IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-4, Issue-9 E-ISSN: 2347-2693

Text Mining of Unstructured Data Using R

M. Siva Lakshmi[1] and MD. Arsha Sultana[2]

[1]B.Tech Student, Computer Science and Engineering, NRI Institute of Technology, Pothavarappadu, India

[2] Assistant Professor, Computer Science and Engineering, NRI Institute of Technology, Pothavarappadu, India

Available online at: www.ijcseonline.org

Received: 22/Aug/2016 Revised: 02/Sept/2016 Accepted: 20/Sept/2016 Published: 30/Sep/2016

***Abstract*—** Text mining is the process of acquiring high-quality information from text that is typically borrowed through the devising of patterns and trends such as statistical pattern learning .It usually involves the process of structuring the input text, deriving patterns within the structured data and finally evaluation and interpretation of the output. It can help an organization to acquire potentially valuable business insights from text-based content such as word documents, email and postings on social media streams like Facebook, Twitter and LinkedIn. Mining unstructured data with natural language processing (NLP), statistical modeling and machine learning techniques can be challenging because natural language text is often inconsistent. So, R is used to mine unstructured data which is the most exhaustive statistical analysis package and it incorporates all of the standard statistical tests, models and analyses for managing and manipulating data.

***Keywords*—** R ,S,Text mining,Statistical Modeling

I. INTRODUCTION

Text Mining is one of the most convoluted scrutinizes in the industry of analytics because dealing with unstructured data is not certainly determine observation and variables (rows and columns). Thus for doing any kind of analytics, first transform this unstructured data into a structured dataset and then continue with normal modeling framework [3]. The additional step of transforming an unstructured data into a structured format is simplified by a Word dictionary especially crucial to do all kind of information eradication. To accomplish a sentiment scrutiny in dictionary is easily available on web world but for a few distinct scrutiny need to create a dictionary of our own.

The dictionary need in the firm obstacle looks like a very slot dictionary because we need to identify all names from the transaction free text. Probability of such a dictionary is reliably very low [1]. So, we need to devise a dictionary and then record integrated data set by using R to build an unstructured data model. R is very convenient to devise dictionary on smaller datasets.

Statistical modeling is a simplified, mathematically-determine way to approximate reality and make predictions from this approximation. It is collecting, encapsulating, analyzing and translating variable numerical data. Statistical methods can be contradicted with deterministic methods which are relevant where observations are precisely reproducible or assumed. Statistical methods are extensively used in life sciences, economics and agricultural science [7]. They also have an important role in the physical science in the study of evaluation errors of random phenomena such as radio activity or meteorological events and retrieving approximate results where deterministic solutions are hard to apply.

R is a programming language and environment developed for statistical scrutiny by practicing statisticians and researchers. It reflects well on a very competent community of computational statisticians [6]. R is free and open source software allowing anyone to use and to customize it. R is licensed under the GNU General Public License, with copyright held by The R Foundation for Statistical Computing. R has no license restrictions and so can run it anywhere and at any time, and even sell it under the conditions of the license.

II. OVERVIEW

Unstructured data is a universal design for characterize data especially not accommodated in a database or some other type of data structure textual or non-textual. Textual unstructured data is spawn in media like email messages, PowerPoint presentations, Word documents etc. Non-textual unstructured data is spawn in media like JPEG images, MP3 audio files and Flash video files. These consequences are in inconsistency and ambiguities that produce it difficult to interpret using traditional programs as compared to data stored in databases or annotated in documents.

In 1998, Merrill Lynch indicated a rule of thumb that around 80-90% of all probably usable business information may derive in unstructured form. This rule of thumb is not based on fundamental or significant research but however is accepted by some. IDC and EMC project data will become to

40 zettabytes by 2020 resulting in a 50-fold success from the beginning of 2010.Unstructured information might report more than 70%–80% of all data in organizations.
Simple techniques for structuring text generally contain manual tagging with metadata or part-of-speech tagging for more text mining. Unstructured Information Management Architecture (UIMA) implements a common structure for transforming information to extract content and devise structured data about the information.

Software that devises machine-process capable structure exploits the linguistic, auditory and visual structure built-in in all forms of human communication. Algorithms can interpret this built-in structure from text for instance by auditing word morphology, sentence syntax and other patterns. Unstructured information can be enhanced and identify to address ambiguities and relevancy-based techniques to simplify search and discovery [2]. Examples of "unstructured data" may consist of books, journals, documents, metadata, health records, audio, video, analog data, images, files, and unstructured text such as body of an e-mail message, Web page, or word-processor document. Main content comes packaged in files or documents that they have structure and are incorporate of structured and unstructured data, but generally this is still referred to as "unstructured data". For example, an HTML web page is identified, but HTML mark-up consistently serves simply for rendering. It does not capture the content of identified elements that support computerized processing content of the page. XHTML identified grant machine processing of elements, despite it consistently does not capture or transfer the semantic content of identified terms.

