Natural Language Processing(NLP)

Manisha V Jadhav

What is NLP?

"Natural language processing (NLP) is a field of computer science, artificial intelligence (also called machine learning), and linguistics concerned with the interactions between computers and human (natural) languages. Specifically, the process of a computer extracting meaningful information from natural language input and/or producing natural language output "

- Wikipedia

What is natural language processing?

- Natural language processing, often abbreviated as NLP, refers to the ability of a computer to understand human speech as it is spoken. NLP is a key component of artificial intelligence (AI) and relies on machine learning, a specific type of AI that analyzes and makes use of patterns in data to improve a program's understanding of speech.

Text Mining vs NLP

NLP (Natural Language Processing)	Text Mining or Text Analytics
Automated Speech	Automated Grouping (n grams approach)
Automated Writing	Automated Classification (bag of words)
Automated Translation	Pattern Discovery

Open Source NLP Libraries

- Apache OpenNLP
- Natural Language Toolkit (NLTK)
- Standford NLP
- MALLET

Feature Availability

Feature	Spacy	NLTK	Core NLP
Easy installation	Y	Y	Y
Python API	Y	Y	N
Multi Language support	N	Y	Y
Tokenization	Y	Y	Y
Part-of-speech tagging	Y	Y	Y
Sentence segmentation	Y	Y	Y
Dependency parsing	Y	N	Y
Entity Recognition	Y	Y	Y
Integrated word vectors	Y	N	N
Sentiment analysis	Y	Y	Y
Coreference resolution	N	N	Y

Different NLP tasks

- **Sentence segmentation, part-of-speech tagging, and parsing**
- **Deep analytics**
- **Machine translation**
- **Named entity extraction**
- **Co-reference resolution**
- **Automatic summarization**
- **Sentiment analysis**

Scope of discussion

- Language of focus :- English
- Domain of Natural Language Processing to be discussed.
  Text linguistics
- Focus on statistical methods.

Why NLP ?

Answering Questions

- ”What time is the next bus from the city after the 5:00 pm bus ?”
- ”I am a 3rd year CSE student, which classes do I have today ?”
- ”Which gene is associated with Diabetes ?”
- ”Who is Donald Knuth ?”

Information extraction

Extraction of meaning from email :-

We have decided to meet tomorrow at 10:00am in the lab.

We have decided to meet tomorrow at 10:00am in the lab.

To do : meeting
Time : 10:00 am, 22/3/2012
Venue : Lab

Machine Translation

मेरा नाम रजत है | => My name is Rajat.

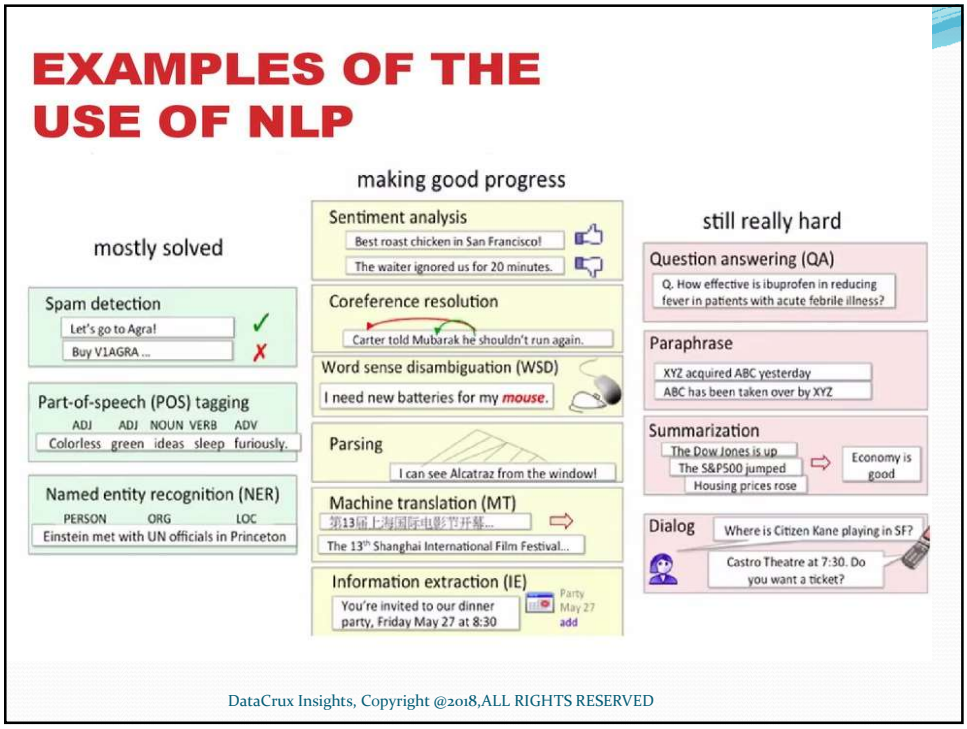
Grass is greener on the other side. => दूर के ढोल सुहावने |

