Text Analytics

Second IEEE India Workshop on Artificial Intelligence & Machine Learning

Prakash B. Pimpale

August 11, 2018

1 Overview

- Introduction
- Text and NLP Basics
- Text Analysis methods

1.1 Introduction

- Analytics
  - We have data available everywhere as part of our day to day processes - personal and business
  - Analysing that to find patterns/trends, to create models to predict and to classify, etc. is data analysis
  - The data can be numeric, textual or multimedia
- Text Analytics
  - The structured data is easier to analyse or at least easier to represent for the analysis - Performance of a company over the years - a tabular structure
  - The textual data is mostly unstructured and comparatively difficult to analyse - email communications, news-paper stories - words, phrases, sentences, pragraphs, documents : text structure
  - Text Analytics follows, generally, a pipeline as follows
    - Identify and retrieve documents for text analytics - read email from folders for analysis
    - Apply cleaning and pre-processing - 'may' extract 'only' the fields that are important for analysis, sender, receiver and email text - other field may also be important for some other kind of target analysis for example date, time and subject
    - Represent the text into formats needed for analysis - tokens, as POS tags, as Chunks tags, Term Document Frequencies (TDMs), etc.
    - Perform exploratory analysis - feel the data, Read certain documents manually, high frequent words, high frequent collocations, high frequent bi-grams, average word length, average document length, etc.

* Apply various text analysis techniques as per requirement : Clustering, Classification, Sentiment Aanalysis, etc.

* Example application to real world problems : Cluster e-mails (documents) into different groups, classify them into professional, personal category or IT related non-IT related, perform sentiment analysis, Named Entity Recognition, Relation Extraction, etc.
* Plot analysis results if those are to be presented

Figure 1: Text Analytics on an email Database

1.2 Natural Language Processing (NLP)

- Field intersecting Computer Science, Computational Linguistics and Artificial Intelligence
- Natural Language - Languages that humans use to communicate with humans - not C++ or Java
- Humans can listen, understand, process and reply/answer
- Doing the same with computers is NLP
- Why NLP is hard and Interesting
  - Code mixing - multilinguality
  - Ungrammatical
  - Idioms
  - Ambiguity

```
1. Films should not show crime, poverty, violence, corruption, sex, cruelty,
misery but must reflect the reality and truth... -- Morality ke Chacha
2. #Microsoft was always in d game...in fact it sets d game...get #freecharge
```