Unstructured data generally occurs in electronic documents. Document management system classifies entire documents which is often favored over data transfer and manipulating within the documents to transfer structure into document. Search engines turn into popular tools for indexing and searching data, especially text.

The generic S philosophy is important for users of S and R to understand because it arranged the moment for the design of the language itself which many programming expert find a bit different and complicated. It is important to understand that the S language had its origin in data scrutiny and do not arrive from a traditional programming language background. Its inventors were focused on computation out how to make data scrutiny easier, first for themselves and then finally for others. In the Evolution of S, John Chambers writes: “[W]e wanted users to be capable to begin in an interactive environment, where they did not consciously think of themselves as programming. Then as their needs became clearer and their sophistication increased, they should be able to slide gradually into programming, when the language and system aspects would become more important.”

The key part here was the transition from user to developer. They wanted to build a language that could easily service both “people”. More technically, they needed to build language that would be suitable for interactive data analysis (more command-line based) as well as for writing longer programs (more traditional programming language-like).

In the early days, a key feature of R was that its syntax is very similar to S, making it easy for S-PLUS users to switch over. While the R’s syntax is nearly identical to that of S’s, R’s semantics, while superficially similar to S, are quite different [4]. In fact, R is technically much closer to the Scheme language than it is to the original S language when it comes to how R works under the hood. Today R runs on almost any standard computing platform and operating system. Its open source nature means that anyone is free to adapt the software to whatever platform they choose. Indeed, R has been reported to be running on modern tablets, phones, PDAs, and game consoles.

One nice feature that R shares with many popular open source projects is frequent releases. These days there is a major annual release, typically in October, where major new features are incorporated and released to the public. Throughout the year, smaller-scale bug fix releases will be made as needed. The frequent releases and regular release cycle indicates active development of the software and ensures that bugs will be addressed in a timely manner [6]. Of course, while the core developers control the primary source tree for R, many people around the world make contributions in the form of new feature, bug fixes, or both. Another key advantage that R has over many other statistical packages (even today) is its sophisticated graphics capabilities. R’s ability to create “publication quality” graphics has existed since the very beginning and has generally been better than competing packages.

Today, with many more visualization packages available than before, that trend continues. R’s base graphics system allows for very fine control over essentially every aspect of a plot or graph. Other newer graphics systems, like lattice and ggplot2 allow for complex and sophisticated visualizations of high-dimensional data [2]. R has maintained the original S philosophy, which is that it provides a language that is both useful for interactive work, but contains a powerful programming language for developing new tools. This allows the user, who takes existing tools and applies them to data, to slowly but surely become a developer who is creating new tools.

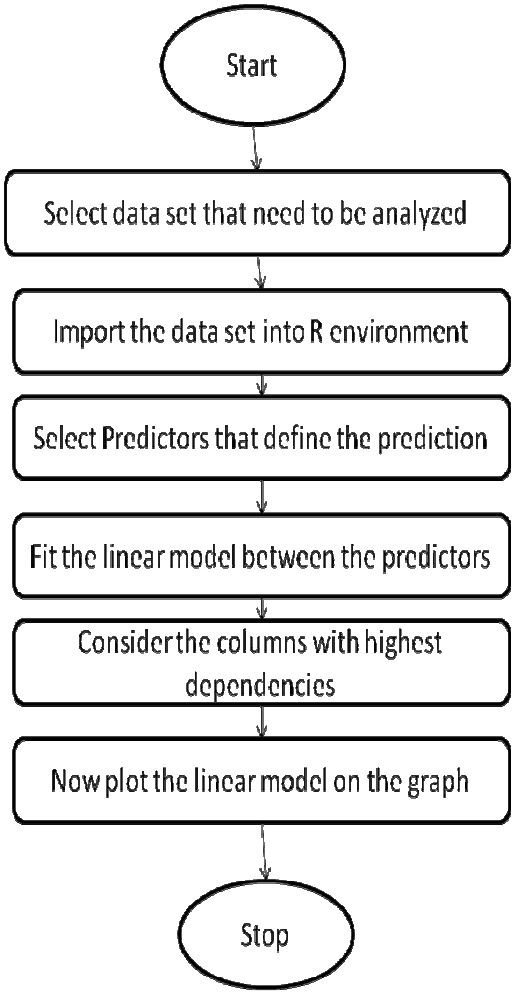

Fig.1.Flow diagram

III. IMPLEMENTATION

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes an effective data handling and storage facility, a suite of operators for calculations on arrays, in particular matrices, a large, coherent, integrated collection of intermediate tools for data analysis, graphical facilities for data analysis and display either on-screen or on hardcopy, and a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities [2]. The term "environment" is intended to characterize it as a fully planned and coherent system, rather than an incremental accumulation of very specific and inflexible tools, as is frequently the case with other data analysis software.

