so that any other unnecessary API requests can be avoided. The downloaded file of all the connections information is a saved in JSON format.

4.3 Searching for a Person with Google+ API

Google+ API has been used to Search the persons with their name. Figure 4.6 demonstrates with the Google+ API how to search for a person. Let make a search query with the person name "Jot Kiran". The basic pattern that is repeatedly use with the Python client is to create an instance of a service *i.e.* parameterized for Google+ access with API key that can then instantiate for particular platform services. To create a connection to the People API service.people () function has been used and then chaining on some additional API operations concluded from reviewing the API documentation online. Code snippet for searching a person is described below:

```
Q = "Jot Kiran"

API_Key = ""

Service=apiclient.discovery.build('plus', 'v1',http=httplib.Http(),
                                  developerKey=API_KEY)

People_feed= service.people().search(query=Q).execute()
```

In this code "Jot Kiran" is a query an API key of a Particular user is given. Here information about peoples is extrcted.

```
[
 {
  "kind": "plus#person",
  "displayName": "Jot Kiran Dhillon",
  "url": "https://plus.google.com/116890775105140943928",
  "image": {
   "url": "https://lh3.googleusercontent.com/-vFJnQ5aLFV8/AAAAAAAAAA
I/AAAAAAAAACo/4XyXd2PucU8/photo.jpg?sz=50"
  },
  "etag": "\"RqKWnRU4WW46-6W3rWhLR9iFZQM/ImLjGgXHuYffzzMezmiBqGkeE4c
\"",
  "id": "116890775105140943928",
  "objectType": "person"
 },
 {
  "kind": "plus#person",
  "displayName": "Jot Kiran",
  "url": "https://plus.google.com/118038718855399879706",
  "image": {
   "url": "https://lh5.googleusercontent.com/-36qQcDFOidU/AAAAAAAAAA
I/AAAAAAAAACk/g9EQHMQE6Us/photo.jpg?sz=50"
  },
  "etag": "\"RqKWnRU4WW46-6W3rWhLR9iFZQM/5UyIhI5LECxthYQKBxd7eldeMHU
\"",
  "id": "118038718855399879706",
  "objectType": "person"
 },
 {
  "kind": "plus#person",
  "displayName": "kiran jot",
  "url": "https://plus.google.com/114165162113521978112",
  "image": {
   "url": "https://lh3.googleusercontent.com/-A1n5wPBqmtA/AAAAAAAAAA
I/AAAAAAAAASA/BMfBKGGrqSU/photo.jpg?sz=50"
  },
  "etag": "\"RqKWnRU4WW46-6W3rWhLR9iFZQM/qjm7-3U2r8zsCKuaOwdWrjD2aNc
\"",
  "id": "114165162113521978112",
  "objectType": "person"
 },
```


Figure 4.5 Shows Results obtained after Searching for "Jot Kiran"

The results gives required list of people named Jot Kiran, but in this result this is difficult to say which one of these persons refers to the well-known "Jot Kiran" that the user is looking for. There are two ways for getting the exact result one is to activity information or request profile for each of these results and try to find them manually one by one. The other way is to provide the avatars considered in each of the results, which is trivial to do by providing the avatars as images. Figure 4.7 demonstrate how to display avatars and the corresponding ID values for each search result by generating HTML and providing it inline as a result in the notebook. Python code for removing HTML tags:

```
html=[ ]
for p in people_feed['items']:html += ['<p><img src="%s" /> %s: %s</p>' %
                                        \(p['image']['url'],p['id'],
                                        p['displayName'])]
HTML(''.join(html))
```

In this code the data is stored in the HTML format . It shows the result with the person name and ID as shown in Figure4.7.

Figure 4.6 Google+ Avatars as Images

4.4 Extracting Google+ Activity Data for Particular User

The fetching some activities from the profile of a Google + user required to adjust the design pattern for searching for people. The fetched data contains the URL of posts and the content of post. The code for fetching Google+ Activity Data is described below:

```
USER_ID = '101485833467132579095'

activity_feed = service.activities ( ).list( userId=USER_ID, collection= 'public',
                                             maxResults = '100' ).execute ( )
```

Sample results for the first item in the results are shown in Figure 4.8 and illustrate the basic nature of a Google+ activity.


```
{
 "items": [
  {
   "kind": "plus#activity",
   "provider": {
    "title": "Google+"
   },
   "title": "If you have two friends in your lifeti
me, you're lucky. If you have one good friend, you'
re more than...",
   "url": "https://plus.google.com/1014858334671325
79095/posts/DRyWavnmfwG",
   "object": {
    "resharers": {
     "totalItems": 0,
     "selfLink": "https://www.googleapis.com/plus/v
1/activities/z12tvhq5blitxde2e232ilkjfljty3vry04/pe
ople/resharers"
    },
    "url": "https://plus.google.com/101485833467132
579095/posts/DRyWavnmfwG",
    "content": "If you have two friends in your lif
etime, you&#39;re lucky. If you have one good frien
d, you&#39;re more than lucky<br />Friendship is li
ke money, easier made than kept.\ufeff",
    "plusoners": {
     "totalItems": 0,
     "selfLink": "https://www.googleapis.com/plus/v
1/activities/z12tvhq5blitxde2e232ilkjfljty3vry04/pe
ople/plusoners"
    },
    "replies": {
     "totalItems": 0,
     "selfLink": "https://www.googleapis.com/plus/v
1/activities/z12tvhq5blitxde2e232ilkjfljty3vry04/co
mments"
    },
    "objectType": "note"
```


Figure 4.7 Fetching Recent Activities for a Particular Google+ User

The entire activity object follows a three-tuple pattern of the form *(actor, verb, and object)*. In this post, the tuples (Jot Kiran) tells us that this particular item in the results is a *note*, which is basically just a status update with some textual content. A closer look at the result disclose that the content is something that Jot Kiran feels strongly about as indicated by the title "if you have two friends in your lifetime" and hints that the note is active as indication by the number of re-shares and comments.

