e-Mail Task - Text Analytics & NLP

Prakash B. Pimpale

Aug 5, 2016

1 Contents

1.1 Overview

- Analytics, Text Analytics and NLP
- Regular Expressions
- Basic NLP Processes
- Exploratory Text Analysis
- Categorization, Sentiment Analysis and Clustering
- References

1.2 Analytics, Text Analytics and NLP

1.2.1 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

1.2.2 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. - Cluster e-mails (documents) into different groups, classify them into professional, personal category or IT related non-IT related, perform sentiment analysis, Names Entity Recognition, Relation Extraction, etc.

1.2.3 NLP

- Where does NLP come into picture ?
  - Sample e-mail text

```
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
```

- Suzanne, Tammy and Bob - are nothing but a character sequence to a computer - S U Z A N N E, T A M M Y - almost like word ‘and’ which is A N D
- NLP tell us more about these words - Suzanne, Tammy and Bob are NNP and Names of Persons - they have similar charteristics - and they are different from ‘and’ which is CC - a conjunction
- e-Mail analysis for contacts task isn’t just a text analytics task
  - You may establish strength of a relationship between two persons by number of e-mails exchanged - numeric analysis
  - You may establish strength of a relationship between two persons from single e-mail where the one appreciate how the other one is close to him and how his help was important in the past - Text analysis
- To analyse such data we need combination of techniques - regular expression patterns, rules, NLP, Machine Learning, etc.

1.3 Regular Expressions

- A way to search, edit and manipulate data
- Senders, receivers, date time, text, etc. can be extracted from the email using regex

```
void RegExExtractEmail(String email){
      //Define pattern
      String fromEmailPattern =
      "From: (([a-zA-Z0-9_\\-\\.]+)@([a-zA-Z0-9_\\-\\.]+)\\.([a-zA-Z]{2,5}))";
      Pattern p = Pattern.compile(fromEmailPattern);

      //Get a matcher for our email text
             Matcher m = p.matcher(email);

       //Extract sender information
       if (m.find()) {
           //getting groups
           System.out.println("0th Matching Pattern: "+m.group(0));
           System.out.println("1st email ID: "+m.group(1));
           System.out.println("2nd Person ID/Name: "+m.group(2));
           System.out.println("3rd: Organization: "+m.group(3));
       }
    }

OUTPUT:
0th Matching Pattern: From: julie.armstrong@enron.com
1st email ID: julie.armstrong@enron.com
2nd Person ID/Name: julie.armstrong
3rd: Organization: enron
```

- Can also be used to remove certain characters sequences like HTML tags
- Can also be used to extract the phone numbers, date, time, etc.
- For details refer: https://docs.oracle.com/javase/tutorial/essential/regex/

1.4 Basic NLP Processes

1.4.1 Sentence Segmentation

- Sentence is a gramatical 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 point) and ? (Question Mark) are not as ambiguous as '.' (PERIOD) for sentence segmentation
- . 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 sentence or not
  - Hand written Rules
  - Regular Expressions
  - Machine Learning Based classifier
- Features can be - for both Rule and ML based
  - Long spaces after the punctuation
  - Is it ? or !
  - Is the . a apart of abbreviation
  - Does the word after . starts with upper case
  - Does the word that contains . is small or all upper case
  - Probabilistic : Is the word with . a frequent at the end/start of sentence
- There are multiple open source sentence segmentation utilities are avaialble, you may not need to implement this

1.4.2 Tokenization

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

```
fill the Ercot load from the wholesale/retail markets starting
Gilbert-smith, Doug; Forney, John M.; May, Tom; Hetrick, Nancy; Twiggs, Thane
```

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

```
fill the Ercot load from the wholesale / retail markets starting
Gilbert-smith , Doug ; Forney , John M . ; May , Tom ; Hetrick , Nancy ; Twiggs , Thane
```

- A token is not mere a character sequence delimited on both sides by space
- Token must be linguistically significant and useful for our application
- Issues and Challenges with tokenization
  - Plain white space tokenization can't handle "He is in I.C.U at MGM-Navi Mumbai." - ICU is one token and MGM and Navi Mumbai are different tokens - If your task demands very accurate tokenization you may have to make list of 'important' tokens in your domain
  - 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 - you want to seperate that or keep together is your decision

  - Special cases have to be handled seperately. Indentify and unify to a standard format - URLs, Dates, Telephone numbers

```
URLs : http://kbcs.in/datascience/courses/Course-Text-Analytics.html
Dates : 13-Dec-1967 or 13/12/1967
Telephone numbers : 123-456-7890 or (123)-456-7890
```