Many users think of R as a statistics system but prefer it to think of it of an environment within which statistical techniques are implemented. R can be extended (easily) via packages. There are about eight packages supplied with the R distribution and many more are available through the CRAN family of Internet sites covering a very wide range of modern statistics [6]. R has its own LaTeX-like documentation format, which is used to supply comprehensive documentation, both on-line in a number of formats and in hardcopy.

R provides two packages for working with unstructured text – TM and Sentiment. TM can be installed in the usual way. Unfortunately, Sentiment has been archived in 2012, and is therefore more difficult to install. Once initially installed, each can be loaded later as library (name).

- The next step is to load the data. This data was saved to a text file and loaded and processed

```
>setwd("G://r lng")
>data ← read.csv ("reviews.csv",header=TRUE)
```

- Now, change the data csv to doc format

```
>dtm← DocumentTermMatrix(courpus)
>courpus←corpus(source)
>courpus
```

- Delete whitespaces between rows

```
>text←data$text
>text←paste(data$text,collapse=" ")
>text
```

- Remove stopwords

```
>stopwords("english")
>courpus←tm_map(courpus,removewords,stopwords("englis
h"))
```

- Next, remove nonessential characters such as punctuation, whitespaces, numbers etc from the text, before processing the actual words themselves.

```
>courpus← tm_map(courpus.content_transformer(tolower))
>courpus←tm_map(courpus,removePunctuation)
>courpus←tm_map(courpus,stripwhitespace)
```

- Denote as matrix and then Find Frequency of words, sort it and see head

```
>dtm2←as.matrix(dtm)
>frequency←colSums(dt2)
>frequency←sort(frequency,decreasing=TRUE)
>head(frequency)
```