4.5 Data Pre-Processing

In the output the content field for the activity considers HTML markup, as evidenced by the HTML entity I've that occurs. In general, an assumption has been considered the textual data rendered as Google+ activities obtain some basic markup, such as
 tags and escaped HTML entities for apostrophes so as a best practice. Figure 4.9 provides an example of how to refine plain text from the content field of a note by introducing a function called "cleanHtml". It takes advantage of a clean_html function provided by NLTK and another handy package for manipulating HTML, called BeautifulSoup, which converts HTML entities back to plain text. It has the ability to process HTML in a reasonable way even if it is invalid and violates standards or other reasonable expectations. This package has been installed via *pip install nltk beautifulsoup4*. For removing the HTML tags following code/algorithm is used:

```
def cleanHtml(html):
  if html == " ": return " "
  return BeautifulStoneSoup(clean_html(html),
      convertEntities=BeautifulStoneSoup.HTML_ENTITIES).contents[0]
print activity_feed[ 'items' ][0][ 'object' ][ 'content' ]
print cleanHtml(activity_feed[ 'items' ][0][ 'object' ][ 'content' ])
```

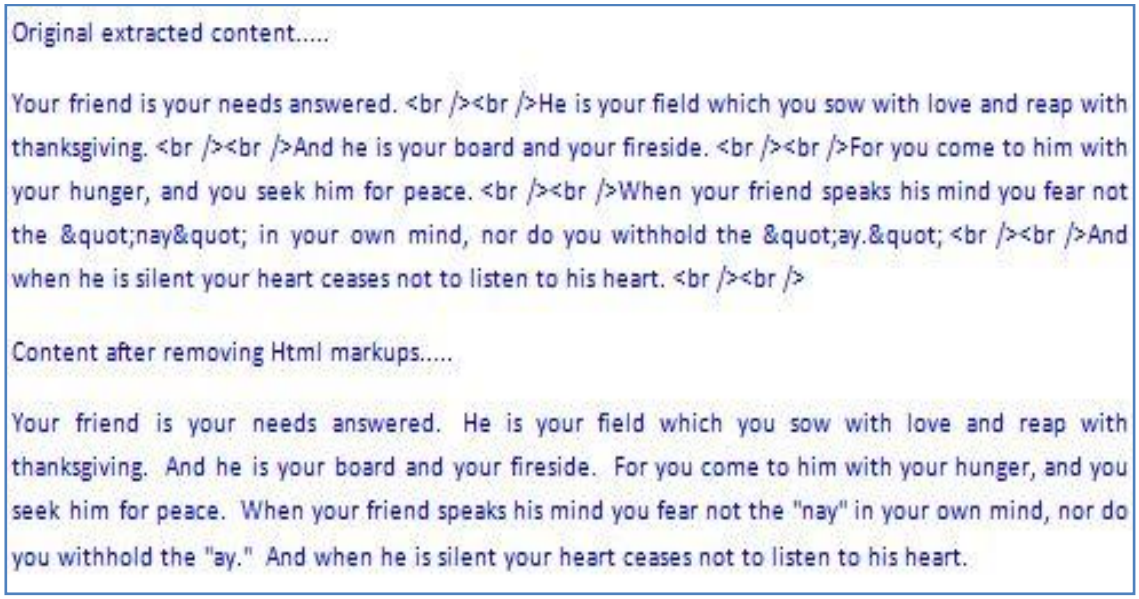

Figure 4.8 Removing HTML Tags from a Google+ post

The ability to query out clean text from Google+ is the basis for the remainder of the text mining exercises in this section, but one additional consideration that has been find useful before focusing the attention elsewhere is a pattern for fetching multiple pages of content.

4.6 Document Centric Analysis

In document centric analysis, the analysis is done on the posts at a document level. Each post is considered as a document and perform frequency analysis, querying post dataset with TF-IDF to extract relevant post, measuring similarity between posts and finding summary of each post, collocation detection *etc*.

4.6.1 Frequency Analysis

Virtually all analysis constitutes the simple work out of counting things on several levels and Google+ dataset is manipulated so that it can be counted and further manipulated in some meaningful ways. Among them more convincing causes for mining and analyzing Google+ activity data is to attempt to respond the query of what humans are discussing about. The simple approach we could apply to reply this query is fundamental frequency analysis.

4.6.2 Querying Google+ Data with TF-IDF

Here by Term Frequency -Inverse Document frequency (TF-IDF) on Google+ activity data that is fetched previously and observe that how TF-IDF has been utilized for

query the Google+ activity data. To make certain queries and qualitatively evaluate the obtained results that how well the TF-IDF metric works. An exact value of the scores is not relevant as it is the capability to sort and discover documents by significance.

4.6.3 Finding Similar Documents

Earlier documents of user's curiosity were found on query, and then the next thing to accomplish is to discover similar posts. Whereas the scores given by TF-IDF can give the means to extract relevant document from a collection on the basis of query given by users, cosine similarity is a popular procedure for comparing the documents to each other, which is the main purpose of discovering similarity between documents. Learning cosine similarity requires a terse knowledge to the vector space models.

4.6.4 Visualizing Document Similarity with Matrix Diagram

For visualizing similar documents, graph-like structures has been used, in which a link among different documents represents similarity among the documents. This type of visualization provides tremendous prospect to introduce more visualization from D3, the state-of-the-art visualization toolkit. D3 is particularly planned with the concerns of data analytics, suggests a common declarative composition, and get a nice centre ground among low and high level interfaces.

A nominal variation is matrix diagrams in which a set of edges and nodes are represented that can be used to make visualizations. A nested loop can calculate the similarity among the documents in the test corpus of Google+ data, and connection among items may be defined based on an easier statistical thresholding measures.

A plus point of a matrix diagram versus a graph-based layout is that there is no prospective for disorganized overlap among edges that shows the connections, so avoid the common "hairball" problem. However, the ordering of rows and columns concern to intuition about the patterns that may be present in the matrix, so careful thought should be given to the best ordering for the rows and columns. Usually, rows and columns have additional features that could be used to order them such that it is simpler to identify patterns in the data.

4.6.5 Collocation Detection

A collocation is an arbitrary and repetitive word combination or in simple words a sequence of words that occur together is called Collocation. Collocations can be used as indexing units, in addition to single-word tokens, since the frequency of a collocation is correlated with the importance of the collocation in the text. Collocation detection is one of many tasks in statistical NLP, and it involves finding interesting word combinations, collocations, in large corpora.

For the clustering of commonly co-occurring words n-grams is a simple and powerful technique. If we evaluate each possible n-gram, we are probable to find that a number of exciting figures appears from document itself with no extra effort needed. N-gram can be considered as a concise method of demonstrating every probable successive order of n tokens from a text of the given document, and it gives the basic data structure for calculating collocations. Here the *Jaccard Index* is used as the scoring function. Contingency table is used to rank the repeated appearance of terms in certain bigram (2-gram) as match with the probabilities of various other tokens those can also be present in that bigram. The Jaccard Index computes similarity among sets, and in this instance, the sample sets are particular comparisons of bigrams that occurred in the text.