- But most of the time pre-trained and freely available tokenizers will be sufficient

1.4.3 Part of Speech (POS) Tagging

- Words have classes/categories depending on the role they play in a sentence
- We learn part of speech in schools :

```
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
- Types of tags
  - Standardised and Popular tagset for English - Penn Treebank Tagset(45)
  - Deeper classification of POS tags : NN and NNS : undegraduate and undergraduates for different form of the words singular, plural- numbers, Punctuations, etc.
  - Penn Tree Tagset complete list: http://www.comp.leeds.ac.uk/ccalas/tagsets/upenn.html
  - POS tags categoriesed into : Closed Class tags and 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 - the words surrounding 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 names Apache OpenNLP, Stanford coreNLP, NLTK POS tagger, SyntaxNet

1.4.4 Chunking

- An easier form of parsing (getting complete structure of 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 iformation - Narendra and Modi independently will not mean much, but Narendra Modi is an informative phrase - a named entity
- Applications like question answering and Information Extraction can use this
- Chunking methods also achieve good accuracy i.e. above 90%

1.4.5 Named Entity Recognition

- Text content mentions people, places, events, currencie, dates, etc.

- These people, places and the other entities may be of interest to us
- Named Entity Recognition helps identify such entities given the text document
- Pre-trained NERs can be used or your own NER can also be developed using patterns and dictionaries

1.4.5.1 Methods For NER

- Basically a classification problem
- Tasks almost similar to POS tagging and Chunking and used as pre-processing step most of the time

```
Mr. Narayan Murthy went to Banglore.

Start Not Person Person Not Not Place End
```

- Most working methods are combination of Rules and Machine Learning
- General Pipeline - KB, Extraction and classification
- Build a Knowledge Base (KB) : List of known entities
  - Difficult task as to be done manually
  - Semi automatic frequency based ways can be found
  - The entities may keep changing/adding
  - An entity may belong to more than one categoy - baby names on cities, cars - Mahindra XUV or companied named after people
  - But Entities like country names, State names, etc. can be maintained
  - Domain specific entities can be added
- Extraction : Extraction of Possible Entities
  - Extract words or group of words that can be entities
  - Use of tokenization, sentence segmentation, POS tagging, chunking
  - Normalization of entities - PM and Prime Minister
- Classification into type
  - Types such as Person, Location, Organzation, etc. Or NIL (Not an Entity)
  - Rule Based classification
  - Machine Learning Based Classification

1.4.5.2 Features for NER

- Feature set can be used both for the Rule based and ML based
- Currencies, Numeric values and Dates are extracted with Regular expressions - easier task
- People, Location and Organization names are most difficult
- For locations - New, Upper, Lower before a word are good indicators

- For names - Mr, Dr, Miss, General, President before a word are good indicators
- Surrounding words
  - Unigrams, Bigrams, trigrams - as per requirement
- POS Tags
- Word capitalization
- First character capitalization
- If the word is surrounded by quotes
- Is there a hyphen in the word
- Is the word present in our gazetteer list
- But these may not work for socila network data - Twitter, Facebook, Whatsapp, etc.
- Suffixes and Prefixes

1.4.6 Relation Extraction

- The relation extraction finds out the relation between extracted entities
- The relations can be of various types workFor(Prakash, C-DAC), IsIn(C-DAC, Mumbai)
- Can be built using regular expressions and frequency of occurences of those patterns
- Pre-trained relation extractors for English are available for certain relationships
- Stanford Relation Extractor works pretty well and is in Java : http://nlp.stanford.edu/software/relationExtractor.html

1.4.7 Normalization

- Same entities may be represented in various ways and bringing them to common form is normalization
  - Short-term courses / short term courses
  - IND / INDIA
  - I.B.M / IBM
  - Govt. of India / Indian Government
- Removing the periods, expanding them to a standard format are some solutions to normalization
- 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
  - You will expand words which are in your list/dictionary 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 : Application dependent

1.4.8 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

1.4.9 Stop word removal

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

1.4.10 Representation

- Term based representations
  - TF-IDF is measure of importance, of a word to a document
  - Term Frequency (TF) is number of times the term occures in a document or email
  - Inverse Document Frequency (IDF) is a measure of how common a term is across the documents - the, a, an, songs, fiction
  - IDF for 'a term' is calculated as - Total no. of documents/No. of documents with 'the term'
  - TF-IDF is a product of the two measures = TF*IDF for a term in a document - how native a terms is to the document
- Document Term Matrix : Frequencies

