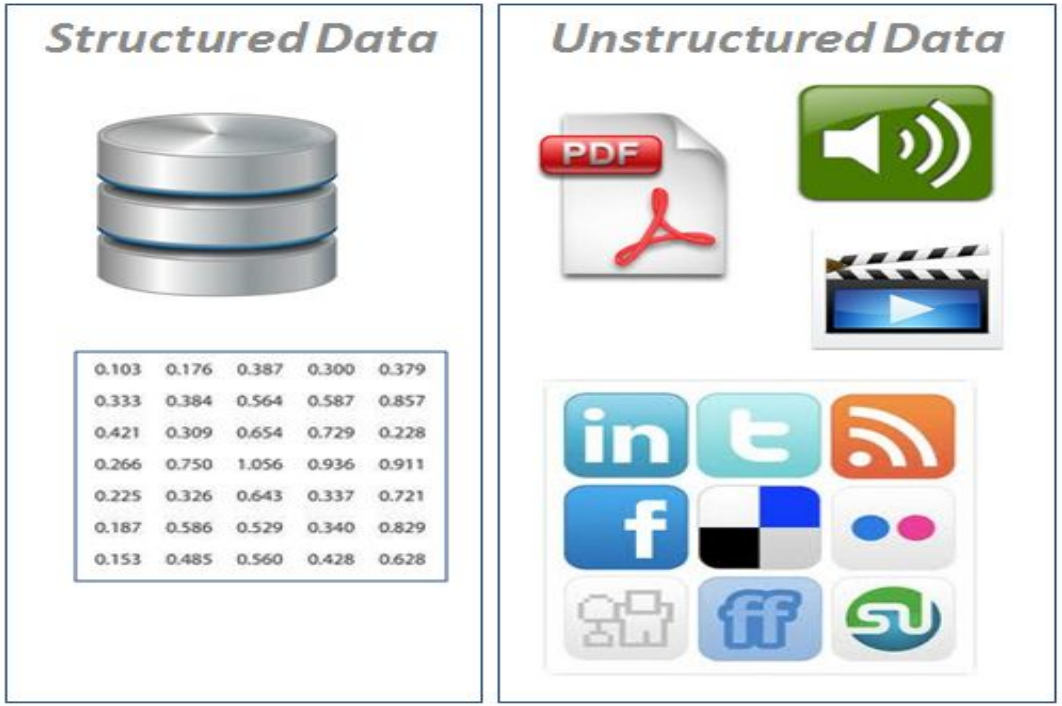
Text Mining - 1

Contents

Structured Vs. Unstructured Data



Features Of Unstructured Data

Does not reside in traditional databases and data warehouses

May have an internal structure, but does not fit a relational data model

Generated by both humans and machines

- Textual and social media content
- Machine-to-machine communication

Examples Of Unstructured Data

Examples of unstructured data include:

- **Personal messaging** – Email, instant messages, tweets, chat
- **Business documents** – Business reports, presentations, survey responses
- **Web content** – Web pages, blogs, wikis, audio files, photos, videos
- **Sensor output** – Satellite imagery, geo-location data, scanner transactions

Value Of Unstructured Data

Unstructured data provides a rich source of information about people, households and economies

- It may enable more accurate and timely measurement of a range of demographic, social, economic and environmental phenomena
  - When combined with traditional data sources
  - As a replacement for traditional data sources

- As a result, presents unprecedented opportunities for official statistics to
  - Improve delivery of current statistical outputs
  - Create new information products not possible with traditional data sources

What Is Text Analysis

- Text Mining is also known as **Text Data Mining (TDM)** and **Knowledge Discovery in Textual Database (KDT)**

- It is a process of identifying novel information from a collection of texts (Also known as a ‘Corpus’)

- **Corpus is a collection of ‘documents’ containing natural language text.** Here, documents, generally, are sentences. Each document is represented as a separate line.

Case Study – HR Appraisal Process Feedback

Background

- The company XYZ carried out Annual Performance Appraisal process which is a routine HR process.
- The employees were asked to give feedback about the overall process and questions used for assessing their performance level.

Objective

- To understand the employee sentiments and incorporate recommendations in the current performance appraisal process.

Available Information

- Feedback and comments from the employees were stored in a text document.

