Artificial Intelligence

Module 7 NLP

Rajesh Kumar,
VIT Chennai

Contents

- Language models
- Model evaluation
- Text classification
- Information retrieval
- Page- Rank Algorithm
- Information extraction

Language models

Natural Language Processing

- A branch of artificial intelligence
- helps computers
  - Understand,
  - Interpret and
  - manipulate human language in text, audio
- It is Interdisciplinary
  - Computer Science,
  - Databases, Information Science
  - Mathematics , Statistics

Natural Language Processing

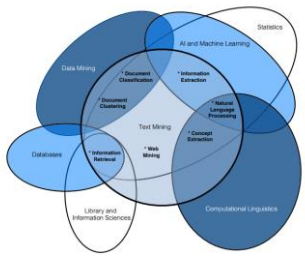
- **Sentiment analysis**
- Identifying
  - the mood
  - Subjective opinions
  - Average sentiment, opinion.
- Context extraction
  - Pull structured information.
- **Content categorization:**
  - **Topic discovery**
  - Summary based on language
  - Detect duplication
  - Alerts
  - Indexing
  - Searching
- Modelling
  - Capture the meaning and theme in text data
  - Optimization
  - Forecasting

Natural Language Processing

- Alexa
- Google voice
- SIRI
- Banking needs
- Information retrieval

- Text to speech
- Speech to text
- Document summarization
- Machine translation
  - Difference with others?? Text to speech

Natural Language Processing



- One need to
  - Collect,
  - Manage,
  - Store,
  - Distribute,
  - Analyze,
  - Visualize
  - Reuse Data

Natural Language Processing

- Basic NLP tasks are
  - Tokenization
  - Stop words removal
  - Lemmatization/Stemming
- RBS analyzes customer feedback data from
  - Emails,
  - Surveys And
  - Call Center Conversations
  - To Identify the root cause of customer dissatisfaction and
  - To implement improvements

Natural Language Processing

- Derive great value from text data
  - By use of
  - Linguistic
  - Algorithm
    - By
    - Transforming
    - Enriching data(text)
- Many techniques - A broad array of approaches based on application
- Statistical
- Machine learning methods to
- Rules-based
- Algorithmic approaches.
- The text and voice-based techniques and data varies widely

Natural Language Processing

- Small data – No information
- Big data – computation cost
- Metadata
  - Data about data
  - It is required
    - to use data effectively,
    - to reuse data,
    - to share data
    - to integrate data
  - Variable names and
  - Labels, value labels,
  - Information on who collected the data, when, by what methods, in which locations, for purpose behind collection

Morphology

- Free morphemes
  - Used as a word
  - Combines with other morphemes
    - The, or, in, of
- Bound morphemes
  - Not used as a word
  - Quickly, Walked, Dogs
- Allmorphs
  - Variants of morphemes
  - Can not be replaced by other
  - Unhappy, Impossible
  - Irrational, Incomprehensible
- Study of the internal structure of the words
- Morphemes –
  - How words are built up from smaller meaningful units
  - Plurals – from noun
    - DOGS, CATS
  - Unladylike – 3 morphemes
    - UN - not
    - Lady - women
    - Like – character of, type of

Stem and Affixes

- Affixes – Bound morphemes
- Bits and pieces adhering to stems to change their
  - Meanings
  - Grammatical functions
- Types of Affixes
- Prefix
  - Unhappy, PRE existing
- Suffiix
  - Talk ing
  - Quick ly
- Infix – Philippines, Sanskrit
- Circumfix – Dutch (ge-berga-te)
- Root - core meaning
- Free morphemes

Morphology

- Derivational morphology
  - Creates new words by changing part-of-speech
  - Verb to noun
  - Drive → Driver
  - logic, logician, illogicality, logical, illogical
- Inflectional morphology
  - Grammatical: number, tense, case, gender
  - It creates the new form of the same word
    - bring, brought, brings, bringing

NLP module in python

```
import nltk
from nltk.stem import PorterStemmer
word_stemmer = PorterStemmer()
word_stemmer.stem('writing')
```