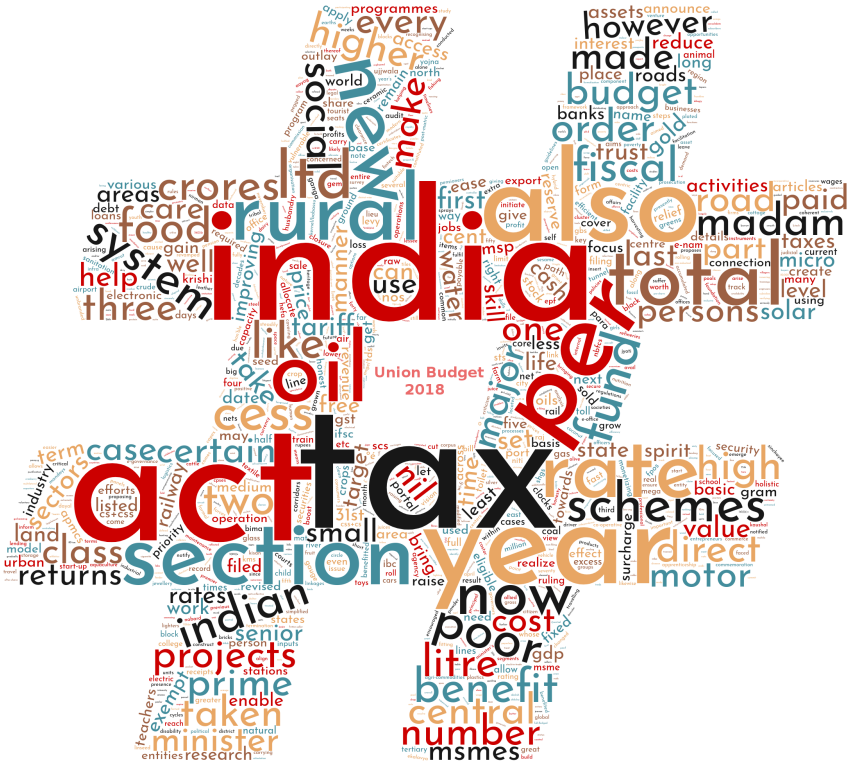

Figure 2: Word Cloud

```
in d game before it's written off by #paytm
3. Get Cold Feet.
4. I saw the man on the hill with a telescope.
5. I Lost my wallet at the bank.
```

- Text Analytics shares and gets benifitted by many NLP techniques : Rule Based and Statistical
- NLP makes the text analytics more meaningful
- An e-mail text from Enron database

```
FYI - This is the "Day 1" list that you worked on consolidated
into one spreadsheet. As you can see the Logistics Managers for
each desk, working with Suzanne and me will be responsible for
getting this done. Suzanne is putting together a master binder
and will start reviewing the requirements with each of you to
get as much done ahead of time as possible. In the master binder
will be the more detailed spreadsheets with the additional, and
critical, notes you included.

Let me know if you have any questions, tks. In addition Tammy
is working on the TPA's and EBB's to get the major EDI pipes
up and running ASAP.

Bob
```

- Just character based analytics : Suzanne, Tammy and Bob - are nothing but character sequences to a computer - S U Z A N N E, T A M M Y, B O B - almost like word 'and' which is A N D
- NLP tell us more about these words - Suzanne, Tammy and Bob are Proper Nouns and Names of Persons - they have similar charteristics (can eat, run, write speak, etc.) - and they are different from 'and' which is CC - a conjunction
- Text Analytics with NLP over sports news articles can help us identify that a string 'Sonia Mirza' is an Indian professional tennis player
- May help you to find 'more' about your dis-satisfied customers, investment firms can discover about companies and events around them
- Basic NLP Tasks
  - Non-NLP : Data Acquision, Text Extraction
  - NLP : Tokenization, Normalization, Lemmatization, Stemming, Stop word removal, Sentence Segmentation, Part of Speech tagging, chunking, parsing, ...

1.3 Text Extraction/Cleaning

- Text data can come from various sources - HTML, XML, JSON, Excel files among others

- Extracting the required text is the first task

```
<div class="last2brdiv"></div><br> Responding to TOI in an email from London, Mallya
said the ED attached his properties and the assets of United Breweries Holdings
Limited, which was a public company and not even the subject of an ED investigation.
"The assets purportedly attached under PMLA date back to several years
prior to the launch of Kingfisher Airlines...." <div class="last1brdiv">
</div><br><br> </div></div></arttextxml><strong>Stay updated on the go
with Times of India <a target="_blank"href="https://play.google.com/store/apps/details\">
News</a> App. Click <a target="_blank" href="http://get.timesofindia.com">
here</a> to download it for your device.
```

- Advertising blocks from sidebars or unwanted text needs to be neglected
- Parsing well formatted HTML/XML
- Regular Expressions
- Specific utilities like boilerpipe to remove unwanted text

1.4 Tokenization

- Text contains various linguistic units: words, punctuation, numbers, alpha-numerics

```
Modi won't go there as he is tired.
```

- Prior to analysis linguistic units need to be identified separately
- Process of segmenting these units : Tokenization

```
Modi will not go there as he is tired .
```

- A token is not mere a character sequence delimited on both sides by space
- Segmented languages - English/Hindi v/s languages like chinese and Japanese
- Issues and Challenges with tokenization
  - Plain white space tokenization can't handle "He is in I.C.U at MGM-Navi Mumbai."
  - Abbreviations need to be handled seperately. Mostly using generic and domain specific lists
  - Hyphenated words need to be dealt with care: forty-two, Mumbai-based, Pre-school, cheap air fares for Navi Mumbai-Pune return trip
  - Special cases have to be handled seperately. Indentify and unify to a standard format

```
Example Special Cases:
URLs : http://kbcs.in/datascience/courses/Course-Text-Analytics.html
Dates : 13-Dec-1967 or 13 Dece 1967
Telephone numbers : 123-456-7890 or 123 456 7890
```