Data Snapshot

Example of data

Text Observations
The process was transparent.
There is a lot of scope to improve the process, as most questions were subjective.
Happy with the process, but salary increment in 2019 is very low as compared to previous years.
Many questions were very subjective. Very difficult to measure the performance.
Questions could have been specific to function. Very general questions.
More research is required to come out with better process next time.
Very happy with the process adopted. Fair and transparent.

* These are the comments received from employees.
Note that, data is not in structured format.

Text Mining In R

```
#Import the data.
#Import text file with one text record in one row
data<-readLines("HR Appraisal process.txt")
head(data)
```

- **readLines**() reads some or all text lines from a file or connection.

```
# Output:
> head(data)
[1] "The process was transparent."
[2] "There is a lot of scope to improve the process, as most questions were subjective."
[3] "Happy with the process, but salary increment in 2019 is very low as compared to previous years."
[4] "Many questions were very subjective. Very difficult to measure the performance."
[5] "Questions could have been specific to function. Very general questions."
[6] "More research is required to come out with better process next time."
```

Interpretation:

- **head()** prints first 6 text lines from the data with each line as one document / observation.

Text Mining In R

```
#Convert this data into 'Corpus'
install.packages("tm")
library(tm)

corp <- Corpus(VectorSource(data))
class(corp)
```

- Install and load **T**ext **M**ining (**tm**) package.
- **Vector source()** interprets each element of the vector as a document.
- **Corpus()** converts and saves data as a corpus.

```
> class(corp)
[1] "SimpleCorpus" "Corpus"
```

Interpretation:

- Class of the data should be Corpus.

* In case NLP is not loaded , Before installing tm , please run the following command
install.packages("NLP")
library(NLP)

Text Mining In R

```
# Inspect Corpus. Here [1:3] displays first 3 textlines.
inspect(corp[1:3])
```

```
<<SimpleCorpus>>
Metadata: corpus specific: 1, document level (indexed): 0
Content: documents: 3

[1] The process was transparent.
[2] There is a lot of scope to improve the process, as most questions
were subjective.
[3] Happy with the process, but salary increment in 2019 is very low as
compared to previous years.
```

```
# Display a particular document from corpus.
writeLines(as.character(corp[[3]]))
```

```
Happy with the process, but salary increment in 2019 is very low as
compared to previous years.
```

- **writelines()** prints text line of specified number in [[]]. Here it is printing 3rd line.

Text Mining In R

```
# Clean the Corpus for further analysis
corp <- tm_map(corp, tolower)
writeLines(as.character(corp[[3]]))
```

```
happy with the process, but salary increment in 2019 is very low as
compared to previous years.
```

```
corp <- tm_map(corp, removePunctuation)
writeLines(as.character(corp[[3]]))
```

```
happy with the process but salary increment in 2019 is very low as
compared to previous years
```

```
corp <- tm_map(corp, removeNumbers)
writeLines(as.character(corp[[3]]))
```

```
happy with the process but salary increment in is very low as compared
to previous years
```

- **tm_map**() applies transformation functions to a corpus.
- **tolower** converts text to lowercase.
- **removePunctuation** removes punctuation.
- **removeNumbers** removes numbers.

Text Mining In R

```
# Clean the Corpus for further analysis
corp <- tm_map(corp, removeWords, stopwords("english"))
writeLines(as.character(corp[[3]]))
```

```
happy process salary increment low compared previous years
```

```
corp <- tm_map(corp, removeWords, "process")
writeLines(as.character(corp[[3]]))
```

```
happy salary increment low compared previous years
```

- **removeWords, stopwords("english")** remove stop words like: i, me, our and, the, is, etc. There are more than 100 in-built English Stopwords in R. Use **stopwords("english")** to view the list of these stopwords.
- If you wish to remove specific words from the corpus, use **tm_map(corp, removeWords, "word")**. Here "**process**" word is removed.

Text Mining In R

```
# Convert to term-document matrix format

tdm <- TermDocumentMatrix(corp)
findFreqTerms(tdm)

# Find terms with frequency of at least 5 and find words having high
association with ‘difficult’, ‘questions’

findFreqTerms(tdm,5)
findAssocs(tdm, 'difficult', 0.60 )
findAssocs(tdm, 'questions', 0.60 )
```

- **TermDocumentMatrix()** finds frequent terms in a document-term or term-document matrix. Default minimum frequency is 1 and maximum is infinite. **DocumentTermMatrix()** and **TermDocumentMatrix()** gives the same output.
- **findFreqTerms()** gives words with minimum specified frequency . **findFreqTerms**(tdm,5) gives words having minimum frequency 5.
- **findAssocs()** gives words with specified minimum correlations with the given word. **findAssocs**(tdm, 'difficult', 0.60) gives words with at least 0.6 correlation with word ‘difficult’.