```
        T1  T2 T3 T4...
Email1  11  7  0  1
Email2   0  3  25 18
.
.
.
```

- Document Term Matrix : Binary

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

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

1.5 Exploratory Text Analysis

- We can explore the given e-mail text to know about the data
- The exploratory text analysis helps to find what is major content of the data, what particular analysis we can perform ahead
- Involves
  - Finding high frequent words from a person's incoming e-mails, outgoing e-mails - Top few words will tell you what he is into
  - You can have a look at high frequent n-gram ('high frequent' in this sentence is a bi-gram) to know more about the person or organization
  - High frequent collocations ('High' and 'frequent' are appearing together a lot of times)
  - High frequent co-occurences in same sentence may tell something about their relationship
  - Find average email length of a person - shorter/longer emils may mean something
  - Average word length used - may mean how informative text he mostly writes - long length words represent more information

1.6 Text Classification

- Categorizing given text document into different classes - Text Classification
- Spam (v/s ham) filtering is one of the very useful applications of text classification

- Gender Identification - from writing style of the author - may be useful in our task
- Problem formultion

```
Assigning a document D to one of the classes from set C1, C2,...Cn
```

1.6.1 Technique for Classification - Rule Based Classification

- Write the if-then rules
- Spam : containsWords (Money, lottary, Cheap loan, A Special Gift Waiting For you, click here)
- Bag of words with classifictory power for every categoy - If a Document contains more than threashold % of those - classify Postive
- Gender identification from names - Various frequency based rules - volvel ending female and non ending male, or based on stats of certain suffixes
- For our case we can classify an e-mail into inside-enron and out-side enron category
- Rule base gives High accuracy for identified cases, but possibility of huge rejections
- But tedious, expensive and risky - But unavoidable and wise to start with when there is no training data!

1.6.2 Technique for Classification - Machine Learning

- Machine Learning

```
Field of study that gives computers the ability to
learn without being explicitly programmed.
```

- Multiple algorithm implementations are already available - WEKA, MALLET, NLTK, other Python libraries
  - Baysian Classifier
  - Artificial Neural Networks
  - Support Vector Machines
  - Decision Trees
- Need the training data i.e. pre-classified emails into different classes
- For Email Task the examples can be classification of an email into a personal-email or professional email - Not everybody has labeled emails as personal
- Features for machine learning is an important and necessary aspect
- Terms are default good features for text classification
- Designing additional features which are suited to a specific problem may help improve performance

- Feature Engineering - Representing data with engineered features
  - Stemming, lammetization of words
  - Special weightage to features in specific zones - Title, first paragraph, last paragraph
- POS tags and related grammatical information
- Effect of any feature engineering needs to be evaluated based on results
- Example features for Spam Filter
  - Source domain of email - A Metadata - mostly from .com than .edu
  - Is the sender know or unknown
  - Number of nonalphanumeric characters in email text - Spams have high
  - Location of 'feature words' - are words like free, drug in subject or text
  - Number of e-mail recipients
- You can come up with similar features for categories you may decide

1.7 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/

1.8 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 emails before you implement something

1.9 References

- Free book : NLP with Python - http://www.nltk.org/book_1ed/
- Free book : Introduction to Information Retrieval - http://nlp.stanford.edu/IR-book/
- Speech and langauge Processing, Daniel Jurafsky and James H. Martin
- Stanford core : http://stanfordnlp.github.io/CoreNLP/
- Core NLP API Documentation: http://stanfordnlp.github.io/CoreNLP/api.html
- SyntaxNet by Google : https://research.googleblog.com/2016/05/announcing-syntaxnet-worlds-most.html
  - Download : https://github.com/tensorflow/models/tree/master/syntaxnet

1.10 About C-DAC Mumbai

Centre for Development of Advanced Computing (C-DAC) is the premier R&D organization of the Department of Electronics and Information Technology (DeitY), Ministry of Communications & Information Technology (MCIT) for carrying out R&D in IT, Electronics and associated areas.

KBCS division at C-DAC Mumbai is involved in research and development of product and technologies in the area of Natural Language Processing, Machine Translation and Transliteration for Indian languages, Machine Learning, Text to Speech systems, Automatic Speech recognition systems, Data Science, among others.

We are located in Juhu, Mumbai and Kharghar, Navi Mumbai.

Know more about us at http://www.kbcs.in/