Word Formation

- Can be used for Machine translation
- The words cannot be converted it into equivalence language words
- Compounding will help to convert
  - Room temperature

  (Translate to Hindi)

- **Compounding**
  - Words formed by combining two or more words
  - It can be various parts-of-speech
    - Adj + Adj - Adj : bitter – sweet
    - N + N - N: rain – bow
    - V + N - V: pick – pocket
    - P + V - V: over – do

Morphology

- Clipping
- Longer words are shortened
  - Doctor → Doc
  - Laboratory → Lab
  - Advertisement → Advertise
  - Dormitory → Dorm
  - Examination → Exam
  - Refrigerator → Fridge
- Blending
  - Parts of two different words are combined
  - breakfast + lunch - brunch
  - smoke + fog smog
  - motor + hotel motel

Lemmatization

- Derivational morphology
  - Creates new words by changing part-of-speech
  - Verb to noun
  - Drive → Driver
  - logic, logician, illogicality, logical, illogical
- word → lemma
  - saw {see, saw}
- What is the root word?
- Word → set of (lemma + tag)
  - saw → < see, verb.past >
    → < saw, noun.sng > singular

Morphological Analysis

Input	Morphological parsed output				
Cats	Cat	+	N	+	PL
Cat	Cat	+	N	+	SG
cities	city	+	N	+	PL
geese	goose	+	N	+	PL
goose	goose	+	N	+	SG
caught	catch	+	V	+	PAST-PART
merging	merge	+	V	+	PRES-PART

Morphological Analysis

- Objective of morphological analysis
  - To take input forms like those in the first column
  - Produce output forms like those in the second column
  - Output contains stem and additional information
  - + N for Noun
    + SG for Singular
    + PL for Plural
    + V for Verb

Challenges Morphological Analysis

- Knowledge of stems / roots is required to solve such problems
  - duck is possible root, not ducks
  - A dictionary (lexicon) of root / stem is required
- Morphotactics
  - The class of morphemes follow other classes of morphemes inside the word.
    - Plural morpheme follows the noun
  - Few endings go on with few words
    do + er : ok
    be + er : not so (can't appear)

- boy → boys
  fly → flys => flies [y → i rule]
- toiling → toil (work hard extreme)
  duckling → duck ? (young duck)
- getter → get + er
  doer → do + er
  beer → be + er

Challenges Morphological Analysis

FSM

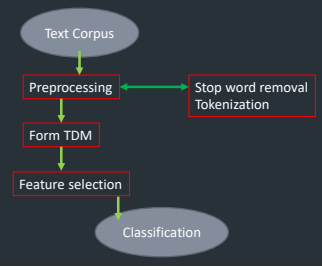
- Spelling change rules using spelling change rules
  get + er → getter
- English
  - 90196 lexicon entries makes 317477 forms (1: 3.5)
- Sanskrit
  - 170000 lexicon entries makes 11 million forms (1:64.7)

Text classification

- **Knowledge distillation phase**
  - Deduces patterns or knowledge from the intermediate form.
  - The intermediate forms are of varying degrees of complexity
    -

- Two phase
  - **Text preprocessing phase**
  - Transforms free-form text documents into an intermediate form i.e. each text document is represented as a vector of words,
  - It is typically done in the popular vector representation for information retrieval.

Text Document classification



DTM - Document Term Matrix

- DTM

Documents	Term			
D1	t1	t2	…	tn
D2				
D3				
Dm				

- Text preprocessing primarily concerns conversion of unstructured data into structured data
  - **Tokenization,**
  - **Stop words removal,**
  - **Stemming**
  - After completion of the preprocessing step, each unique word forms
    - A token or
    - A feature in the data.
- DTM is Based on these features
  - rows corresponds to documents in the collection and columns represents to terms.