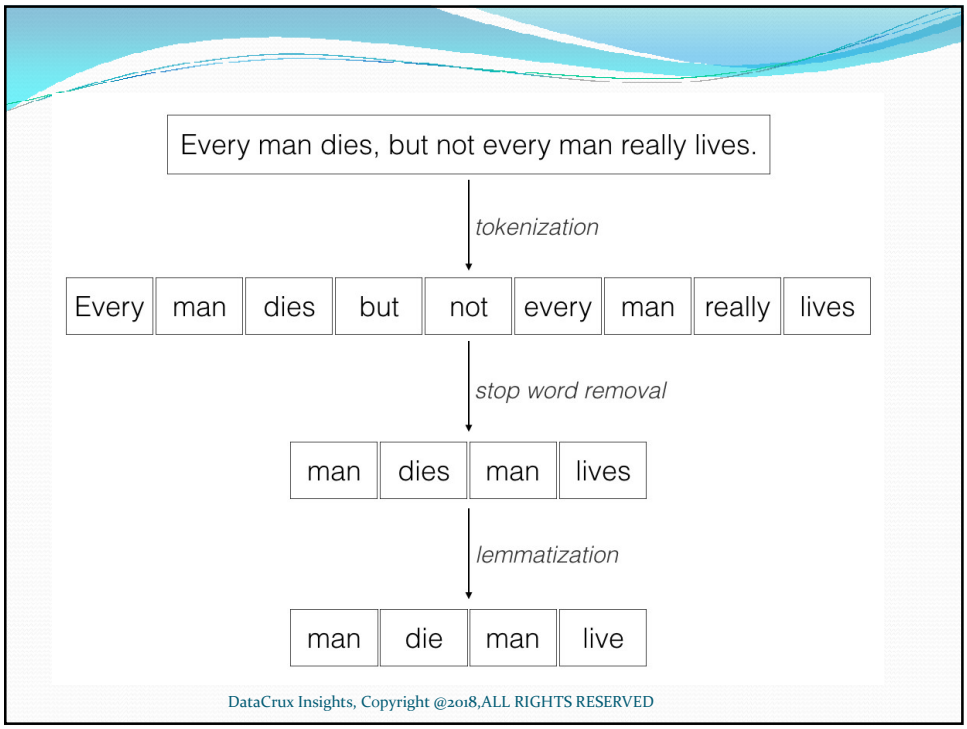
Google's Translation : घास दूसरी तरफ हरियाली है |

Other applications

- Text summarization
  - Extract keywords or key-phrases from a large piece of text.
  - Creating an abstract of an entire article.
- Context analysis
  - Social networking sites can 'fairly' understand the topic of discussion

  " 4 of your friends posted about Indian Institute of Technology, Guwahati".
- Sentiment analysis
  - Help companies analyze large number of reviews on a product
  - Help customers process the reviews provided on a product.





Tasks in NLP

- Tokenization / Segmentation
- Disambiguation
- Stemming
- Part of Speech (POS) tagging
- Contextual Analysis
- Sentiment Analysis

Segmentation

- **Segmenting** text into words

"The meeting has been scheduled for this Saturday."

"He has agreed to co-operate with me."

"Indian Airlines introduces another flight on the New Delhi–Mumbai route."

"We are leaving for the U.S.A. on 26th May."

"Vineet is playing the role of Duke of Athens in A Midsummer Night's Dream in a theatre in New Delhi."

- **Named Entity Recognition**

Stemming

- Stemming is the process for reducing inflected (or sometimes derived) words to their stem, base or root form.