Text Mining In R

```
# Output:
```

```
> findFreqTerms(tdm)
 [1] "transparent"     "improve"         "lot"           "questions"     "scope"
 [6] "subjective"      "compared"        "happy"         "increment"     "low"
[11] "previous"        "salary"          "years"         "difficult"     "many"
[16] "measure"         "performance"     "function"      "general"       "specific"
[21] "better"          "come"            "next"          "required"      "research"
[26] "time"            "adopted"         "fair"          "benchmark"     "extremely"
[31] "industry"        "methodology"     "rating"        "effort"        "excellent"
[36] "team"            "congratulations" "department"    "improvement"   "needs"
[41] "approach"        "current"         "discussion"    "frequent"      "manager"
[46] "using"           "evaluate"        "possible"      "work"          "disappointed"
[51] "little"          "biased"          "need"          "expected"      "method"
[56] "used"            "good"            "changes"       "clear"         "twice"
[61] "year"            "can"             "consultant"    "hire"          "clearer"
[66] "last"            "selfassessment"  "particular"    "toward"        "appraisal"
[71] "think"           "carried"         "organization"  "way"           "modified"
[76] "communication"   "overall"         "satisfied"     "remains"       "keep"
[81] "members"         "show"            "make"          "minor"         "robust"
[86] "will"            "removed"         "replaced"      "headvery"      "nice"
[91] "smooth"          "appreciate"      "processmust"
```

```
> findFreqTerms(tdm,5)
[1] "questions"   "subjective"  "happy"       "difficult"   "measure"     "performance" "fair"        "work"
>
```

```
> findAssocs(tdm, 'difficult', 0.60 )
$difficult
    measure performance    approach       using
       1.00        0.90        0.61        0.61

> findAssocs(tdm, 'questions', 0.60 )
$questions
subjective
      0.67
```

Interpretation:

- questions, subjective, happy, difficult, measure, performance, fair, work are appearing more than 5 times.
- Word ‘difficult’ is having high correlation with measure, performance.

Word Cloud In R

Word cloud, as the name suggests, is an **image showing compilation of words**, in which, **size of words indicates its frequency or importance**.

```
# Install and load package “wordcloud”
install.packages("wordcloud")
library(wordcloud)

# Convert tdm object to a matrix
m <- as.matrix(tdm)
m
```

