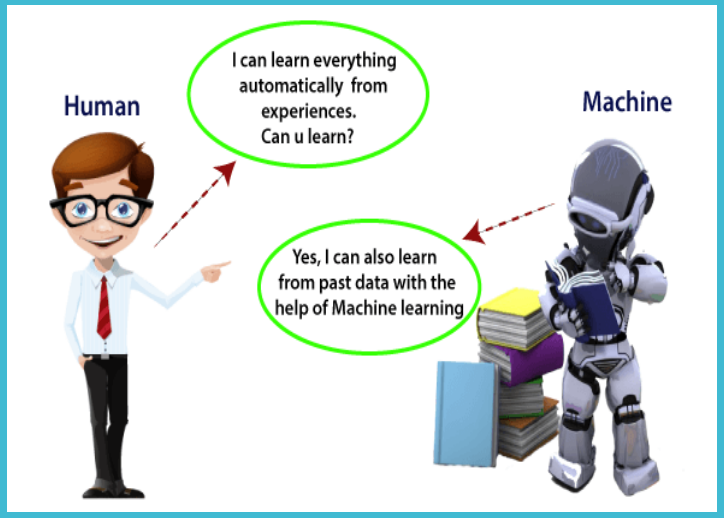
Natural Language Processing



Marwadi University

Department of Information and Communication Technology

Unit 8: NLP

Artificial Intelligence (01CT0703)

Prof. Nishith Kotak

What is NLP?

- Study of interactions between humans and computers
- NLP=Computer Science + AI + Computational Linguistics

What is NLP?

- Computer aided text analysis of human language.
- The goal is to enable machines to understand human language and extract meaning from text.
- It is a field of study which falls under the category of machine learning and more specifically computational linguistics.
- The "Natural Language Toolkit" is a python module that provides a variety of functionality that will aide us in processing text.

Paradox in ML

- Sentiment (Sad, Happy ???)
- Ambiguity
- Intent (Sarcasm or Slang)
- Context (Is GOOGLE a noun or Verb??)

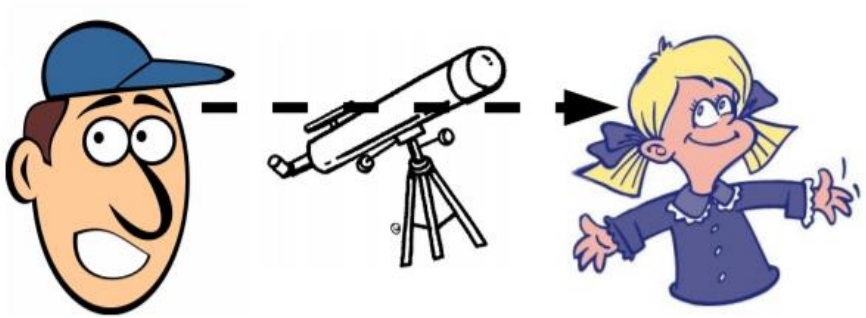
Eg. of conflict in Context

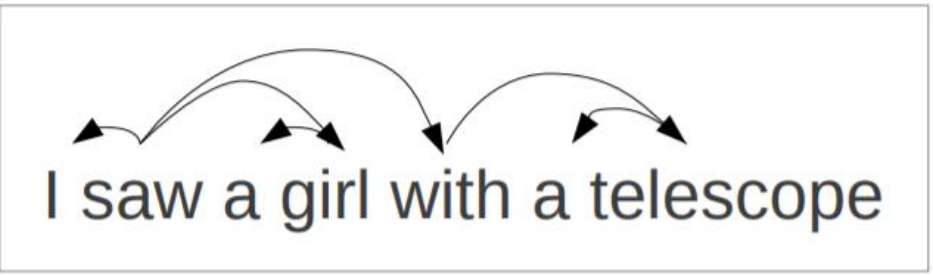
- Friend: Spell your name…
- Me: N..I..S..H..I..T..H
- Friend: Wrong…. Its Y..O..U..R..N..A..M..E