- You 'may' have to develop your own tokenizers depending on the target application

1.5 Normalization

- Same entities are represented in various ways
  - Short-term courses / short term courses
  - IND / INDIA
  - I.B.M / IBM
  - Govt. of India / Indian Government
- Bringing them to a common form is Normalization
- Can be achieved by removing the periods, expanding them to a standard format, casing, etc.
- Domain dependent, carefuly crafted rules need to be written
  - You would remove periods from words with small length, upper case letters and multiple periods in it. Ex. I.B.M
  - You will expand words which are in your master list and are of special interest to you
- And then there can be variations
  - NARENDRA MODI / Narendra Modi
  - Lowecase all of them and match
  - BUT case is helpful in many cases
  - Careful rules need to be written to handle exceptions - Start of sentence, Proper nouns

1.6 Lemmatization & Stemming

- Lemmatization is reducing words to their grammatical base forms
  - am, are, is - Base form: be
  - Saw, Seeing - see
- Makes use of dictionaries and sophisticated morphological analysis techniques
  - Inflection handling - cars / car
  - Suffix and Prefix handling - pre-school / school, wood-like / wood
- Stemming also means reducing words to their base forms (stems)
  - The stems may not be always grammatical
  - It's crude chopping of suffixes
  - Most used stemmer : Portar Stemmar
  - A carefuly crafted, priority based stemming

```
cars - car
seeing - see
compressed - compress
```

```
compression - compress
compress - compress

Example RULES
sses - ss : caresses - caress
ies - i : ponies = poni
ss - ss : caress = caress
s - REMOVE : cats = cat
(*vovel*)ing - REMOVE : Talking = Talk, Sing = Sing (i.e. if the resulting stem c
```

1.7 Stop word removal

- Extremely common words add very little value to the text document : Stop Words
- Opt to remove them depending on your application
- A list of such words can be prepared, they are mostly functional words and not content words
  - Functional words : a, an, the, to
- Again be careful while removing them
  - Flights to Delhi from Mumbai are better than those from Chennai : we may lose directional information
  - If next step is sentence segmentation - removing 'The' may cause damage to the accuracy

1.8 Sentence Segmentation

- Sentence is a grammatical unit that is complete in itself

```
A set of words that is complete in itself, typically containing
a subject and predicate, conveying a statement, question, exclamation,
or command, and consisting of a main clause and sometimes one or more
subordinate clauses.
```

- Syntactically it ends with ./!/?
- ! (Exclamation Mark) and ? (Question Mark) are not as ambiguous as '.' (PERIOD)
- . can be part of sentence boundary, abbreviation (M.G.M) or a number (70.45)
- Needs a classifier to decide if it's end of a sentence or not
  - Hand written Rules
  - Regular Expressions
  - Machine Learning Based classifier
- Features for both Rule and ML based
  - Long spaces after the punctuation

  - Is it ? or !
  - Is the . a apart of abbreviation
  - Does the word after mark starts with upper case
  - Does the word that contains the mark is lower or all upper case
  - Probabilistic : Is the word with the mark frequent at the end/start of sentence

1.9 Part of Speech (POS) Tagging

- Words have classes/categories depending on the role they play in a sentence
- We learned parts of speech in school :

```
noun (Table, Shyam), verb (Run, is), adjective (beautiful),
adverb (beautifuly), pronoun (I, she, he), preposition (to, at, before, but),
conjunction (and, but, when), interjection (oh!, ouch!)
```