Word Cloud In R

```
                Docs
Terms            1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34
  transparent    1 0 0 0 0 0 1 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0
  improve        0 1 0 0 0 0 0 0 0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  lot            0 1 0 0 0 0 0 0 0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0
  questions      0 1 0 1 2 0 0 0 1  0  0  0  0  1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0
  scope          0 1 0 0 0 0 0 0 0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  subjective     0 1 0 1 0 0 0 0 1  0  0  0  0  1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0
  compared       0 0 1 0 0 0 0 1 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  happy          0 0 1 0 0 0 1 0 1  0  0  0  0  0  0  1  0  0  0  0  1  0  0  0  1  0  0  0  0  0  1  0  0  0
  increment      0 0 1 0 0 0 0 1 0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0
  low            0 0 1 0 0 0 0 1 0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  previous       0 0 1 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  salary         0 0 1 0 0 0 0 1 0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0
  years          0 0 1 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  difficult      0 0 0 1 0 0 0 0 0  0  0  0  1  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  1  0  0  0  0  0
  many           0 0 0 1 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  measure        0 0 0 1 0 0 0 0 0  0  0  0  1  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  1  0  0  0  0  0
  performance    0 0 0 1 0 0 0 0 0  0  0  0  1  0  0  0  0  0  0  1  0  0  0  0  1  0  0  0  1  0  0  0  0  0
  function       0 0 0 0 1 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  general        0 0 0 0 1 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  specific       0 0 0 0 1 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
                Docs
Terms            35 36 37 38 39 40 41 42 43 44 45 46 47 48
  transparent     0  0  0  0  0  0  0  0  0  0  0  0  0  0
  improve         0  0  0  0  0  0  0  1  0  0  0  0  0  0
  lot             0  0  0  0  0  0  0  0  0  0  0  0  0  0
  questions       1  1  1  0  0  1  0  0  0  0  0  1  0  0
  scope           0  0  0  0  0  0  0  1  0  0  0  0  0  0
  subjective      0  0  1  0  0  0  0  0  0  0  0  1  0  0
  compared        0  0  0  0  0  0  0  0  0  0  0  0  0  0
  happy           0  0  0  0  1  1  0  0  0  1  0  0  0  0
  increment       0  0  0  0  0  0  0  0  0  0  0  0  0  0
  low             0  0  0  0  0  0  0  0  0  0  0  0  0  0
  previous        0  0  0  0  0  0  0  0  0  0  0  0  0  0
  salary          0  0  0  0  0  0  0  0  0  0  0  0  0  0
  years           0  0  0  0  0  0  0  0  0  0  0  0  0  0
  difficult       0  0  1  0  0  0  0  0  0  0  0  0  0  0
  many            1  0  0  0  0  0  0  0  0  0  0  0  0  0
  measure         0  0  1  0  0  0  0  0  0  0  0  0  0  0
  performance     0  0  1  0  0  0  0  0  0  0  0  0  0  0
  function        0  0  0  0  0  0  0  0  0  0  0  0  0  0
  general         0  0  0  0  0  0  0  0  0  0  0  0  0  0
  specific        0  0  0  0  0  0  0  0  0  0  0  0  0  0
 [ reached getOption("max.print") -- omitted 73 rows ]
```

Interpretation:

- There are 48 docs (text lines).
- Example of how to read this output table: Term ‘transparent’ is appearing once in docs 1,7,23 and so on.,

Word Cloud In R

```
# Calculate total frequency of words & creating a data frame of it
v <- sort(rowSums(m), decreasing=TRUE)
myNames <- names(v)
d <- data.frame(word=myNames, freq=v)
head(d)
```

	word	freq
questions	questions	13
happy	happy	10
subjective	subjective	8
fair	fair	7
performance	performance	6
work	work	6

```
# Create color palette
pal2 <- brewer.pal(8,"Dark2")
```

- **brewer.pal** () was developed by Cynthia Brewer. It makes the color palettes from ColorBrewer available as R palettes.
- **Arguments:**

  Number of colors included in the palette: 8

  Palette Name: 'Dark 2'
- Check out different palettes at **http://colorbrewer2.org/**

Word Cloud In R

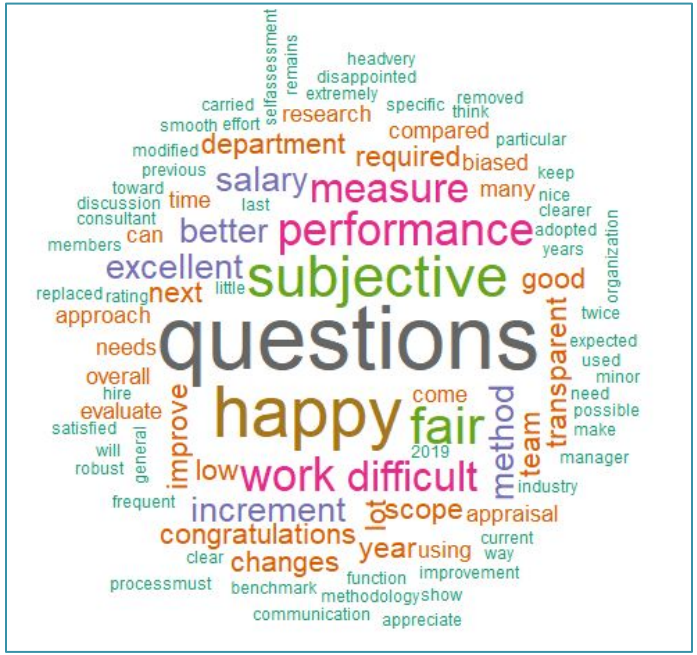
```
# Get Word Cloud

wordcloud(d$word, d$freq, random.order = FALSE , min.freq =
1,colors=pal2)
```

- First and second argument in **wordcloud()** are the words **(d$word)** and the frequency **(d$freq)** respectively.
- **random.order=FALSE** plots words in decreasing frequency. By default, plot words in random order.
- **min.freq =** words with frequency below min.freq will not be plotted.
- **colors =** color words from least to most frequent with specified color palette.