DTM …

- There are different ways of constructing the DTMs.
  - **value** in the cells can be determined in many different ways.
  - Binary Document Term Matrix
  - Term Frequency
  - TF-IDF

Binary Document Term Matrix

Documents	Term				Result	statistics	Analysis
	t1	t2	...	tn			
D1	0	1		0	0	1	0
D2							
D3							
Dm							

- Each row represents a document and each column represents a distinct word or feature
  - An entry of '1'in the column denotes the presence of the word in the document
  - '0' denotes its absence.
  - A particular word occur or not in that document

Term Frequency

- Term Frequency
- Inverse term frequency

Documents	Term			
D1	t1	t2	...	tn
D2	2	3		7
D3	0	4		1
Dm	2	0		5

- Count of terms in document are filled in
- TF-IDF

$$\omega_{ij} = tf_{i,j} * \log\frac{N}{df_i}$$

N = # Total document

df_i = # Number of words in the document

$tf_{i,j}$ = # Number of terms in document

ω_{ij} = tf-idf score

DTM …

- DTM

Documents	Term			
D1	t1	t2	...	tn
D2				
D3				
Dm				

- There are different ways of constructing the DTMs.
  - **value** in the cells can be determined in many different ways.
  - Binary Document Term Matrix
  - Term Frequency
  - TF-IDF

Types of Modelling/ Learning

- Unsupervised learning
- No outcome (yet) to model is not known.
- Data is not labelled
- Clustering of cases
  - Types of customers
- Association Rule Mining
  - Clusters of actions by cases
  - Groups of products purchased together

- Supervised Learning:
  - You have some data where the outcome is already known. Labelled
  - Methods focus on recovering that outcome and
    - Prediction to new outcomes
  - Classification Problems
    - Regression
    - Logistic Regression
    - SVM
    - Decision tree

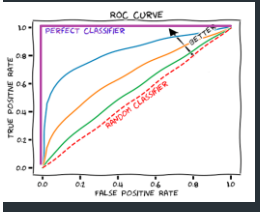
Modelling Evaluation

		Condition Phase (Worst Case) Condition Positive/ Shaded	Condition Negative/ Unshaded	
Testing Phase (Best Case)	Test Positive/ Shaded	True positive shaded T_p *(Correct)*	False positive shaded F_p *(Incorrect)*	Precision/Positive Predictive Value (PPV) $\frac{T_p}{T_p+F_p} \times 100\%$
	Test Negative/ Unshaded	False negative unshaded F_n *(Incorrect)*	True negative unshaded T_n *(Correct)*	Negative Predictive Value (NPV) $\frac{T_n}{T_n+F_n} \times 100$
		Sensitivity/Recall Rate (RR) $\frac{T_p}{T_p+F_n} \times 100\%$	Specificity Rate (SR) $\frac{T_n}{T_n+F_p} \times 100\%$	

- Training data
  - Validation data
- Testing data
- Overfitting
- TP – True Positive
- TN – True Negative
- FP – False Positive
- FN – False Negative
- Confusion matrix

Image from https://i.stack.imgur.com/tApzI.png

Performance Evaluation



- F1-Score =

2* Precision * Recall/ (Precision + Recall)

- AUC = Area under curve => 0 – 1

(Sensitivity + Specificity) *.5

Value closure to 1 is better

- MSE = [sum of (Actual – Predicted)**2]/n
- MAPE – Mean absolute average error

= [sum of [Actual – Predicted / Actual]] *100/n

Page Rank Algorithm

- A system for ranking web pages
- A link from page A to page B as a vote, by page A, for page B.
  - The weight of vote depends on the page
- Determination of the importance of a webpages based on link structure
- A probability distribution to represent the likelihood that
  - A person randomly clicking on links will arrive at any particular page
- Organize the information on web and make it universally accessible and useful.