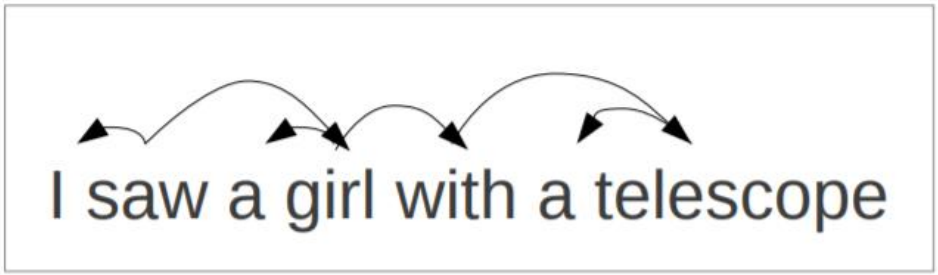
☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺

Ambiguity

I SAW A GIRL WITH A TELESCOPE







Common NLP Tasks

Easy

- Chunking
- Part-of-Speech Tagging
- Named Entity Recognition
- Spam Detection
- Thesaurus

Medium

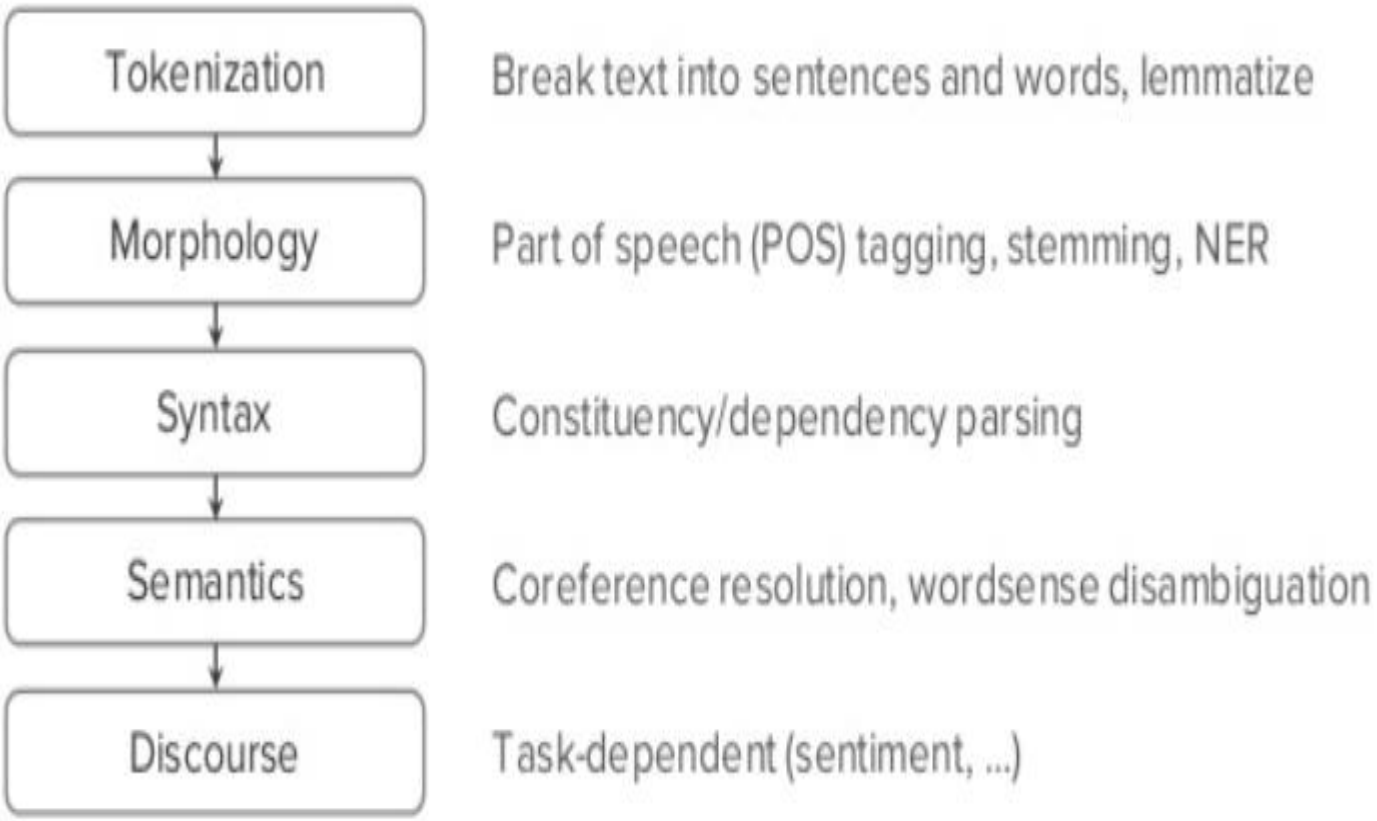
- Syntactic Parsing
- Word Sense Disambiguation
- Sentiment Analysis
- Topic Modeling
- Information Retrieval