- car, cars -> car
- run, ran, running -> run
- stemmer, stemming, stemmed -> stem

Stemming vs. Lemmatization

- The purpose of both stemming and lemmatization is to reduce morphological variation.
- Stemming reduces word-forms to (pseudo)stems, whereas **lemmatization** reduces the word-forms to linguistically valid lemmas (morphological stems).
  - Stemming: car, cars, car's, cars' => car
  - Lemmatizing: am, are, is => be ;
    drive, drives, drove, driven => drive
- In a way, lemmatization deals only with inflectional variance, whereas stemming may also deal with derivational variance;

POS tagging

- **Part of speech (POS) recognition**

" Today is a beautiful day. "

Today	is	a	beautiful	day
Noun	Verb	Article	Adjective	Noun

"Interest rates interest economists for the interest of the nation."
(word sense disambiguation)

Word Sense Disambiguation

- Same word different meanings.

"He approached many banks for the loan."

vs

"IIT Guwahati is on the banks of Bhramaputra."

"Free lunch." vs "Free speech."

Sentiment Analysis

- Reviews about a restaurant :-

  "Best roast chicken in New Delhi."

  "Service was very disappointing."

- Another set of reviews

  "iPhone 4S is over-hyped."

  "The hype about iPhone 4S is justified."

Supervised vs. Unsupervised

- Supervised
  - Use of large training data to generalize patterns and rules
  - Example: Hidden Markov Models
- Unsupervised
  - Don't require training; use in-built rules or a general algorithm; can work straightaway on any unknown situations or problem
  - The algorithm may be developed as a result of linguistic analysis
  - Example: 'Text Rank' Algorithm for text summarization

Motivation for NLP

- Understand language analysis & generation
- Communication
- Language is a window to the mind
- Data is in linguistic form
- Data can be in Structured (table form), Semi structured (XML form), Unstructured (sentence form).

Two Contrasting Views of Language

- Language as a phenomenon
- Language as a data

Language Processing

- *Level 1* – Speech sound (*Phonetics & Phonology*)
- *Level 2* – Words & their forms (*Morphology, Lexicon*)
- *Level 3* – Structure of sentences (*Syntax, Parsing*)
- *Level 4* – Meaning of sentences (*Semantics*)
- *Level 5* – Meaning in context & for a purpose (*Pragmatics*)
- *Level 6* – Connected sentence processing in a larger body of text (*Discourse*)

Examples of Levels

- L1 : sound
- L2 : Dog - Dog(*s*), Dog(*ged*)
  Lady – Lad(*ies*)
  Should we store all forms of words in the lexicon?
- L3 : Ram goes to market ***(right)***
  goes Ram to the market ***(wrong)***
- L4 : translation from unstructured to structured representation
  go : (event)
  agent : Ram
  source : ?
  destination : market

Example (Contd.)

- L5 : User situation & context

  "*Is that water?*" – the action to be performed is different in a chemistry lab and on a dining table.
- L6 : Backward & forward references –
- Coreference resolution

  "*The man went near the dog. It bit him.*"

  Often co reference & ambiguity go together as in –

  "*The dog went near the cat. It bit it.*"

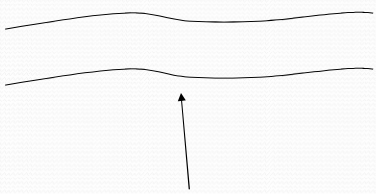
Statistical Concerns

- L1 : speech (make sense of sound)

  Approach –
  - Learning based
  - Probabilistic

Noisy Channel Metaphor

Speech Signal



Text

Noisy

- I want food.

- It is cold today.

Data-Driven Approach

The issues in this approach are -

- Corpora collection (coherent piece of text)
- Corpora cleaning – spelling, grammar, strange characters' removal
- Annotation
  - Named entity recognition
  - POS detection
  - Parsing
  - Meaning

The biggest challenge for NLP is *Ambiguity*.

32

Ambiguity in Natural Language

Ambiguity can be of 2 types –

- Lexical – multiple meanings of words
  - It is dealt with in "*lexical semantics*"
    - Ex - "*The bank organized a loan mela on the bank of the river*"
- Structural –
  - It is dealt with in parsing.
    - Ex – "*I saw the boy with a telescope*"

Latent Dirichlet Allocation for Topic Modeling(LDA)

- There are many approaches for obtaining topics from a text such as – Term Frequency and Inverse Document Frequency. Non Negative Matrix Factorization techniques.