Page Rank Algorithm

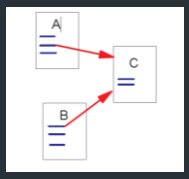
- Algorithm does not rank the whole website,
  - It is determined for each page individually.
  - Page A is recursively defined by the PageRank™ of those pages which link to page A

PR(A) = (1-d) + d (PR(T1)/C(T1) + ... + +PR(Tn)/C(Tn))

Where:

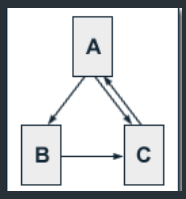
- PR(A) is the PageRank of page A,
- PR(Ti) is the PageRank of pages Ti which link to page A
- C(Ti) is the number of outbound links on page Ti
- d damping factor 0 - 1

Page Rank Algorithm



- Backlinks and Forward links:
  - A and B are C's backlinks
  - C is A and B's forward link

Page Rank Algorithm



- A small web consisting of three pages A, B and C
- PR(A) = 0.5 + 0.5 PR(C)
- PR(B) = 0.5 + 0.5 (PR(A) / 2)
- PR(C) = 0.5 + 0.5 (PR(A) / 2 + PR(B))
- We get the following PageRank™ values for the single pages:
- PR(A) = 1.07692308
- PR(B) = 0.76923077
- PR(C) = 1.15384615

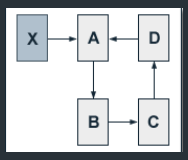
Iterative computation

Iteration	PR(A)	PR(B)	PR(C)
0	1	1	1
1	1	0.75	1.25
2	1.125	0.75	1.125
3	1.0625	0.78125	1.15625
4	1.078125	0.765625	1.15625
5	1.078125	0.76953125	1.15234375
6	1.076171875	0.76953125	1.154296875
7	1.077148438	0.769042969	1.153808594
8	1.076904297	0.769287109	1.153808594
9	1.076904297	0.769226074	1.153869629
10	1.076934814	0.769226074	1.153839111
11	1.076919556	0.769233704	1.153846741
12	1.07692337	0.769229889	1.153846741
13	1.07692337	0.769230843	1.153845787
14	1.076922894	0.769230843	1.153846264
15	1.076923132	0.769230723	1.153846145

The Random Surfer Model

- The probability for the random surfer reaching one page is
  - The sum of probabilities for the random surfer following links to the page
- The surfer gets bored sometimes and click some random link is modelled by damping factor d(0-1).
- It as a model of user behaviour,
  - A surfer clicks on links at random with no regard towards content.
  - The random surfer visits a web page with a certain probability
  - The probability that the random surfer clicks on one link is solely given by the number of links on that page.
  - one page's PageRank is divided by the number of links on the page.

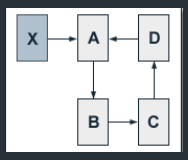
Inbound link importance



A constant Pagerank PR(X) = 10, d=.5

PR(A) = 0.5 + 0.5 (PR(X) + PR(D)) = 5.5 + 0.5 PR(D)
PR(B) = 0.5 + 0.5 PR(A)
PR(C) = 0.5 + 0.5 PR(B)
PR(D) = 0.5 + 0.5 PR(C)
The PageRank Value
PR(A) = 19/3 = 6.33
PR(B) = 11/3 = 3.67
PR(C) = 7/3 = 2.33
PR(D) = 5/3 = 1.67

Inbound link importance



A constant Pagerank PR(X) = 10, **d=.75**
PR(A) = 0.25 + 0.75 (PR(X) + PR(D)) = 7.75 + 0.75 PR(D)
PR(B) = 0.25 + 0.75 PR(A)
PR(C) = 0.25 + 0.75 PR(B)
PR(D) = 0.25 + 0.75 PR(C)
The results gives us the following PageRank values:
PR(A) = 419/35 = 11.97
PR(B) = 323/35 = 9.23
PR(C) = 251/35 = 7.17
PR(D) = 197/35 = 5.63
d × PR(X) / C(X) = 0.75 × 10 / 1 = 7.5

Is raising PAGE RANK possible?

- The probability for the random surfer reaching one page is
  - The sum of probabilities for the random surfer following links to the page
- The surfer gets bored sometimes and click some random link is modelled by damping factor d(0-1).

- Add new pages to your website
  - as many as you can
- Swap links with websites which have high PageRank™ value
- Raise the number of inbound links
  - Advertise your website on other sites