- Retrieve data from dataset

```
>text←data$text
```

- Now, Finally represent in a graph

```
>barplot(head(frequency))
```


IV. RESULTS

After installing required libraries and tm package, load the data in R

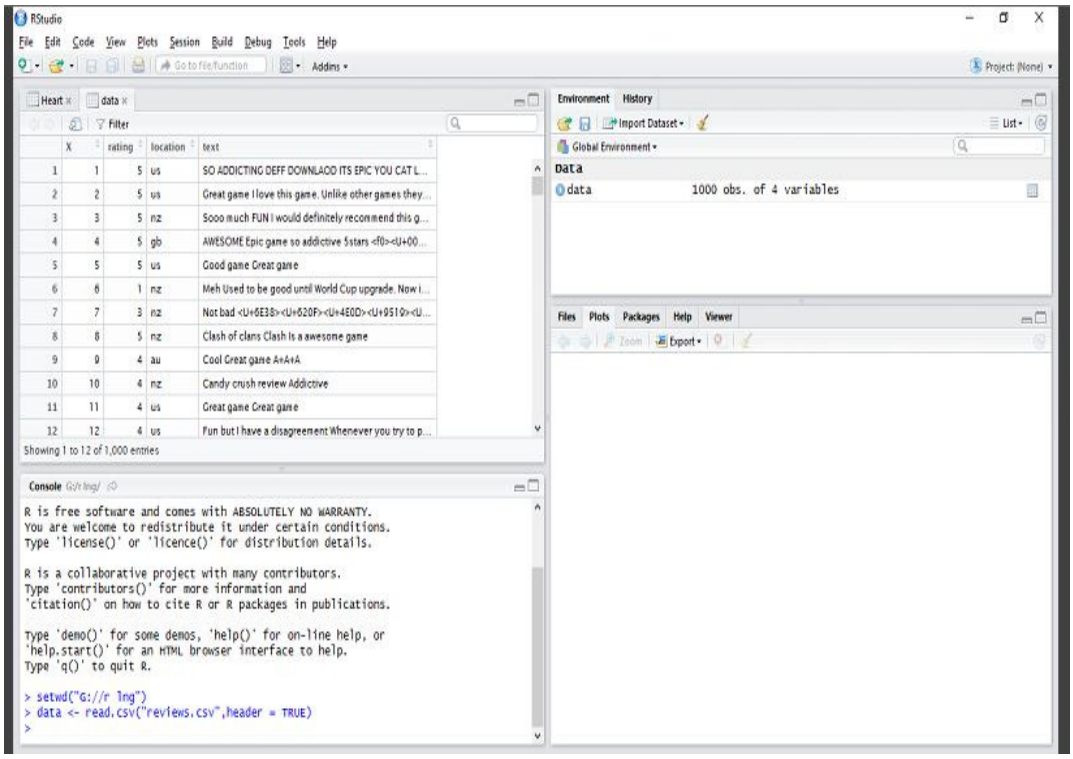

Fig.2. Input data

Change the data csv into doc format

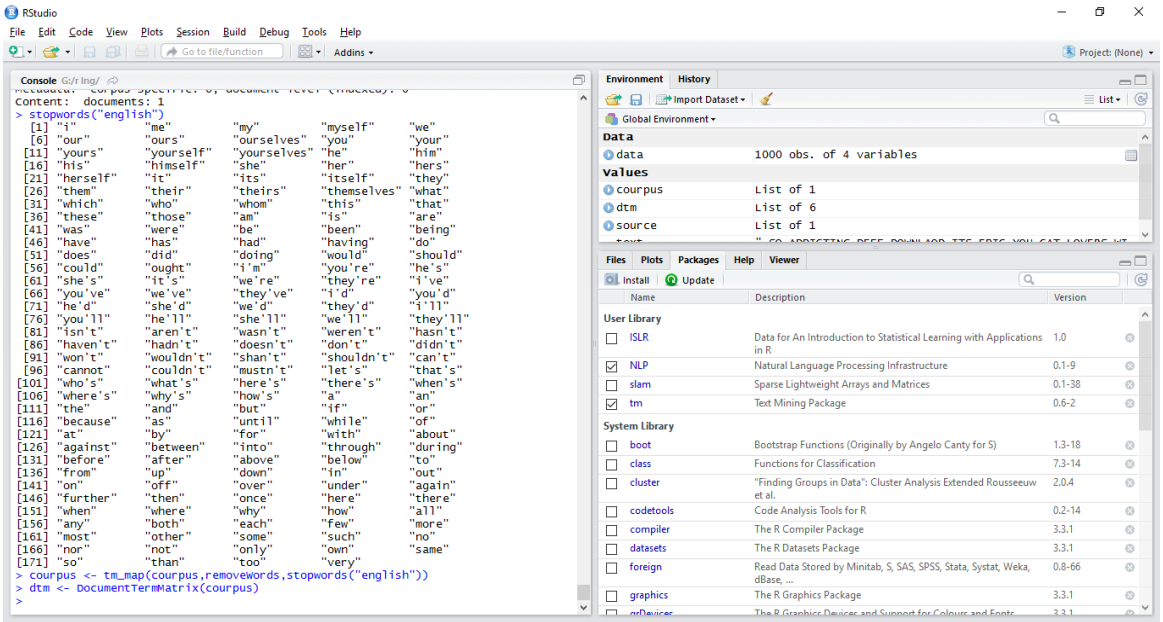