Word Cloud In R

```
# Output :
```



Interpretation:
Word questions has the largest size, indicating most frequent word followed by happy and subjective and so on..

Text Mining Using ggplot2

Plotting frequent terms as a bar plot

```
term.freq <- rowSums(m)
term.freq <- subset(term.freq, term.freq >= 5)
```

```
# Transform as a dataframe
```

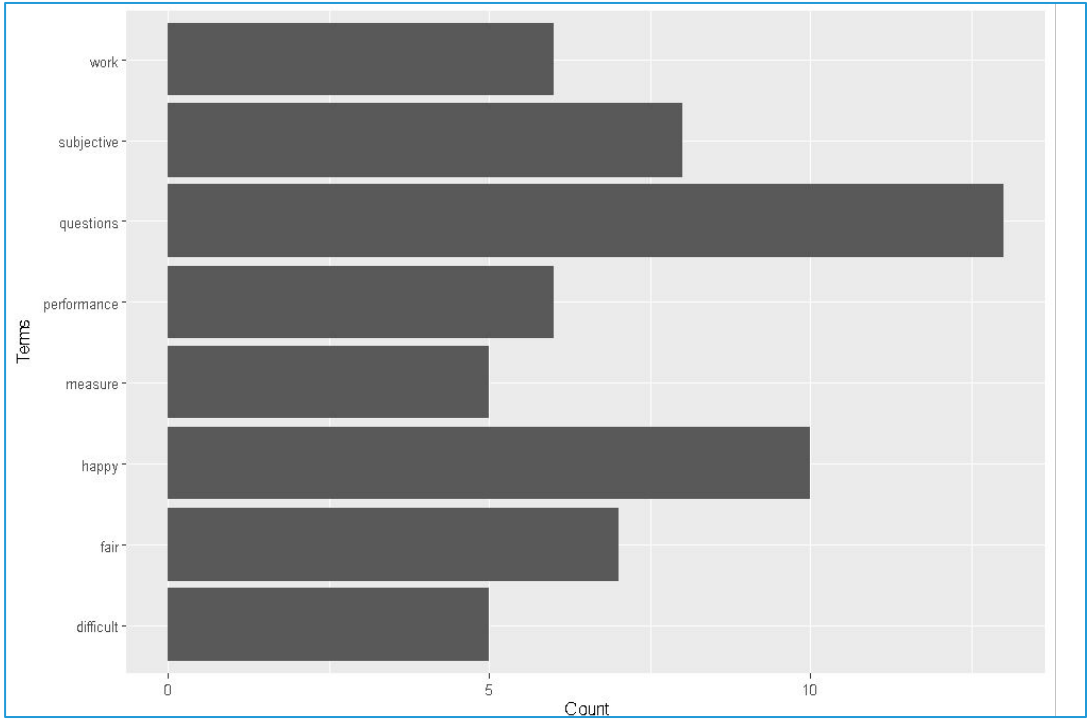
```
df <- data.frame(term = names(term.freq), freq = term.freq)
```

```
# Horizontal bar plot
```

```
install.packages("ggplot2")
library(ggplot2)
ggplot(df, aes(x = term, y = freq))+
  geom_bar(stat = "identity") +
  xlab("Terms") + ylab("Count") + coord_flip()
```

Text Mining Using ggplot2

```
# Output :
```



Interpretation:

- Graph shows the frequency of the words appearing at least 5 times on a horizontal bar graph. questions is the most frequent word with frequency more than 10.

Quick Recap

Unstructured Data	• Does not reside in traditional databases and data warehouses. • Example: emails, tweets, feedback, blogs, webpages, etc.
Text Analysis	• Process of identifying novel information from a collection of texts. (Also known as a 'Corpus')
Text mining in R	• Install **'tm'** package. Convert data into corpus. • Clean the corpus: convert all to lowercase/uppercase, remove punctuation, numbers, stopwords, words.
Word Cloud in R	• An image showing compilation of words, in which, size of words indicates its frequency or importance. • Install **'wordcloud'** package.