4.6.6 Computing Summary of a Post

Summary is an abstract of a document which provides an overview of entire document. To summarize user's Google+ Posts by using Luhn's algorithm for document summarization. The fundamental principle behinds the Luhn's approach is that the relevant sentences in a document are those which include repeatedly appearing terms. The first step of this algorithm is to select a sensible value for *N* and select the top most '*N'* number of words for analysis. The hidden assumption which is considered in this algorithm is that the topmost *N* number of words is sufficiently expressive to differentiate the documents by their nature. The sentence which contains plenty of such words can become part of the abstract. The next step is to apply an empirical approach to every sentence and drag out certain divisions of sentences for being a part of the summary/abstract of a document. The algorithm uses a threshold distance metrics to collected words for scoring every sentence and scoring each collection. The absolute score for every sentence is equivalent to the maximum score

obtained for several collections occurring in sentence of the document. A collection is represented as a words chain considering more than one term, where all significant terms lie in the range of scores obtained by distance threshold of its closest neighbor. The next step is to find out the sentence which can be a part of abstract. This is done by using two methods, the first one uses a numerical threshold to drag out sentences by calculation the mean value and standard deviation for the scores attained, and the second approach merely gives the top *N* sentences as a summary.

4.7 Entity Centric Analysis

This analysis is focuses on identifying various entities present in a Google+ post and then analyze various interactions made by those entities, as opposed to doing document-centric analysis which involves searching by keyword or understanding an input given as a search query, as a specific kind of entity and modifying the outcomes correspondingly. Apart from the internal method that is needed to complete this end, the resulting user knowledge is considerably more influential because the outcomes match to a layout that more closely fulfil the prospects of user.

4.7.1 Extraction of Entities Present in Google+ Posts

Here POS tags are analyzed that are practise to the tokens and identify nouns and phrases as the entities. When talking about data mining, discovering the entities in a text data is known as entity extraction or named entity recognition, depending on the nuances of precisely what we are trying to accomplish. Whereas a list of extracted entities likes the preceding one lends itself to being scanned quickly for patterns. Case is definitely a vital characteristic of the text, but there are some stimulating entities present in the given document in lowercase.

Although the list of entities absolutely does not express the entire significance of the text as efficiently as the abstract determined previously, recognising such entities can be tremendously meaningful for scrutiny while they possess a well-defined semantic meaning and like the words which are frequently occurs in the text. In fact, even the low frequency words are essential because they contains some fundamental meaning such as they are locations, ideas, people, objects *etc.*, which are commonly the all-important information present in the data.

4.7.2 Analyzing Interaction between Entities

Here various entities from the Google+ posts are analyzed and collect the relevant information and then the interaction between these entities is discovered. Entities are collected on the basis of per-sentence that is pretty useful for calculating the connections among entities using a sentence as a context window.

A small amount of distortion in the outputs is approximately predictable, but obtaining outputs that are very comprehensible and helpful even if results include a convenient quantity of distraction is a precious objective. The quantity of exertion necessary to get perfect outcomes that are almost free from all the noises can be enormous**.** In most of the conditions, this is absolutely impractical due to the intrinsic difficulty considered in human language and the restraints of most presently accessible toolkits, including NLTK. If we are proficient to create definite guesses regarding data domain or have proficient awareness of the nature of the present noise, we may be proficient to formulate heuristics that are effectual without risking an improper quantity of information loss but it is a quietly complex proposition.

5.1 Analyzing Frequency of words and their Rank

Here the frequency of words from the Google+ posts is analyzed using Zip's Law of natural language. It emphasize that frequency of a word within a set of posts is inversely proportional to its rank in the frequency table and it gives some thumb rules that can be helpful in determine frequency distribution for words within a set of documents. After cleaning the extracted documents by removing HTML markups and removing various stopwords present in the documents, the text of each document is tokenized into words and frequency of each word is calculated, and a frequency distribution is built on the basis of frequency of words present in the activity text and their rank as shown in Figure5.1.

<FreqDist with 104 samples and 137 outcomes>

Rank	Tokens	Frequency
0	word	3
1	believe	3
2	admission	2
3	better	2
4	waters	2
5	soul	2
6	know	2
7	awkwardly	1
8	enemy	1
9	needed	1
10	swirling	1
11	forbid	1
12	deal	1

Figure 5.1 Frequency of words present in activity text and their rank

Then frequency of every word present in each activity and corresponding rank of that word computed according to Zipf's law is plotted as a graph as shown in Figure 5.2, which hugs both the axis closely. This graph represents that higher frequency words lies on the top most rank in the frequency table. When the same graph is plotted on a log-log scale, it resembles somewhat to a straight line with negative slope as shown in Figure 5.3.

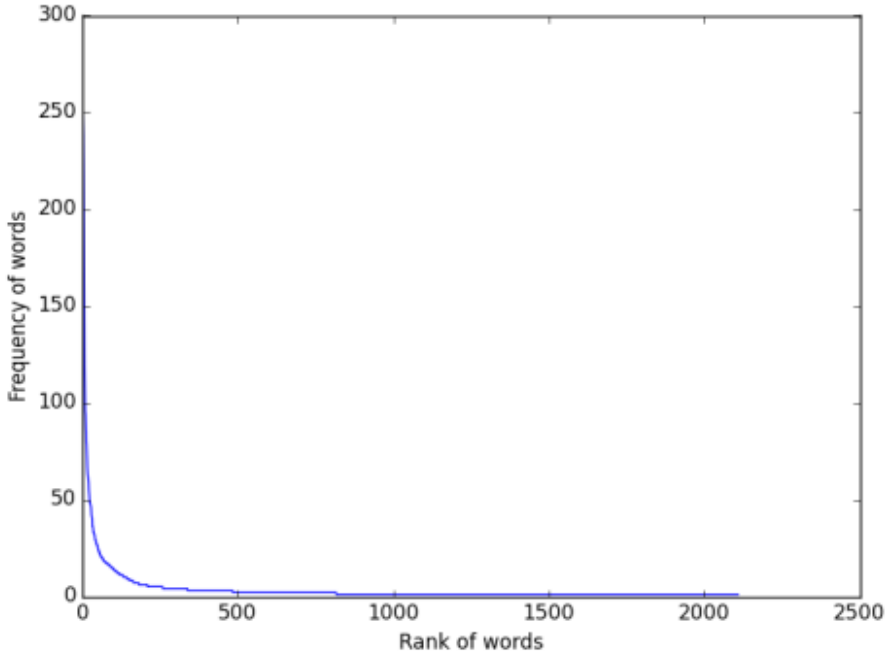

Figure 5.2 The Frequency Distribution for Terms appearing in Google+ Data.