Fig.3. Doc format

Delete whitespaces between rows

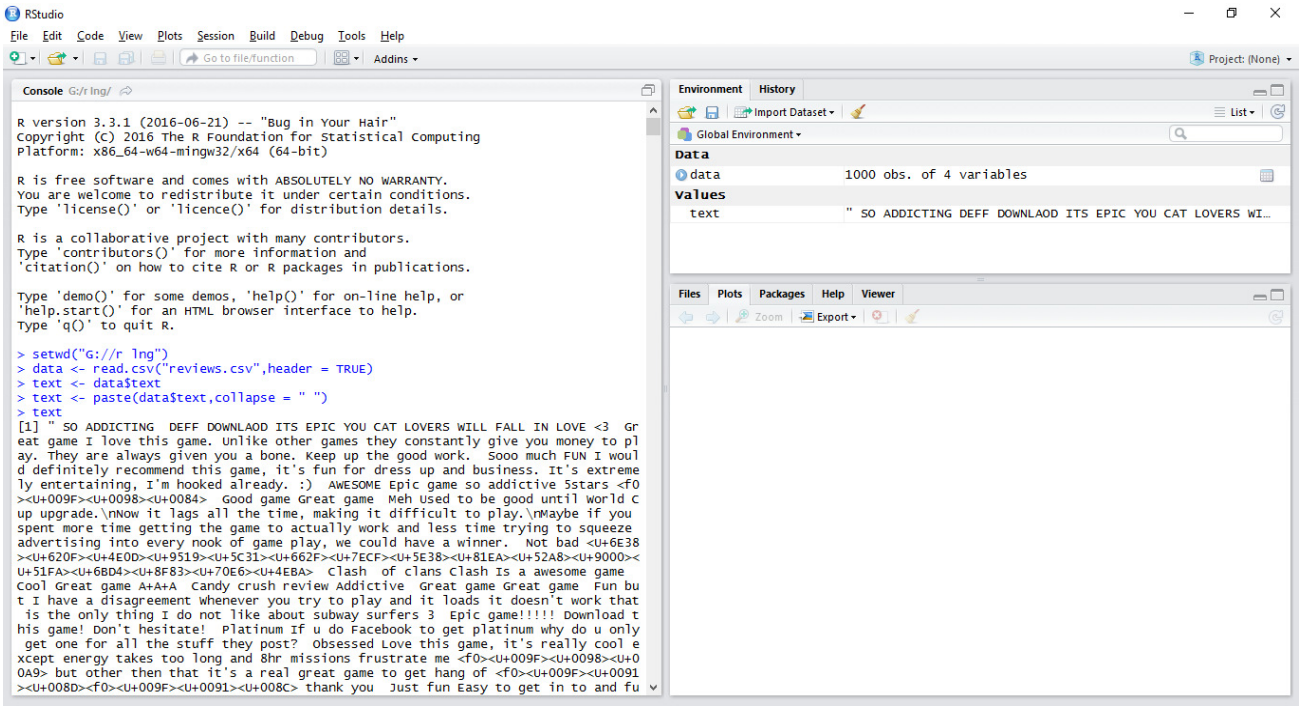

Fig.4.Removing Whitespaces

Remove stopwords

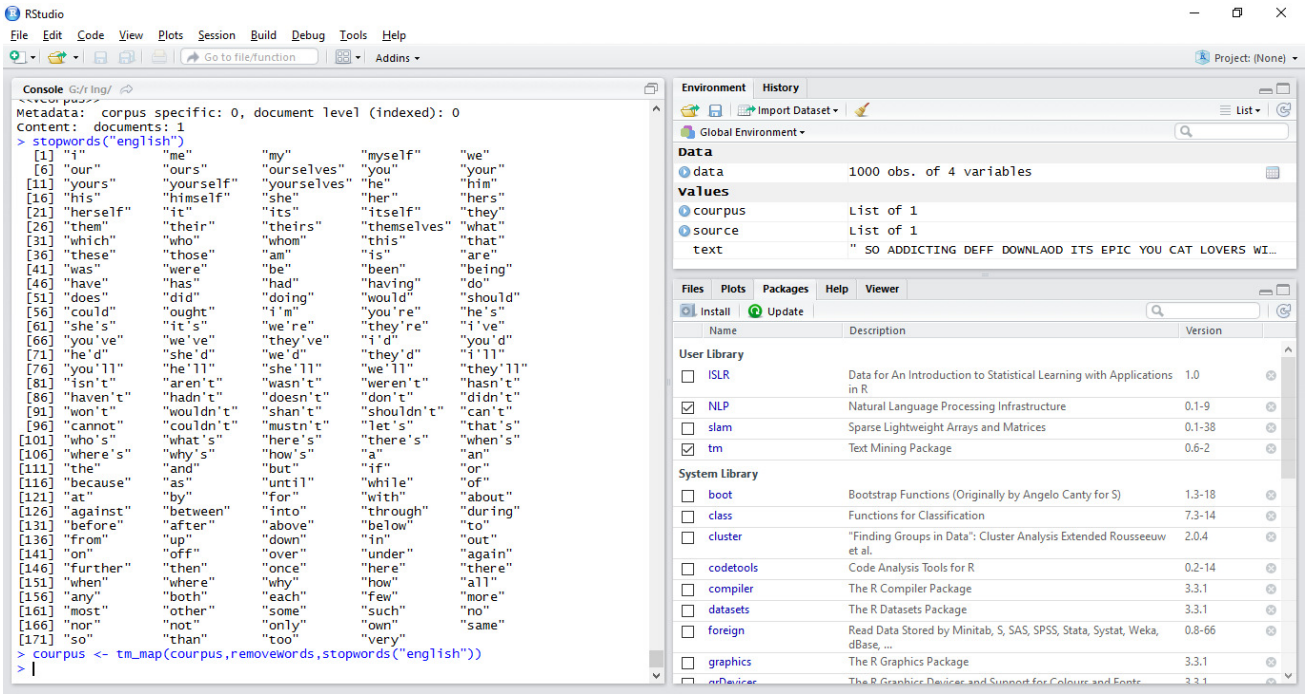

Fig.5.Removing stopwords

Next, remove nonessential characters such as punctuation, whitespaces, numbers etc from the text, before processing the actual words themselves.