Hard

- Machine Translation
- Text Generation
- Automatic Summarization
- Question Answering
- Conversational Interfaces

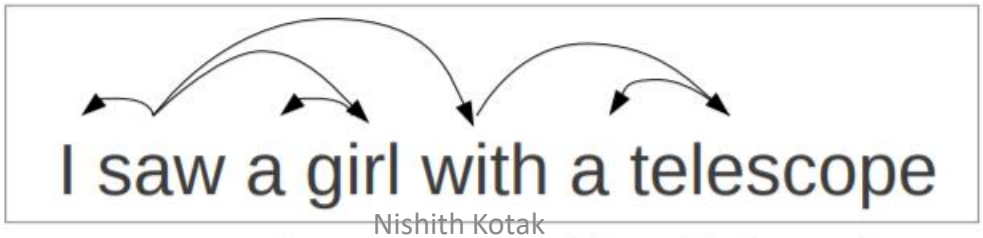
Tokenization	Segmenting text into words, punctuations marks etc.
Part-of-speech (POS) **Tagging**	Assigning word types to tokens, like verb or noun.
Dependency Parsing	Assigning syntactic dependency labels, describing the relations between individual tokens, like subject or object.
Lemmatization	Assigning the base forms of words. For example, the lemma of “was” is “be”, and the lemma of “rats” is “rat”.
Sentence Boundary Detection (SBD)	Finding and segmenting individual sentences.
Named Entity Recognition (NER)	Labelling named “real-world” objects, like persons, companies or locations.
Entity Linking (EL)	Disambiguating textual entities to unique identifiers in a Knowledge Base.
Similarity	Comparing words, text spans and documents and how similar they are to each other.
Text Classification	Assigning categories or labels to a whole document, or parts of a document.
Rule-based Matching	Finding sequences of tokens based on their texts and linguistic annotations, similar to regular expressions.
Training	Updating and improving a statistical model’s predictions.
Serialization	Saving objects to files or byte strings.