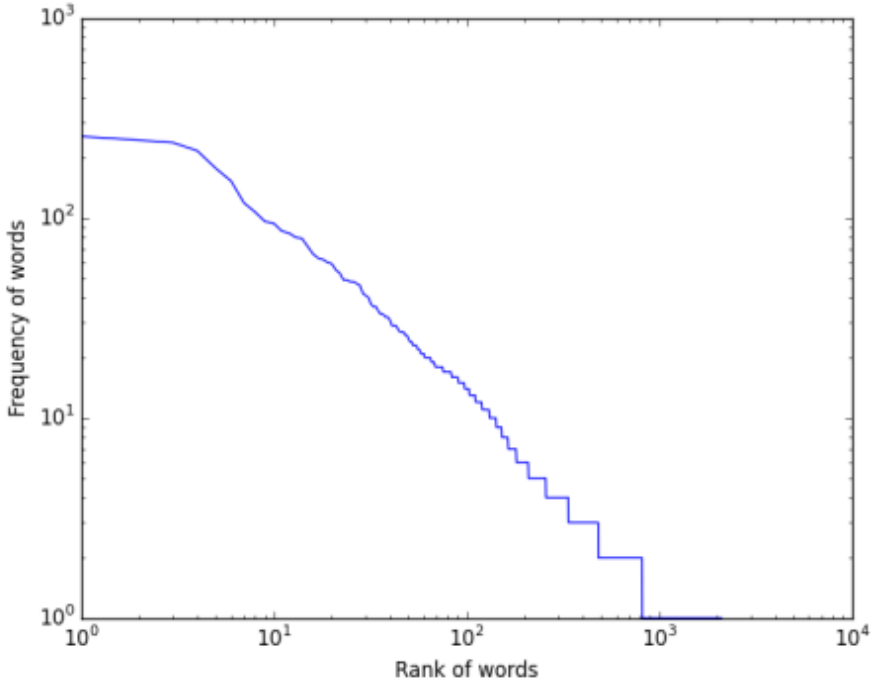

Figure 5.3 The Frequency Distribution Plotted on Log-Log Scale.

After that, the extraction of some useful information from each activity present in the datasets such as total number of sentences, tokens, unique words, Hapaxes, most frequent words *etc*. as shown in Figure 5.4 is done. Using sentences tokenizer

and then a word tokenizer, firstly the text can be break down into sentences and then every sentence into tokens. After extracting tokens, then find out number of unique words, number of hapaxes and most frequent words in each document.

```
"If you're alone, I'll be your shadow. If
 you want to cry, I'll be your shoulder.
If you want a hug,...
        Num Sentences:          6
        Num Words:              65
        Num Unique Words:       30
        Num Hapaxes:            17
        Top 10 Most Frequent Words (sans
stop words):
                ... (1)
                want (2)
                need (2)
                'll (5)
                smile (1)
                friend (1)
                happy (1)
                `` (2)
                hug (1)
                alone (1)
```

Figure 5.4 Frequency Analysis of a Google + activity

5.2 Querying Activities with TF-IDF to extract Document of Interest

Now TF-IDF is applied to the Google+ activity data and its importance in querying the document datasets is analyzed. By passing multiple query terms TF-IDF is used to score the documents by relevance. For getting appropriate documents from a collection of documents TF-IDF is one of very powerful technique.TF express the importance of a term in a particular document like the entire number of times it occurs in a document of a corpus, however it is usually the case that has been normalize it with computing the whole terms in the document, thus on the whole score used for document length comparative to a frequency of term. IDF shows the importance of a term in the entire collection of documents. The score comes after multiplying both tf_idf, used for both factor and be an integral portion of every main search engine. Based on the TF-IDF score the most relevant document based on the given query can be extracted as shown in Figure 5.5. It is the ability to explore and sort documents by user's requirement. The Figure shows the top 3 scorer documents according to a mutual query "Friend, friendship" from the corpus of activity data.

Figure 5.5 Retrieval of Document of Interest based on TF-IDF

5.3 Finding Similarity between Posts

While TF-IDF can gives the means to small down a large set of documents based on search terms or user's query, cosine similarity is the general method for find the comparison among documents with each other, that is the nature of discovery a similar document within a corpus. Now this cosine similarity metric is used to find out the similar activity documents after representing each document as a vector in N dimensional space based on the TF-IDF of each word present in the document. The cosine of angle among any two different vectors is a suitable metric for measuring similarity of the vectors. Calculating the cosine similarity between documents illustrated like term vectors is a very well-organized metric. In this a term vector is generated for every document and then calculates the dot product or called the inner product of unit vectors for given documents. It can be included that as closer the two vectors are to each other, the smaller the angle among those documents and thus the greater the cosine of the angle among the document. On the bases of score obtained from cosine similarity metric the most similar document to all the documents can be retrieved as shown in Figure 5.6.

```
Most similar to Your friend is your needs answered.

He is your field which you sow with love and reap wi
th thanksgiving... (https://plus.google.com/10148583
3467132579095/posts/KVShJwmjeZ8)
        True friendship is never serene.
Oooh I get by with a little help from my friends.
A friend is someone... (https://plus.google.com/1014
858334671325790 95/posts/YkvfkhbyFjE)
        score 0.585406
```

Figure 5.6 Clustering Posts by Similarity Computation using Cosine Similarity

Now the similarity between every document presents in a corpus can be visualized as a matrix diagram showing an adjacency matrix, where each cell *'ij'* describes an edge from vertex *i* to vertex *j* as shown in Figure 5.8. Here, vertices describe documents in the corpus, while edges define similarity in a corpus.