- why we POS tag?
  - A computer doesn't know what a 'Ramesh' means, assigning a POS will enrich information about character sequence R A M E S H
  - Establishing relationship between words become easy : Ramesh & Sania are persons and so these words may behave similarly or should be treated similarly
  - Disambiguate between words. Ex. 'I can bank upon you' and 'It's a river bank'
- Types of tags
  - Standardised and Poupular tagset for English - Penn Treebank Tagset(45), for Hindi ((LDC-IL Tagset and BIS Tagset)
  - Deeper classification of POS tags : NN and NNS (undegraduate and undergraduates), numbers(CD), Punctuations, etc.
  - Penn Tree Tagset complete list: http://www.comp.leeds.ac.uk/ccalas/tagsets/upenn.html
  - POS tags are categoriesed into
    - Closed class tags
    - Open class tags
  - Closed Class tags: Tags that have fixed elements as part of them : pronouns (He, She, They), Prepositions (at, on, to, near)
  - Open Class tags: Tags that are open to taking new elements : Nouns (Pinging), Verbs(SSHed), Adjectives, Adverbs
- Assigning these POS tags automatically is POS tagging
  - Rule based methods - dictionaries of POS tags, context based rules on previous words, next words, regular expression with specific suffixes (like 'ly')
  - Statistical Methods - Most probable POS tags for the existing 'context'
  - Context - Words surrounding the word subject to POS tagging

```
Jog   saw   a   can   .
NNP   NN    DT  NN    .
VB    VBD   DT  MD    .

"Jog/NNP" "saw/VBD" "a/DT" "can/NN"  "./."
```

- State of the art:
  - Existing POS taggers have good accuracy - more than 95%
  - 90% is bench mark, as most of the words are non-ambiguous
  - Some free POS taggers Apache OpenNLP, Stanford coreNLP, NLTK POS tagger

1.10 Chunking

- An easier form of parsing (getting complete structure of a sentence) - shallow parsing

```
Sentence:
          Narendra Modi visited New York in USA.
POS     :
          Narendra/NNP Modi/NNP visited/VBD New/NNP York/NNP in/IN USA/NNP ./.

Complete Parse:
          (ROOT
            (S
              (NP (NNP Narendra) (NNP Modi))
              (VP (VBD visited)
                (NP
                  (NP (NNP New) (NNP York))
                  (PP (IN in)
                    (NP (NNP USA)))))
              (. .)))


Chunks:
       (Narendra Modi)/NP (visited)/VP  (New York)/NP (in)/PP (USA)/NP
```

- A single word may not provide much information - Narendra and Modi independently will not mean much, but Narendra Modi is an informative phrase - a named entity
- The king of Mahishmati informative compared to the, king, of, Mahishmati
- Applications like question answering and Information Extraction can use this
- Chunking methods also achieve good accuracy i.e. above 90%

1.11 Text Representation for Analysis

- Document Term vectors
  - Incidence vectors
  - Count vectors
  - TF.IDF vectors
- Word vectors
  - Word2vec : word embedings

```
require(tm)
textV <- c("Boy, he is a handsome boy", "She is a girl",
           "He is handsome handsome", "She is beautiful")
docs <- Corpus(VectorSource(textV))
# Term Incidence vectorizer
dtm <- DocumentTermMatrix(docs, control=list(tolower=TRUE,
                                             removePunctuation=TRUE,
                                             stopwords=TRUE,
                                             removeNumbers= TRUE,
                                             weighting=weightBin,
                                             wordLengths = c(2,20)))

# Dataframe from the DTM for the training
dtmMatrix <- as.matrix(dtm)
dtmMatrix
```

```
##     Terms
## Docs beautiful boy girl handsome
##    1         0   1    0        1
##    2         0   0    1        0
##    3         0   0    0        1
##    4         1   0    0        0
```

```
#TF - Count vectorizer
dtm <- DocumentTermMatrix(docs, control=list(tolower=TRUE,
                                             removePunctuation=TRUE,
                                             stopwords=TRUE,
                                             emoveNumbers= TRUE,
                                             weighting=weightTf,
                                             wordLengths = c(2,20)))
# Dataframe from the DTM for the training
dtmMatrix <- as.matrix(dtm)
dtmMatrix
```

```
##     Terms
## Docs beautiful boy girl handsome
##    1         0   2    0        1
##    2         0   0    1        0
##    3         0   0    0        2
##    4         1   0    0        0
```

```
#TF.IDF
dtm <- DocumentTermMatrix(docs, control=list(tolower=TRUE,
                                             removePunctuation=TRUE,
                                             stopwords=TRUE,
                                             removeNumbers= TRUE,
                                             weighting=weightTfIdf,
                                             wordLengths = c(2,20)))
#stopwords("en")
#str(dtm)
# Dataframe from the DTM for the training
dtmMatrix <- as.matrix(dtm)
dtmMatrix
```

```
##     Terms
## Docs beautiful      boy girl  handsome
##    1         0 1.333333    0 0.3333333
##    2         0 0.000000    2 0.0000000
##    3         0 0.000000    0 1.0000000
##    4         2 0.000000    0 0.0000000
```