Classical NLP Pipeline



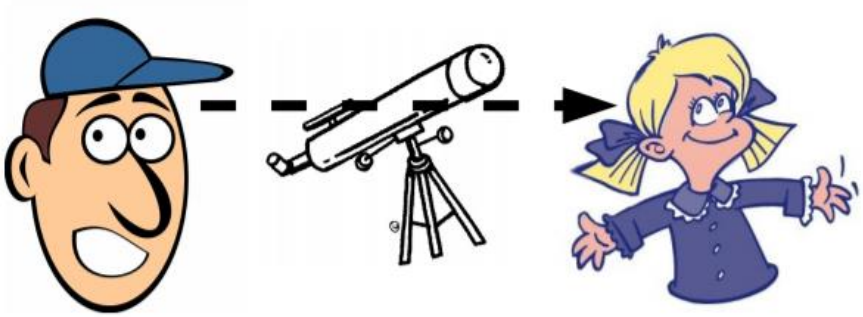
Dependency Parsing

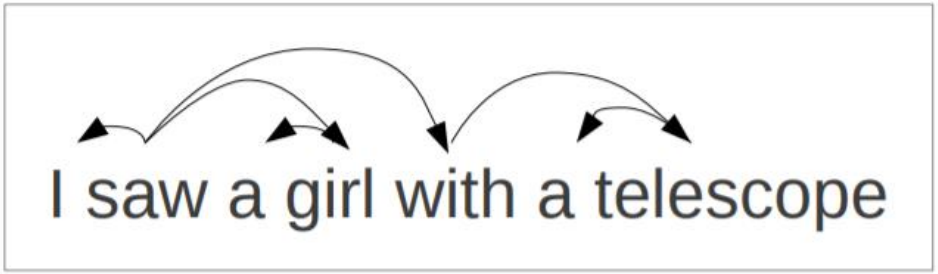
- Words don't stand by their own but they are connected with each other by some hidden structures.
- It is the task of **extracting a dependency parse** of a sentence that represents its grammatical structure and defines the relationships between "**head**" words and words, which modify those heads i.e. **dependents**.
- Each word is connected with other word by a direct link called dependencies.

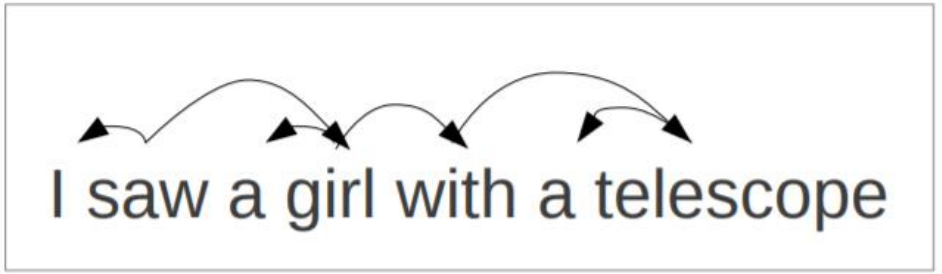


Dependency Parser

Dependencies also resolves the ambiguities







Tokenization
Stemming
Lemmatization

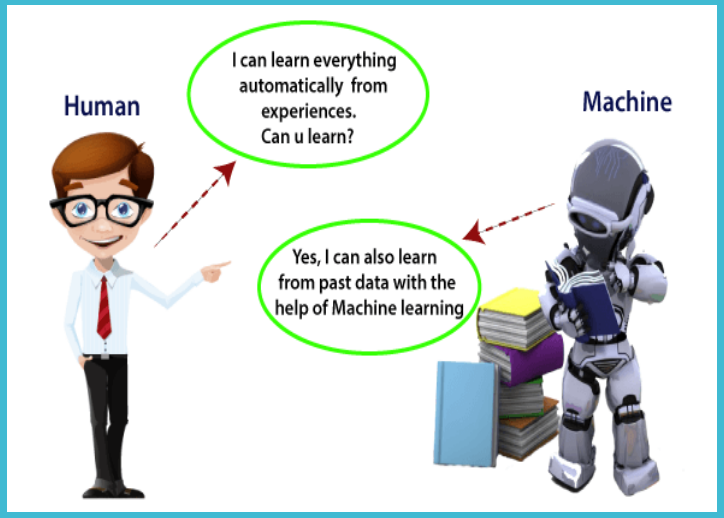
- Stemming and Lemmatization both generate the **root form** of the inflected words. The difference is that **stem** might **not be an actual word** whereas, lemma is an actual language word.

- Stemming follows an algorithm with steps to perform on the words which makes it faster. Whereas, in lemmatization, you used **WordNet** corpus and a corpus for stop words as well to produce lemma which makes it slower than stemming. You also had to define a parts-of-speech to obtain the correct lemma.