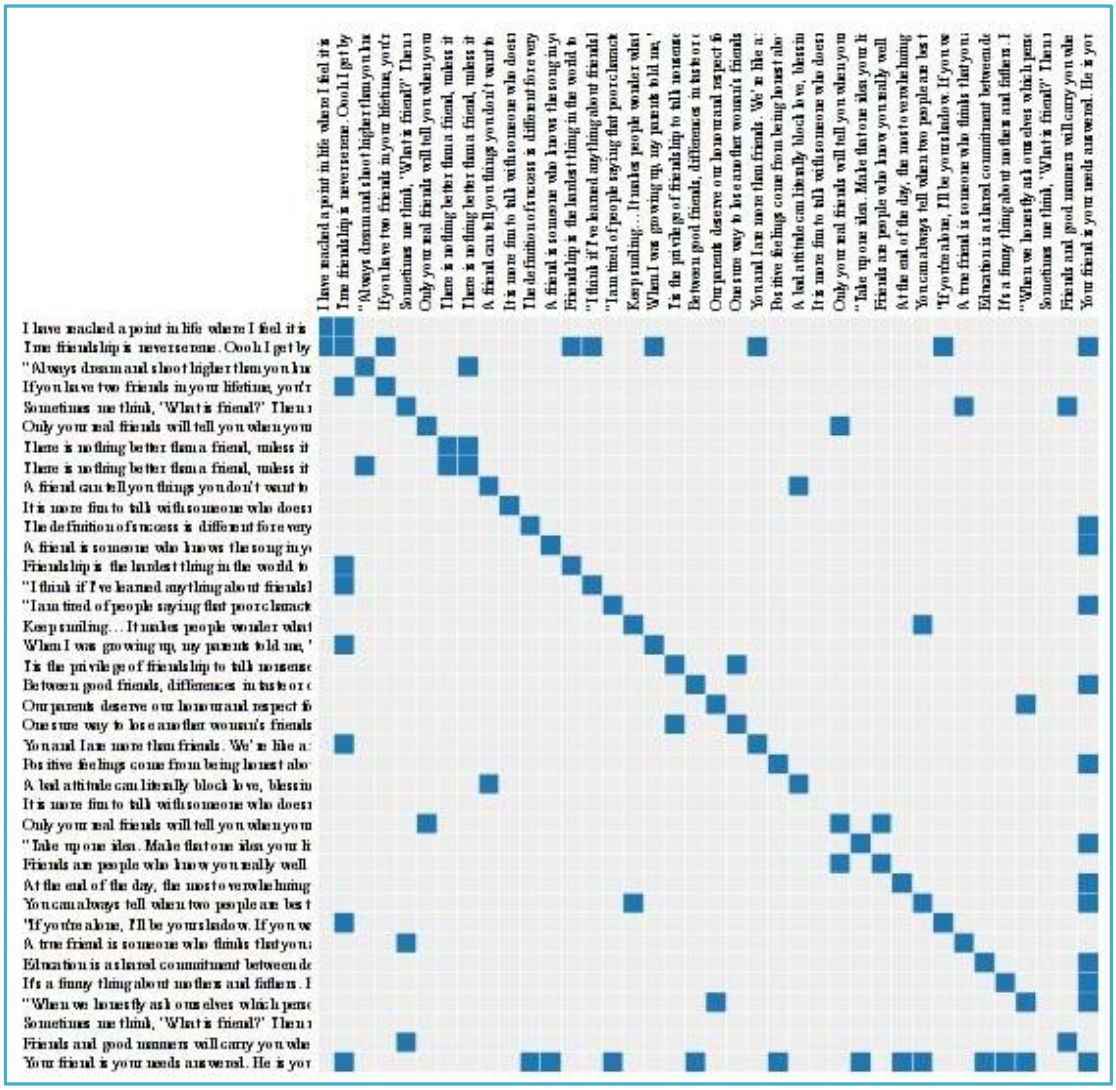

Figure 5.7 Visualizing Document Similarities with a Matrix Diagram

5.4 Collocation Detection

For calculating collocations n-grams provides the foundational data structure, *n*-gram is an approach of representing all likely consecutive chain of *n* tokens from a text. N-grams are very simple most fundamental technique for clustering commonly co-appearing words. NLTK is used for computing collocations in the text document. The Jaccard Index is the fraction that computes the similarity among set of documents. This is described like the frequency of a specific collocation is splitting by the whole number of collocations which includes minimum of one term in the collocation of user's query. It is helpful for identifying the possibility of whether the specified terms really appear as collocation, and levelling the possibility of likely collocations. Here, various collocations present in the activity data are found by analyzing all possible number of bigrams and searching for frequent bigrams among them with the ignorance of various stop words present in the dataset. Figure 5.8shows the various collocations obtained from user's Google+ activity dataset.

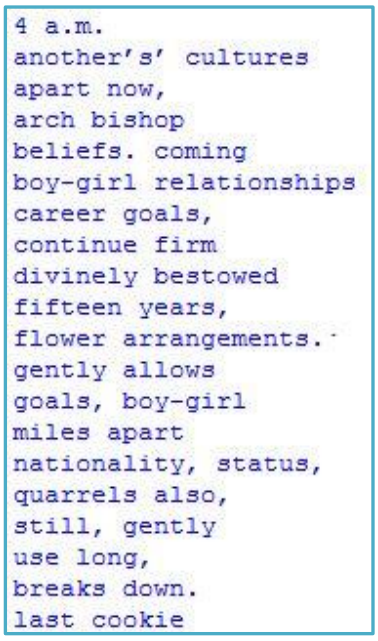

Figure 5.8 Collocations present in Google+ Activity Data

5.5 Finding Summary/Abstract of a Google+ Post

Luhn's algorithm is used to summarize documents. This algorithm used to summarize the Google+ posts. Summary is obtained by using two approaches one is average score simply gives the top *N* sentences and another is uses a statistical threshold called

Standard deviation to drag out sentences by calculating the mean and standard deviation for the outcomes acquired. With this algorithm the possible expectation is that the top *N* words are adequately expressive to distinguish the behaviour of the given document, and so as to any two sentences present in the document; the sentence that obtains more of these words will be included more expressive. For all sentence the final score is equal to the greatest score for at all cluster occurring in a sentence of a given document. A cluster is described as a chain of words including two or more significant words, where every significant word is inside a space threshold of its adjacent neighbour. Now summary of each activity is found by using Luhn's algorithm as shown in Figure 5.9.

Figure 5.9 Top-N Summary and Mean Scored Summary of a Post

An another possible way of visualizing the outputof document summarization by showing the full text of the post with the sentencesthat are included as part of the summary in bold so that it is easy to see what is included in the summary and what is not from the entire post. The output genrated as a summary in a collection of HTML files that it can be viewed within the browser without the use of a web server.The result is the entire text of the document with the sentences producing the summary are highlighted in bold as shown in Figure 5.10.