- LDA assumes documents are produced from a mixture of topics. Those topics then generate words based on their probability distribution. Given a dataset of documents, LDA backtracks and tries to figure out what topics would create those documents in the first place.

	W1	W2	W3	Wn
D1	0	2	1	3
D2	1	4	0	0
D3	0	2	3	1
Dn	1	1	3	0

Document-Term matrix

LDA

M1->Document-Topics matrix

	K1	K2	K3	K
D1	1	0	0	1
D2	1	1	0	0
D3	1	0	0	1
Dn	1	0	1	0

	W1	W2	W3	Wm
K1	0	1	1	1
K2	1	1	1	0
K3	1	0	0	1
K	1	1	0	0

M2 ->Topics-Term matrix

- P1 – p(topic t / document d) = the proportion of words in document d that are currently assigned to topic t
- P2 – p(word w / topic t) = the proportion of assignments to topic t over all documents that come from this word w

Parameters of LDA

- Alpha and Beta Hyperparameters
- alpha represents document-topic density and Beta represents topic-word density. Higher the value of alpha, documents are composed of more topics and lower the value of alpha, documents contain fewer topics. On the other hand, higher the beta, topics are composed of a large number of words in the corpus, and with the lower value of beta, they are composed of few words

- ✓ LDA is a generative model like Dirichlet Clustering. You start with a known model and try to explain the data by refining the parameters to fit the model to the data.
- ✓ LDA does this by assuming that the whole corpus has some k number of topics, and each document is talking about these k topics. Therefore, the document is considered a mixture of topics with different probabilities for each.

The algorithm is similar to k-means.

Initialize k clusters

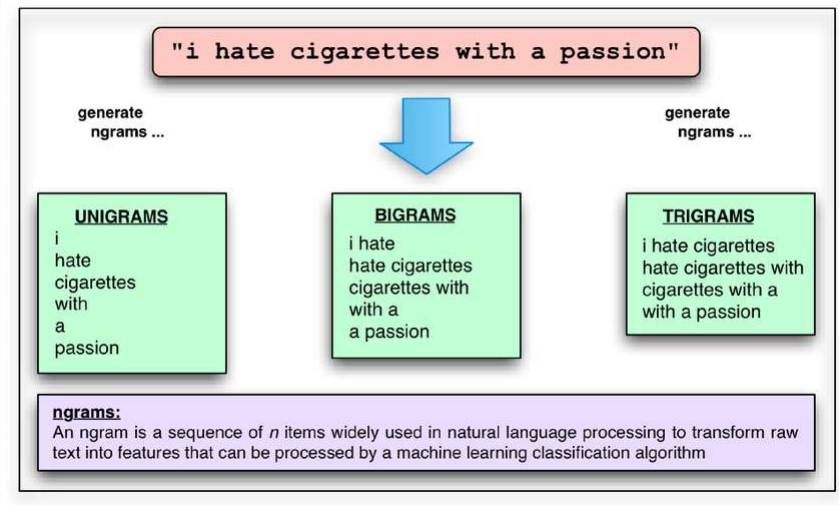
- ✓ Until converged
  - Compute the probability of a point belong to a cluster for every <point,cluster> pair
  - Re compute the cluster centres using above probability membership values of points to clusters

Example

- ✓ Twitter uses Mahout's LDA implementation for user interest modeling, and maintains a (periodically sync'ed) with Apache trunk) fork of Mahout on GitHub.

n-gram model

- An ***n*-gram model** is a type of probabilistic language model for predicting the next item in such a sequence in the form of a $(n-1)$–order Markov model.



Text Chunking

- **What is chunking**
- Text chunking, also referred to as **shallow parsing**, is a task that follows Part-Of-Speech Tagging and that adds more structure to the sentence. The result is a grouping of the words in "chunks".

tf-idf (*term frequency-inverse document frequency*)

Decomposed into 2 parts

TF(Term Frequency) * IDF(inverse document frequency)

Where:
TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document)

IDF(t) = log_e(Total number of documents / Number of documents with term t in it)
.