Stemming Lemmatization

Word	Stemming	Lemmatization
Studies	Studi	Study
Studying	Study	Study

Word Embeddings of NLP



Marwadi University

Department of Information and Communication Technology

Unit 8: NLP

Artificial Intelligence (01CT0703)

Prof. Nishith Kotak

Bag of Words

Awesome
Thank You

Bag of Words

Awesome
Thank You

- Awesome Thank You
- Great thank you
- Not bad not good

awesome	thank	you	great	not	bad	good
1	1	1	0	0	0	0

Awesome Thank You
[1,1,1,0,0,0,0]

Limitation of Bag of Words

- If we use all English words, vector would be very long and hence sparse.
- Frequent Words have More Power

- the man like the girl

the	man	girl	like	love
2	1	1	1	0

- the man love the girl

the	man	girl	like	love
2	1	1	0	1

- the the the the the

the	man	girl	like	love
5	0	0	0	0

Limitation of Bag of Words

- Cannot handle out of the vocabulary/ unseen words
- Eg.: gr8, goooood, gud mrng, thnk u, etc

Document Similarity

- d1: the best Italian restaurant enjoy the best pasta
- d2: American restaurant enjoy the best hamburger
- d3: Korean restaurant enjoy the best bibimbap
- d4: the best the best American restaurant

Bag of words document similarity

	Italian	restau rant	enjoy	the	best	pasta	American	ham burger	Korean	bibimbap
d1	1	1	1	2	2	1	0	0	0	0
d2	0	1	1	1	1	0	1	1	0	0
d3	0	1	1	1	1	0	0	0	1	1
d4	0	1	0	2	2	0	1	0	0	0

	Italian	restaurant	enjoy	the	best	pasta	American	hamburger	Korean	bibimbap	
d1	1	1	1	2	2	1	0	0	0	0	[1, 1, 1, 2, 2, 1, 0, 0, 0, 0]
d2	0	1	1	1	1	0	1	1	0	0	[0, 1, 1, 1, 1, 0, 1, 1, 0, 0]
d3	0	1	1	1	1	0	0	0	1	1	[0, 1, 1, 1, 1, 0, 0, 0, 1, 1]
d4	0	1	0	2	2	0	1	0	0	0	[0, 1, 0, 2, 2, 0, 1, 0, 0, 0]

- d1: the best Italian restaurant enjoy the best pasta
- d2: American restaurant enjoy the best hamburger
- d3: Korean restaurant enjoy the best bibimbap
- d4: the best the best American restaurant

$$\text{Cosine similarity} = \frac{A \cdot B}{\|A\| * \|B\|}$$

$$\frac{[1, 1, 1, 2, 2, 1, 0, 0, 0, 0] * [0, 1, 0, 2, 2, 0, 1, 0, 0, 0]^T}{\text{sqrt}(1+1+1+4+4+1+0+0+0+0) * \text{sqrt}(0+1+0+4+4+0+1+0+0+0)}$$

Document ID	Document	Cosine similarity with d4
d1	The best Italian restaurant enjoy the best pasta	0.82
d2	American restaurant enjoy the best hamburger	0.77
d3	Korean restaurant enjoy the best bibimbap	0.65
d4	The best the best American restaurant	1

TF-IDF

- TF = how frequently a term occurs in a document
- IDF = Log (Total # of Docs / # of Docs with the term in it)

word	TF				IDF	TF * IDF			
	d1	d2	d3	d4		d1	d2	d3	d4
Italian	1/8	0/6	0/6	0/6	log(4/1)=0.6	0.075	0	0	0
Restaurant	1/8	1/6	1/6	1/6	log(4/4)=0	0	0	0	0
enjoy	1/8	1/6	1/6	0/6	log(4/3)=0.13	0.016	0.02	0.02	0
the	2/8	1/6	1/6	2/6	log(4/4)=0	0	0	0	0
best	2/8	1/6	1/6	2/6	log(4/4)=0	0	0	0	0
pasta	1/8	0/6	0/6	0/6	log(4/1)=0.6	0.075	0	0	0
American	0/8	1/6	0/6	1/6	log(4/2)=0.3	0	0.05	0	0.05
hamburger	0/8	1/6	0/6	0/6	log(4/1)=0.6	0	0.1	0	0
Korean	0/8	0/6	1/6	0/6	log(4/1)=0.6	0	0	0.1	0
bibimbap	0/8	0/6	1/6	0/6	log(4/1)=0.6	0	0	0.1	0