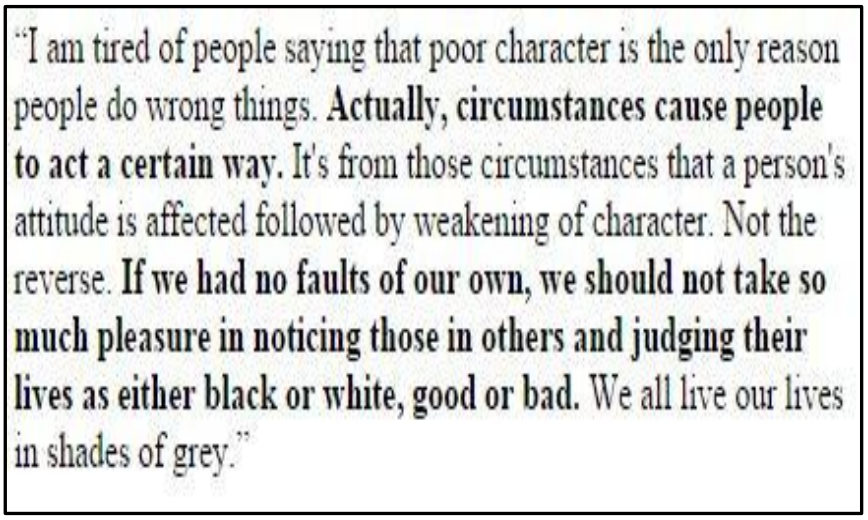
"I am tired of people saying that poor character is the only reason people do wrong things. **Actually, circumstances cause people to act a certain way.** It's from those circumstances that a person's attitude is affected followed by weakening of character. Not the reverse. **If we had no faults of our own, we should not take so much pleasure in noticing those in others and judging their lives as either black or white, good or bad.** We all live our lives in shades of grey."

Figure 5.10 HTML output showing a Post and its Summary

5.6 Entity Centric Analysis

Here it detects the various entities from document of interest and using those entities as the base of analysis with understanding a explore input as a specific kind of entity and designing the outputs appropriately.

5.6.1 Entity Extraction

Entire nouns and phrases can be extracted from the posts of a user of Google+ and then index these nouns and phrases as entities occurring in posts shown in Figure 5.11. The list of entities does not providing the actual sense of the passage of posts as efficiently as the summary calculated, determining those entities is very important on the basis of analysis while their sense is at a semantic level and are not the commonly appearing terms.

```
Tom said: Oh, there you are. Ralph relpied: Hi. Did you get the classes you want
ed? Tony asked: Not ...
--------------------------------------------------------------------------------
------------------------
        Mumbai (1)
        Raj Did (1)
        Nisha (1)
        Pause ) Coffee (1)
        Thursday (1)
        Gee (1)
        Ralph (1)
        Zimbabwe (1)
        Hi (1)
        Rose (1)
        Uh (1)
        Aman (1)
        Tuesday (1)
        Pick (1)
        Sita (1)
        Tony (1)
        Might (1)
        Haryana .Meena (1)
        Ram (1)
        Ajay (1)
        Did (1)
        Prices (1)
        Laughs ) Want (1)
        Friday (1)
        Meenu (1)
        Well (1)
        Physics (1)
        Laughs (1)
        Laughs ) (1)
        Gujrat (1)
        Rina (1)
        Oh (1)
        Biology (1)
        Friday (1)
        Tom (1)
```

Figure 5.11 Entities Extracted from a Google+ Post

5.6.2 Analyzing Interaction between Entities

The extracted entities of Google+ posts as shown in Figure 5.11, the next objective is to analyze the interaction between these entities. For analyzing Interaction among these entities, it considers the verbs then calculate triples of the form -verb-subject-object so that it can be identified which entity interacts with which another entity in a post, and the scenery of the communications between entities as shown in Figure 5.12.

```
Physics; lectures; morning; Tuesday

na need; alarm



Rose; Friday; classes; spoil
(; Laughs ) Want
mansi; anil; u; anil answerd; Mumbai; masni




thousand; students



na; lot









Meenu; Haryana .Meena; dollar

money; year
jot; Well; bucks
Might; one-I 'll


Nisha; half
(; Laughs ); Laughs

shortage; dorm space; year.rashmi
```

Figure 5.12 Analyzing Interactions between Entities

When the analysis of the interaction among entities t it also deal with a definite amount of noise in output are approximately expected, but obtaining results that are extremely understandable and valuable even though they do include reasonable amount of noise is a worthy aim. There is a huge amount of effort is required to achieve a noise-free result. It is very difficult to remove this noisy data as of the intrinsic density of natural language and drawbacks of presently available toolkits, including NLTK. Still, the communications gives a few quantity of “gist” which is useful.

Now various entities and interactions among those entities can be visualized in a HTML markup form in which the entities and their interactions are highlighted in bold as shown in Figure 5.13.

True friendship is never serene. **Oooh** I get by with a little **help** from my friends. A **friend** is **someone** who gives you total freedom to be yourself. An **insincere** and evil **friend** is more to be feared than a **wild beast**; a **wild beast** may wound your **body**, but an evil **friend** will wound your mind. Be courteous to all, but intimate with few, and let those few be well tried before you give them your confidence. I value the **friend** who for me finds **time** on his **calendar**, but I cherish the **friend** who for me does not consult his **calendar**. Friends should be like books, few, but hand-selected. **Louis**, I think this is the **beginning** of a beautiful friendship. The only way to have a friend is to be one. Be slow to fall into friendship, but when you are in, continue firm and constant. True **friends** are like **diamonds** – bright, beautiful, valuable, and always in style. I have two **friends**, **Steve** and Martin. But I'd happily replace both for the friendship of Steve Martin. Some people go to priests. Others to poetry. I to my friends. I never had any friends later on like the ones I had when I was twelve. There's not a word yet, for old friends who've just met. **Lots** of **people** want to ride with you in the **limo**, but what you want is **someone** who will take the **bus** with you when the **limo** breaks down. There is **nothing** like **puking** with **somebody** to make you into old friends. Wishing to be **friends** is quick **work**, but **friendship** is a slow ripening fruit. It takes a great **deal** of **courage** to stand up to your **enemies**, but a great **deal** more to stand up to your friends. **Do** I not destroy my **enemies** when I make them my friends? **Anybody** can sympathise with the **sufferings** of a **friend**, but it requires a very **fine nature** to sympathise with a **friend**'s success. The best mirror is an old friend. There is **nothing** better than a **friend**, unless it is a **friend** with chocolate. A **friend** is one that knows you as you are, **understands** where you have been, accepts what you have become, and still, gently allows you to grow. **Remember George**, no **man** is a **failure** who has friends. **Friendship** is the hardest **thing** in the world to explain. It's not something you learn in school. But if you haven't learned the **meaning** of **friendship**, you really haven't learned