1.12 Analysing the Text

1.12.1 Document Similarity - Cosine

```
require(tm)
textV <- c("Party BJINCP won the elections",
           "Elections in Karnatak are due",
           "He acted in the movie")
docs <- Corpus(VectorSource(textV))
dtm <- DocumentTermMatrix(docs, control=list(tolower=TRUE,
                                             removePunctuation=TRUE,
                                             stopwords=TRUE,
                                             removeNumbers= TRUE,
                                             weighting=weightTf,
                                             wordLengths = c(2,20)))
# Dataframe from the DTM
```

```
dtmMatrix <- as.matrix(dtm)
#see any two rows
dtmMatrix[3,]
```

```
##     acted    bjincp       due elections  karnatak     movie     party
##         1         0         0         0         0         1         0
##       won
##         0
```

```
require(lsa)
#cosine between 1 and 2
cosine(dtmMatrix[1,], dtmMatrix[2,])
```

```
##           [,1]
## [1,] 0.2886751
```

```
#cosine between 1 and 3
cosine(dtmMatrix[1,], dtmMatrix[3,])
```

```
##      [,1]
## [1,]    0
```

1.12.2 Text Classification / Documet Classification

- Text Classification Problem
  - Organizing things into various categories
  - Numeric data : 1 2 3 4 5 6 7 8 9 10
    - 1 2 3 4 5 | 6 7 8 9 10
    - 1 3 5 7 9 | 2 4 6 8 10
    - Classified into n <= 5 and others & even and odd classes
  - Text data
    - News about movie launch, election, cricket
    - Overlapping classes : Fintech and Data Science
  - Text classification can help classify dynamic information on the fly - emails, customer queries, social media posts, news articles
  - Helpful in many tasks other than just organizing
  - Techniques we learn can help classify other kind of data also, but we focus on text
- Methods- Rule Based Classification
  - IF-THEN rules are used to classify
  - Spam : IfcontainsWords (Money, lottary, Cheap loan, A Special Gift Waiting For you, click here, XXX)

  - Tedious, expensive and risky - But unavoidable and wise to start with when there is no training data!

- Methods- Machine Learning Based
  - Machine learning methods like Naive Bayes
  - Artificial Neural Networks (ANN)
  - Support Vector Machines
  - Decision Trees
  - The tokens are considered as features
  - New features can be engineered by use of stemming, lemmatization, POS tagging, Chunk information, N-grams, etc.