Cosine Similarity using TF-IDF

Document	TF-IDF Bag of Words	Cosine similarity with d4
The best Italian restaurant enjoy the best pasta	[0.075, 0, 0.016, 0, 0, 0.075, 0, 0, 0, 0]	0
American restaurant enjoy the best hamburger	[0, 0, 0.02, 0, 0, 0, 0.05, 0.1, 0, 0]	0.5
Korean restaurant enjoy the best bibimbap	[0, 0, 0.02, 0, 0, 0, 0, 0, 0.1, 0.1]	0
The best the best American restaurant	[0, 0, 0, 0, 0, 0, 0.05, 0, 0, 0]	1

Advantage of TF-IDF

- Easy to get document similarity
- Keep relevant words score
- Lower just frequent words score

Drawbacks of TF-IDF/BoW

- Only based on terms
- Weak on capturing document topics
- Weak in handling synonyms

(different words but same meanings)

- Word sequence not considered

Eg:

MU Dean speak in college auditorium

Dr. R.B. Jadeja motivated student

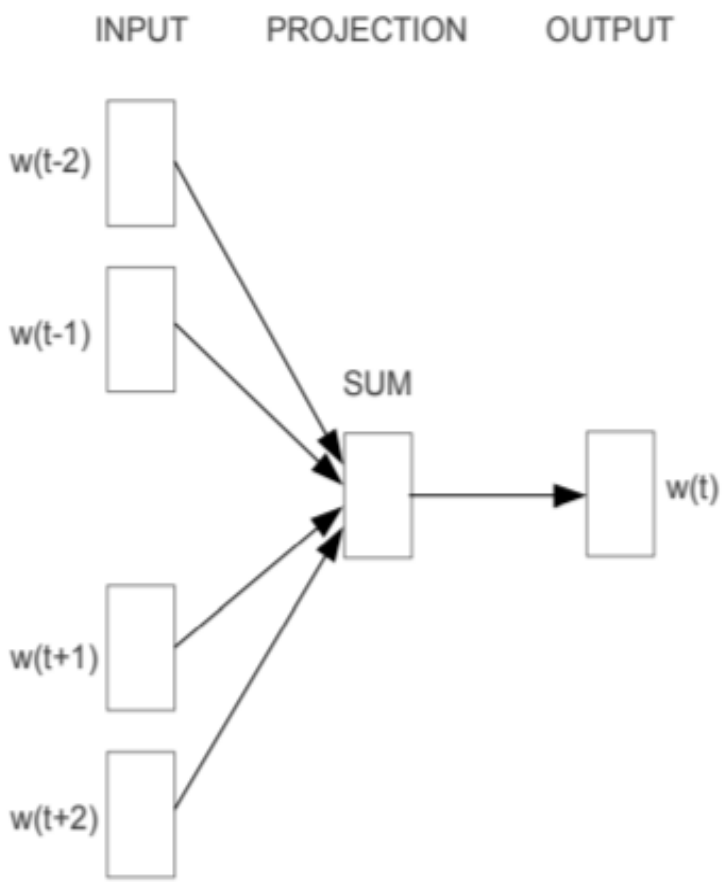
Remedies

- LSA
- LDA
- GloVe
- ConceptNets

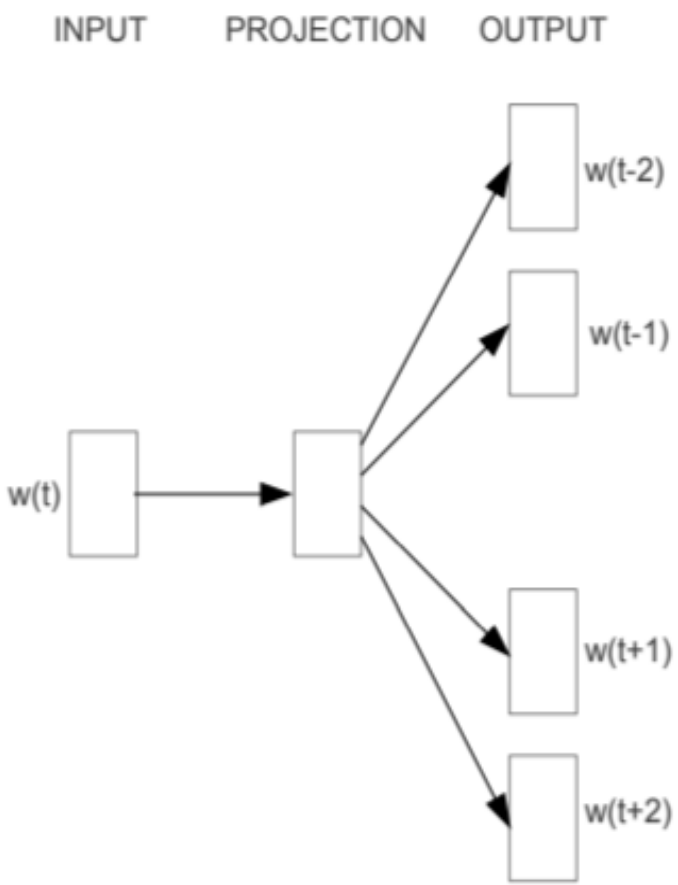
and many others

Word Embeddings

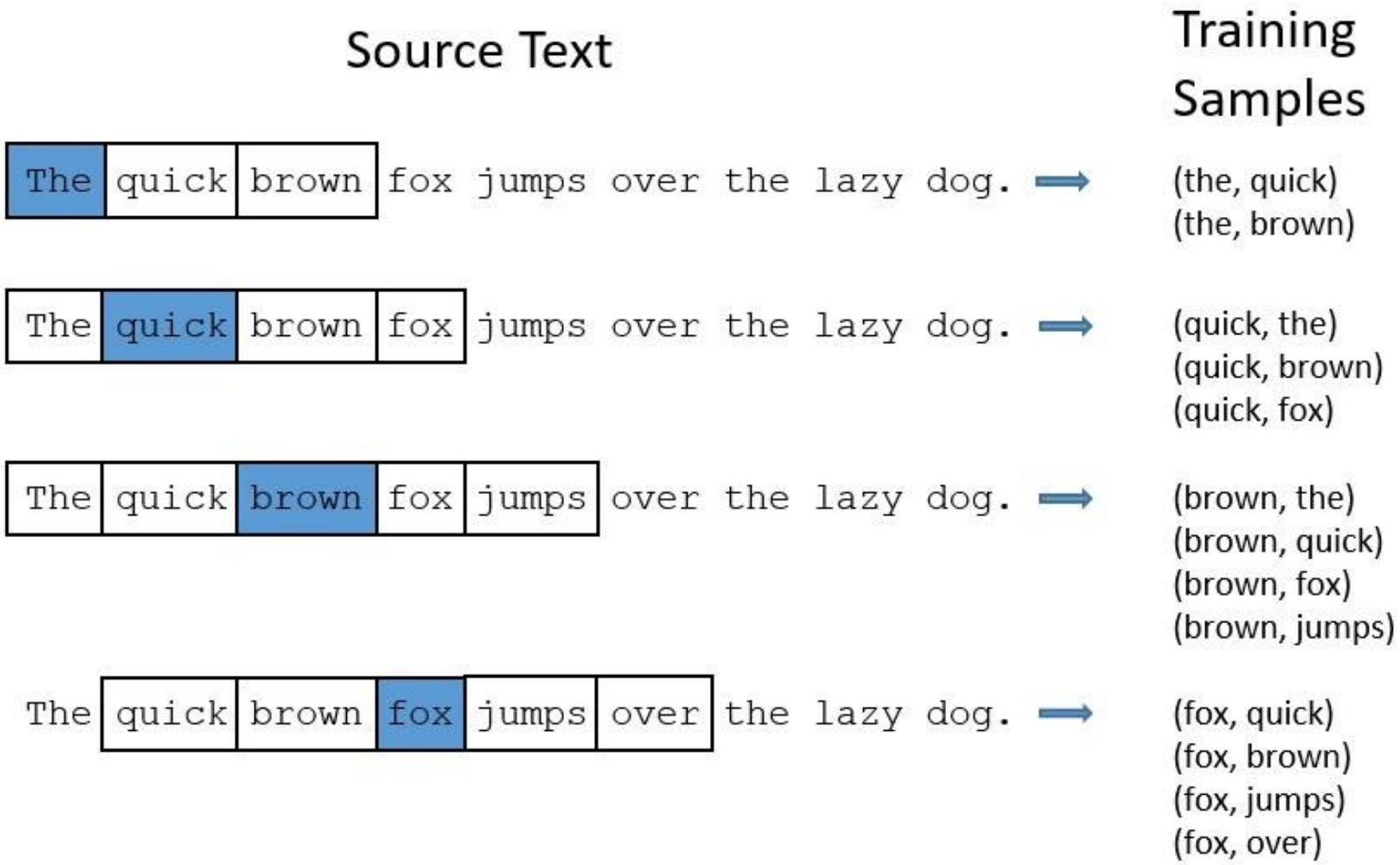
- Word2Vec (One Hot Encoding)
- CBOW
- N-gram
- Skip N-gram