Figure 5.13 Visualizing Entities and their Interactions in HTML form

Chapter 6 Conclusion and Future Scope

6.1 Conclusion

In this thesis, Google+ activity data is extracted from Google+ APIs. Python programming language is used to interactively analyze and explore this human language data. Firstly various HTML markups and stopwords present in the Google+ activity data are removed. After that some basic frequency analysis is performed to find out most frequent words and number of sentences, words, Hapaxes *etc*. A frequency distribution is made and plotted on a graph which shows that the most frequent words in the post occurs on the top in the frequency table. The relevant post from the corpus of posts containing human language data is retrieved by using TF-IDF metric and similarity of different posts is computed by using cosine similarity metric. The similarity of documents is visualized by matrix diagram to have a better insight into the clusters obtained after computing document similarity. Then this activity dataset is analyzed to find out various collocations present in the text data. The abstract/summary of a post data is determined by using Luhn's document summarization algorithm. This method has been used to analyze gist of the post without going through the whole post. After that various entities from Google+ posts we extract and interactions between them are analyzed. By analyzing interactions between various entities has been identified "How Entities are communicating".

6.2 Future Work

- To mine the comment feed and to find trends based on commenting frequency, count of comments and count of times a post is re-shared related to the GooglePlus users may also be analysed.
- The text mining fundamentals related to the other blogs present in the web by using various web scraping and crawling frameworks can be analysed for harvesting various web pages.
- Bayesian classifier may be explored to label training samples such as documents and train a classifier for documents entity extraction using NLTK.

References

[1] R. Agrawal and M. Batra, "*A Detailed Study on Text Mining Techniques*", International Journal of Soft Computing and Engineering, Vol. 2, No. 6, pp. 2231-2307, 2013.

[2] R. Ferreira, F. Freitas, L. D. S. Cabral, R. D. Lins, R. Lima, G. Franca, S. J. Simske and L. Favaro, "*A Context Based Text Summarization System*", In Proceedings of 11th IAPR International Workshop on Document Analysis System, pp. 66–70, 2014.

[3] A. Ito, Y. Uno, R Masumura, M. Ito and S. Makino, "*Relevant document retrieval using a spoken document*", In Proceedings of 9thInternational Symposium on Communication and Information Technology, Vol. 3, No.4, pp. 1438-1488, Sep. 2009.

[4] L. Shuang and H. Zhi, "*Analysis of distributed information retrieval*", In Proceeding of International conference on Multimedia Technology, pp. 5297–5300, July 2011.

[5] S. Beliga and S. Martincic, "*Non-Standard Words as Features for Text Categorization",* In Proceeding of International Convention on Information and Communication Technology, Electronics and Microelectronics, pp. 1165-1169, May 2014.

[6] W. Zhao, V. S. Martha, G. Chen, and X. Xu, "*Fast Information Retrieval and Social Network Mining via Cosine Similarity Upper Bound*", In Proceedings of International Conference on Soft Computing, pp. 940-943, September 2013

[7] J. Paralic, and I. Kostial, "*Ontology-based information retrieval*", In Proceedings of the 14th International Conference on Information and Intelligent systems, pp. 23-28, 2013.

[8] U. Rashid, I. A. Niaz and M. A. Bhatti, "M^3L: Architecture for Multimedia Information Retrieval", 6th IEEE International Conference on Information Technology: New Generations, Vol. 14, pp. 1067-1072, April 2009.

[9] E. Loper and B. Steven, "*NLTK: The Natural Language Toolkit*", In Proceedings of the ACL-02 Workshop on Effective tools and methodologies

for teaching natural language processing and computational linguistics, vol. 1, pp. 63-70, 2002.

[10] F. Smadja, "*Retrieving Collocations from Text: Xtract*", Journal of Computational Linguistics-Special issue on using large corpora, Vol. 19, No. 1, pp. 143-177, March 1993.

[11] L. Muflikhah and B. Baharudin, "*Document clustering using concept space and cosine similarity measurement*", IEEE International Conference on Computer Technology and Development, Vol. 1, pp. 58-62, 2009.

[12] W. Chen, G. Wang and F. Yin, "*Document Similarity Calculation Model of CSLN*", In Proceedings of 5th IEEE International Conference on Software Engineering and Service Science, pp. 859-862, June. 2014.

[13] H. Alam, A. Kumar, M. Nakamura, F. Rahman, Y. Tarnikova and C. Wilcox, "*Structured and Unstructured Document Summarization: Design of a Commercial Summarizer using Lexical Chains*", In Proceedings of the 7th International Conference on Document Analysis and Recognition, pp. 1147-1753, 2013.

[14] Z. He, C. Chen, J. Bu, C. Wang and L. Zhang, "*Document Summarization Based on Data Reconstruction*", In proceedings of 26th AAAI international conference on Artificial Intelligence, pp. 620-626, 2012.

[15] C. Li, F. Liu, F. Weng and Y. Liu, "*Document Summarization via Guided Sentence Compression*", In Proceedings of Conference on Empirical Methods in Natural Language Processing, pp. 490-500, October2013.

[16] M. Jamali and H. Abolhassan, "*Different Aspects of Social Network Analysis* ", In Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence, pp. 66-72, Mar. 2006.

[17] M. Cha , H. Haddadi and P. K. Gummadi, "*Bigram-based natural language model and statistical motion symbol model for scalable language of humanoid robots*", In Proceedings of IEEE International Conference on Robotics and Automation, pp. 1232-1237, 2012.

[18] B. Pang and L. Lee, "Chinese text feature extraction method based on bigram", In Proceedings of IEEE International Conference on Circuits and Systems, Vol.2, No.1-2, pp. 342-346, 2013.

[19] C. T. Butts, "*Social network analysis: A methodological introduction*", Asian Journal of Social Psychology, Vol. 11, No. 1, pp. 13-41, 2008.

[20] L. C. Freeman, "*The Development of Social Network Analysis—with an Emphasis on Recent Events*", In Proceedings of the 5th Conference on Language Resources and Evaluation, pp. 417-422, 2006.

[21] T. Gruber, "*Collective knowledge systems: Where the Social Web meets the Semantic Web*", Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 6, No.1, pp. 4-13, 2008.