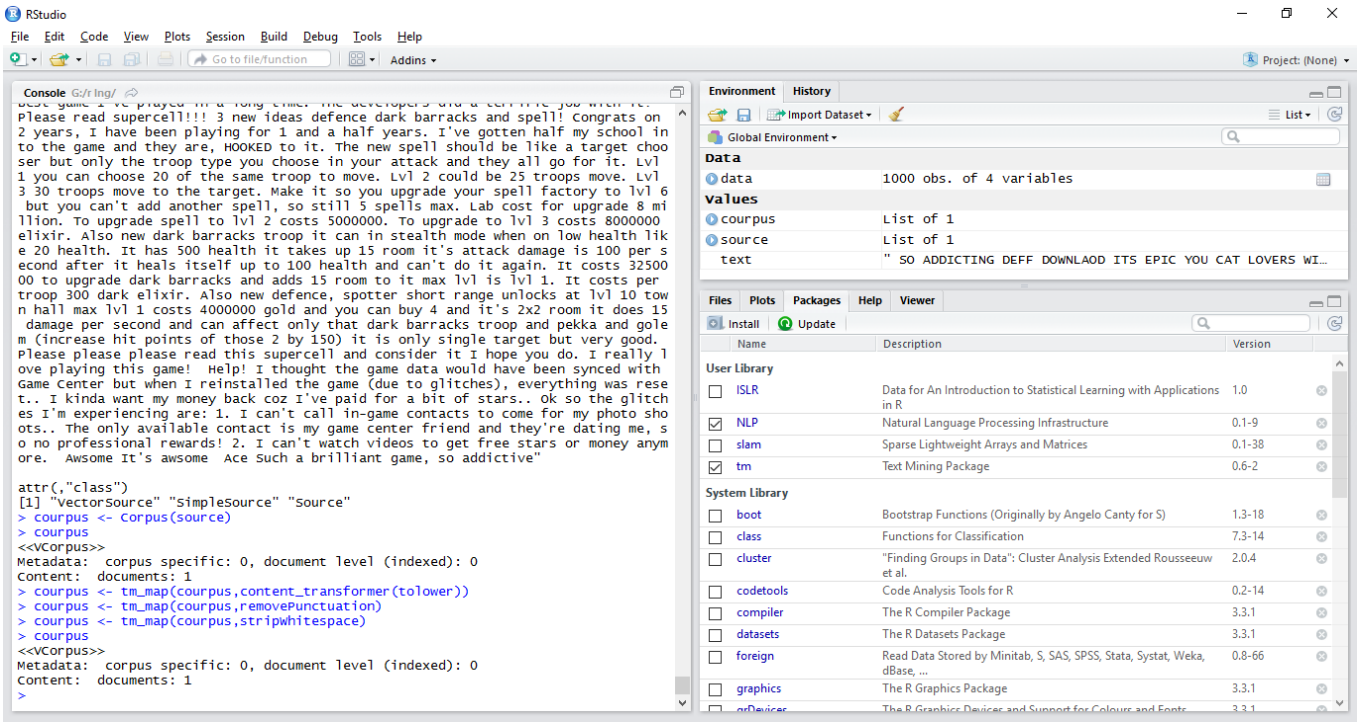

Fig.6. Removing nonessential characters

Denote as matrix and then Find Frequency of words, sort it and see head

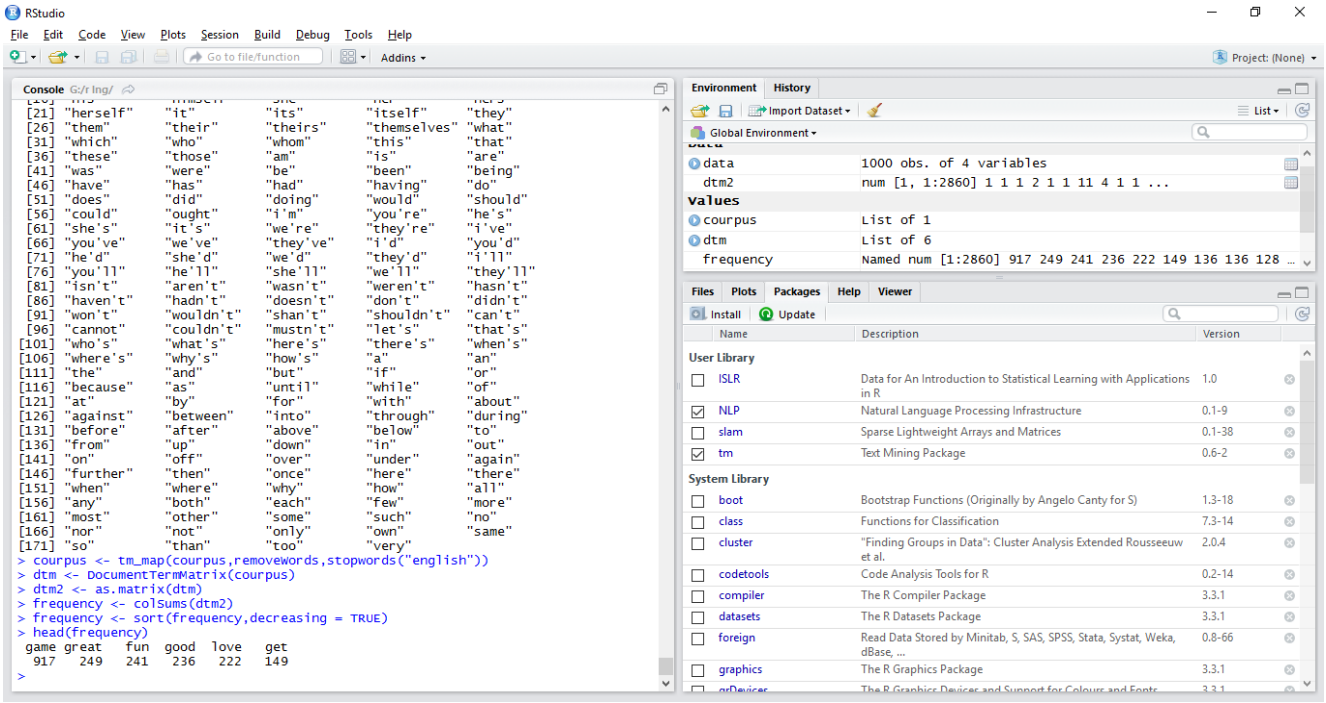

Fig.7. Freq of Words

Retrieve data from dataset

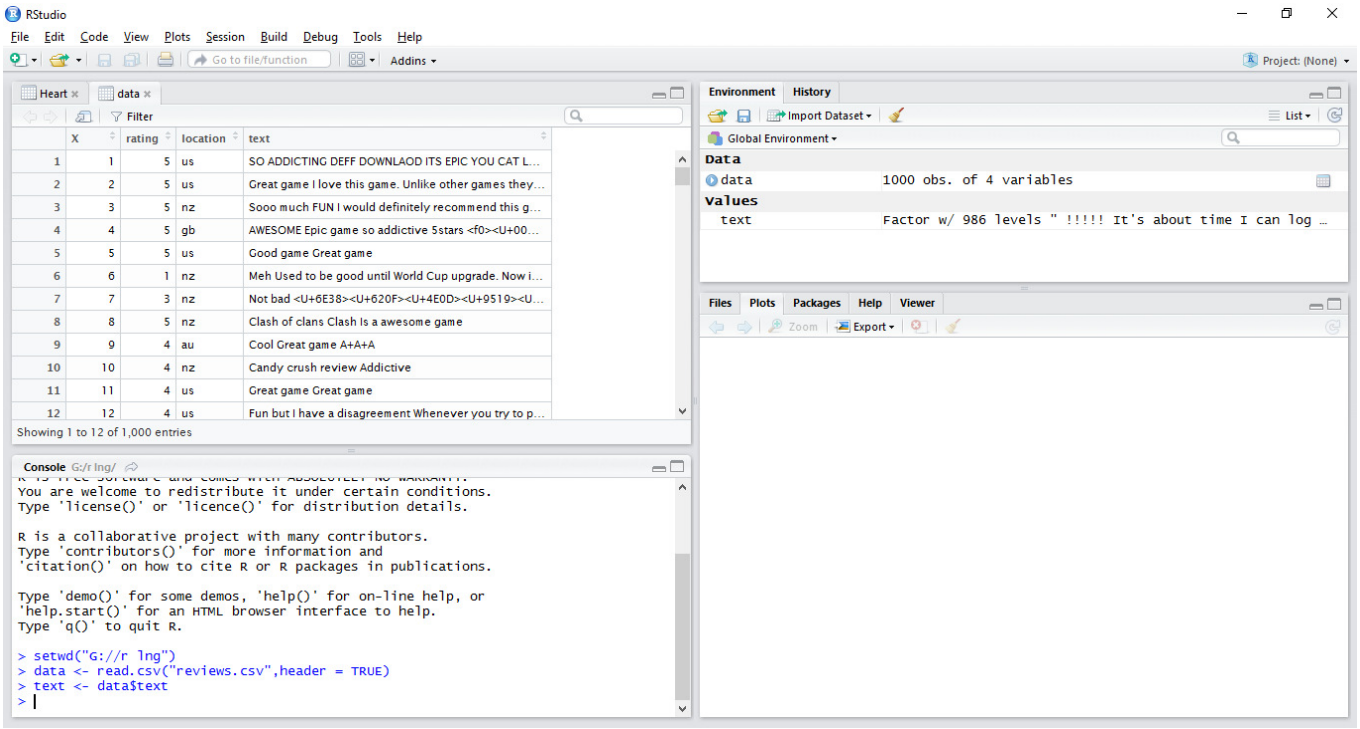

Fig.8. Retrieve Data

Now, finally represent in a graph

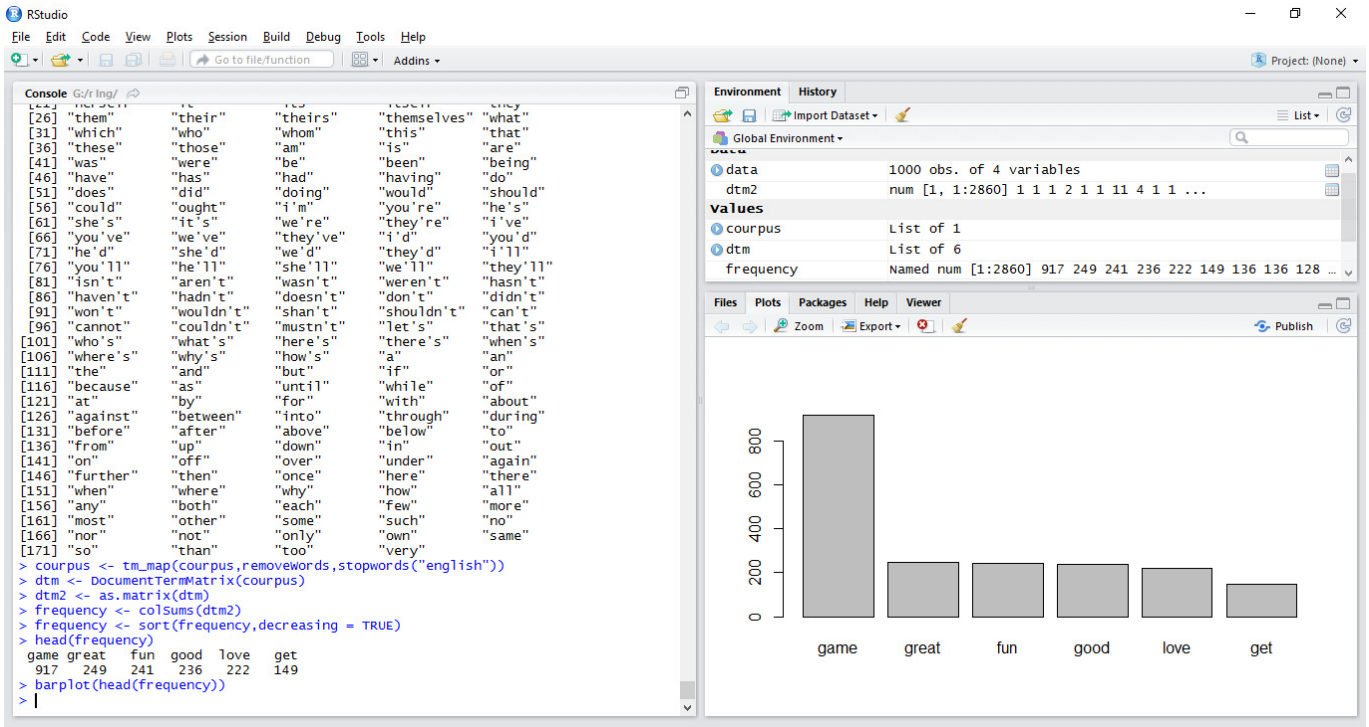

Fig.9. Graph representation

V.CONCLUSION

R is used to mine unstructured data which is the most exhaustive statistical analysis package and it incorporates all of the standard statistical tests, models and analyses for managing and manipulating data. By using R only useful information can be gathered by removing unnecessary nonessential characters from unstructured data. So, we can easily predict the status of firm retrieving header frequency from unstructured data.