1.12.3 Sentiment Analysis

- Sentiment Mining/Opinion Mining or Analysis indentifies sentiment/mood of the text - Positive text or negative text
- A movie review/product review is opnion of the person for a movie/product - that can be positive or negative - identifying this category is sentiment analysis
- So given an e-mail we can classify if it contains positive, negative or neutral vibes - which can then be used to label relationship of involved people as positive or negative
- Primarily a classification task - can be solved with rules or a learning based approach
- Some rule based use bag of positive and negative words classify a text as positive or negative - lists available online
- Pre-trained Machine Learning based sentiment extractor - http://nlp.stanford.edu/sentiment/

1.12.4 Text Clustering

- Clusters given document into various groups based on the features
- For example of e-mails, we can cluster users based on word they use
- User-terms feature can be built

```
        T1 T2 T3 T4...
User1   1  0  0   1
User2   1  1  1   0
.
```

- e-mails Can be clusterd into groups using same technique
- Document term matrix

```
     T1 T2 T3 T4...
Email1   1  0  0   1
Email2   1  1  1   0
.
```

- Not just terms but other features can also be used for clustering
  - A person's organization
  - Average length of his emails ??
  - Number of non-functional words

1.13 Some Advice

- Consider lowercasing everything while matching
- 60% of your time will actually go into making your data analysis ready - classification/sentiment/clustering - that's ok!
- Regular Expressions are a very handy tool
- To create training data, if you want to use machine learning, bootstrap - create some data - train - classify and correct classification errors - retrain
- Explore your data, do things by manually reading it before you implement something

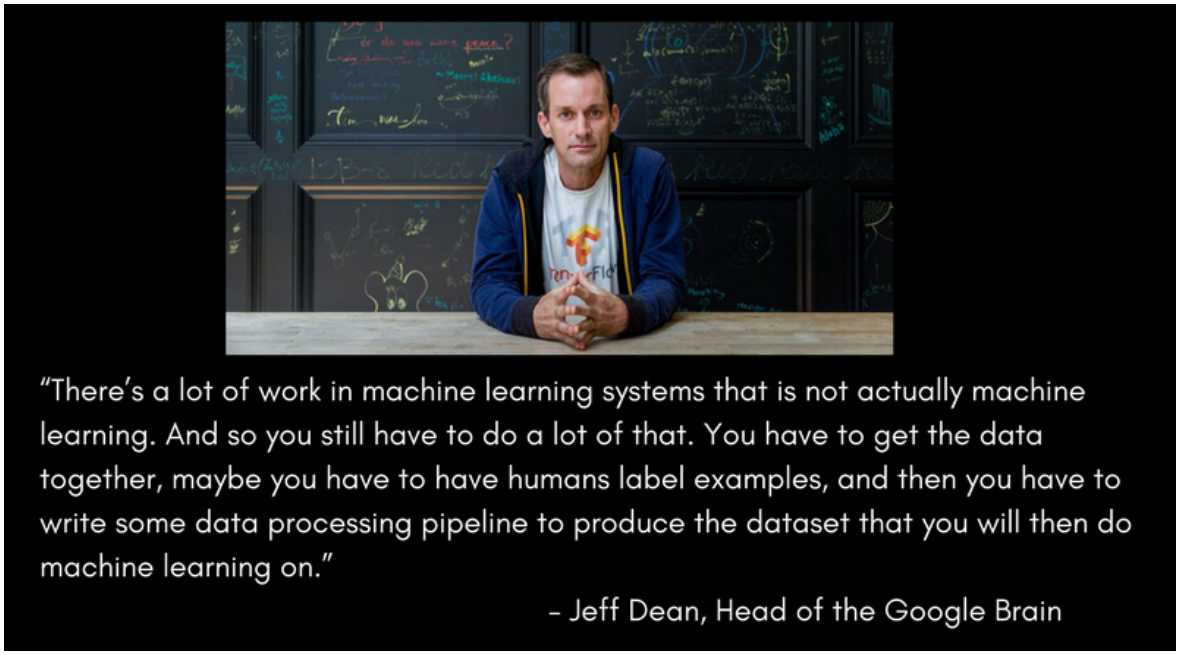

Figure 3: Advice from the Jeff - Mostly true with the Text

1.14 Thank you

- We can be contacted for AI/NLP/ML assignments at
  - kbcs@cdac.in
  - prakash@cdac.in
  - sasi@cdac.in

- Call us at 022-27565303 Ext. 270
- Visit us at www.kbcs.in/datascience
- Or at https://www.cdac.in/index.aspx?id=edu_ctp_DataScience