[22] I. H. Witten, Z. Bray, M. Mahoui and B. Teahan, "*Text mining: A new frontier for lossless compression*", In Proceedings of IEEE Conference on Data Compression, pp. 198-207 ,March 1999.

[23] V. Gupta and G. S. Lehal, "*A Survey of Text Mining Techniques and Applications*", Journal of Emerging Technologies in Web Intelligence, Vol. 1, No. 1, pp. 60-76, 2009.

[24] J. H. Kroeze, M.C. Matthee and T. J. D. Bothma, "*Differentiating Data- and Text-Mining Terminology*", In Proceedings of the Annual Research Conference of the South African Institute of Computer Scientists and Information technologists on Enablement through Technology, pp. 93-101, 2003.

[25] D. Sanchez, M. J. Martin-Bautista and I. Blanco, "*Text Knowledge Mining: An Alternative to Text Data Mining*", In Proceedings of IEEE International Conference on Data Mining Workshops, pp. 644-672, 2008.

[26] D. D. Lewis, K. S. Jones, "*Natural language processing for information retrieval*",Communications of the ACM, Vol. 39, No. 1, pp. 92–101, 1996.

[27] T. Cioara, I. Anghel, I. Salomie and M. Dinsoreanu, "*A context-based semantically enhanced information retrieval model*", In Proceedings of the 5th IEEE International Conference Intelligent Computer Communication and Processing, pp. 245–250, Aug. 2009.

[28] Y. J. Nasukawa, T. W. Niblack and R. Bunescu, "*Information Retrieval based on Grid*", In Proceedings of the IEEE international conference Intelligent Computation Technology and Automation, Vol. 2, pp. 537–540, May 2010.

[29] L. Shuang, H. Zhi, "*Analysis of Distributed Information Retrieval*", In Proceedings of the IEEE International Conference on Multimedia Technology, pp. 5297-5300, July2011.

[30] N. J Belkin, "*Interaction with Texts: Information Retrieval as Information-Seeking Behavior***",** Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 3, pp. 55-66, 1993.

[31] Y. Huang, G. Li and Q. Li, "*Rough Ontology Based Semantic Information Retrieval*", 6th IEEE International Symposium on Computational Intelligence and Design, Vol. 1, pp. 63-67, Oct.2013.

[32] A. Mislove, M. Macron, K. P. Gummadi, P. Druschel and B. Bhattacharjee, "*Measurement and analysis of online social networks*", In Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, pp. 29-42, 2007.

[33] F. E. John and D. Abbott, "*A Comparison of Leading Data Mining Tools*" In Proceedings of Fourth International Conference on Knowledge discovery and Data Mining, pp. 19-25, Aug. 1998.

[34] N. Oren, "*Reexaminingtf.idf based information retrieval with Genetic Programming*", In Proceedings of the 31st International Conference on very large data bases, pp. 661-672, 2005.

[35] W. Zhang, T Yoshida and X. Tang, "*A comparative study of TFIDF, LSI and multi-words for text classification*", Journal of Expert Systems with Applications, Vol. 38, No. 3, pp. 2758-2765, 2011.

[36] X. Guo, Y. Xiang and Q. Chen, "*A Vector Space Model approach to Social Relation Extraction from Text Corpus*", In Proceedings of the 8thIEEEInternational Conference on Fuzzy systems and knowledge Discovery, Vol. 3, pp.1756-1759, 2011.

[37] L. Liu, M. Zhong and R. Lu, "*Measuring Word Similarity Based on Pattern Vector Space Model*", In Proceedings of IEEEInternational Conference on Intelligence and Computational intelligence, Vol. 3, pp. 72-76, 2009.

[38] A. Wibowoand A. Handojo and A. Halim, "*Application of Topic Based Vector Space Model with WordNet*", In Proceedings of IEEE International Conference on Uncertainty Reasoning and Knowledge Engineering, Vol.1, pp. 133-1360, 2011.

[39] X. Xiaoping and Yan Junhu, "*A Kind of Improved Vector Space Model*", In Proceedings of 8th IEEE International Conference on Dependable, Autonomic and Secure Computing, pp. 861-862, 2009.

[40] N. Dehak, R. Dehak, J. Glass, D. Reynolds and P. Kenny, "*Cosine Similarity Scoring without Score Normalization Techniques*", Odyssey, pp. 15, June 2010.

[41] S. Tata and J. M. Patel, "*Estimating the Selectivity of tf-idf based Cosine Similarity Predicates*", ACM SIGMOD, Vol. 36, No. 2, pp. 7-12, 2007.

[42] S. S. Nyein, "*Mining Contents in Web Page Using Cosine Similarity*", In Proceedings of 3rd International Conference on Computer Research and Development, Vol. 2, pp. 472-475, March 2011.

[43] M. Rezvani and S. M. Hashemi, *"Enhancing Accuracy of Topic Sensitive PageRank using Jaccard Index and Cosine Similarity"*,In Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Vol. 1, pp. 620-624, Dec. 2012.

[44] Y. Dong, Z. Sun and H. Jia, "*A cosine similarity-based negative selection algorithm for time series novelty detection*" Journal on Mechanical Systems and Signal Processing, Vol. 20, No.6, pp.1461-1472, 2006.

[45] L. Muflikhahand B. Baharudin, "Document Clustering using Concept Space and Cosine Similarity Measurement", International Conference on Computer Technology and Development, Vol. 1, pp. 58-62, Nov. 2009.

[46] D. Shen, J. T. Sun, H. Li, Q. Yang and Z. Chen, "*Document Summarization using Conditional Random Fields*", Vol. 7, pp. 2862-2867, 2007.

[47] M. Pennacchiotti and P. Pantel, "Entity Extraction via Ensemble Semantics", In Proceedings of Conference on Empirical Methods in Natural Language Processing on Association for Computation, Vol. 1, pp. 238-247, 2009.

[48] M.A. Russell, Mining the Social Web, O'Reilly Media.

[49] W. Wang. C. Xiao, X. Lin and C. Zhang, "Efficient Approximate Entity Extraction with Edit Distance Constraints", In Proceedings of the ACM SIGMOD International Conference on Management of data, pp. 759-770, 2009.

List of Publications

[1] I. Arya and V. P. Singh, "*Querying Google+ Activities Data with TF-IDF and computing Document Similarity using Cosine Similarity Metric*", IEEE 8th International Conference on Contemporary Computing (IC3), JIIT Noida, Aug 20-22, 2015. [**Communicated**]