REFERENCES

[1] Mr. Rahul Patel, Mr. Gaurav Sharma,"A survey on text mining techniques", Int. Journal of Engineering and Computer Science, Volume-03, Issue 5, Page No. (5621-5625), May 2014.

[2] Minakshi R. Shinde1, Parmeet C. Gill, "Pattern Discovery Techniques for the Text Mining and its Applications", Int. Journal of Science and Research (IJSR), Volume-03 Issue 5, May 2014.

[3] S.S. Patil and V.M. Gaikwad , "Developing New Software Metric Pattern Discovery for Text Mining", International Journal of Computer Sciences and Engineering, Volume-02, Issue-04, Page No (119-125), Apr -2014

[4] Abhilasha Singh Rathor, Dr. Pankaj Garg," Analysis on Text Mining Techniques", Int. Journal of Advanced Research in Computer Science and Software Engineering, Volume -06, Issue 2, Page No (132-137), February 2016.

[5] Vishakha D. Bhope and Sachin N. Deshmukh, "Comparative Study on Information Retrieval Approaches for Text Mining", Int. Journal of Computer Sciences and Engineering, Volume-03, Issue-3, Page No (102-106), Mar 2015.

[6] Vishakha D. Bhope and Sachin N. Deshmukh, "Comparative Study on Information Retrieval Approaches for Text Mining", International Journal of Computer Sciences and Engineering, Volume-03, Issue-03, Page No (102-106), Mar -2015

[7] StatisticalModeling,https://en.wikipedia.org/wiki/Statistical_model, June 2016.

AUTHORS PROFILE

M. Siva Lakshmi is presently B.Tech Student, Dept. of Computer Science & Engineering, NRI Institute of Technology, Pothavarappadu, India.

MD. Arsha Sultana is presently Assistant Professor, Dept. of Computer Science & Engineering, NRI Institute of Technology, Pothavarappadu, India.