CBOW



Skip-gram


Source Text
Training Samples
The quick brown fox jumps over the lazy dog.
(the, quick)
(the, brown)
The quick brown fox jumps over the lazy dog.
(quick, the)
(quick, brown)
(quick, fox)
The quick brown fox jumps over the lazy dog.
(brown, the)
(brown, quick)
(brown, fox)
(brown, jumps)
The quick brown fox jumps over the lazy dog.
(fox, quick)
(fox, brown)
(fox, jumps)
(fox, over)

Prediction of N-Grams

This is the house that Jack built.
This is the malt
That lay in the house that Jack built.
This is the rat,
That ate the malt
That lay in the house that Jack built.
This is the cat,
That killed the rat,
That ate the malt
That lay in the house that Jack built.

P(house | This is the)=?????
P(Jack | that)=?????

Applications of N-gram

- In OCR

If some words are not neatly visible, can be easily predicted by N-Gram models

- Correcting the sentence

For example, "deer sir" instead of "Dear sir"

- Speech to Text Processing/Speech Recognition

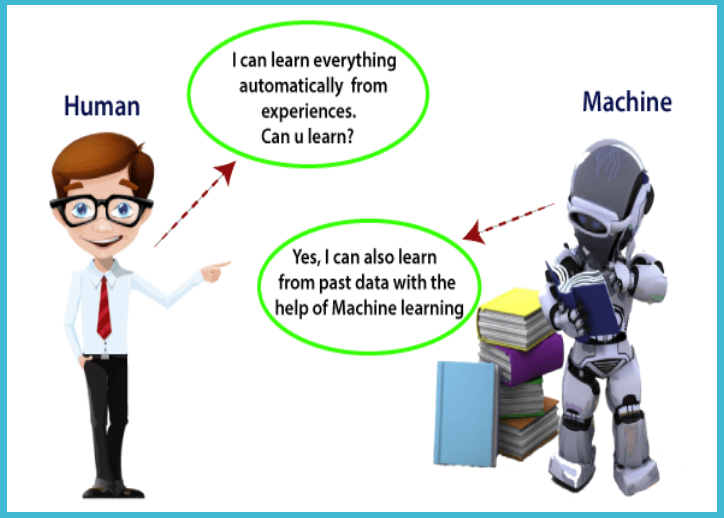
Having same sound, but different meaning
For eg. "EYE" and "I"
I am eating VS EYE am eating
His EYE got hurt VS His I got hurt

Applications of N-gram

- Suggestions while training

SMS suggestions of words

TextRank for Text Summarization



Marwadi University

Department of Information and Communication Technology

Unit 8: NLP

Artificial Intelligence (01CT0703)

Prof. Nishith Kotak

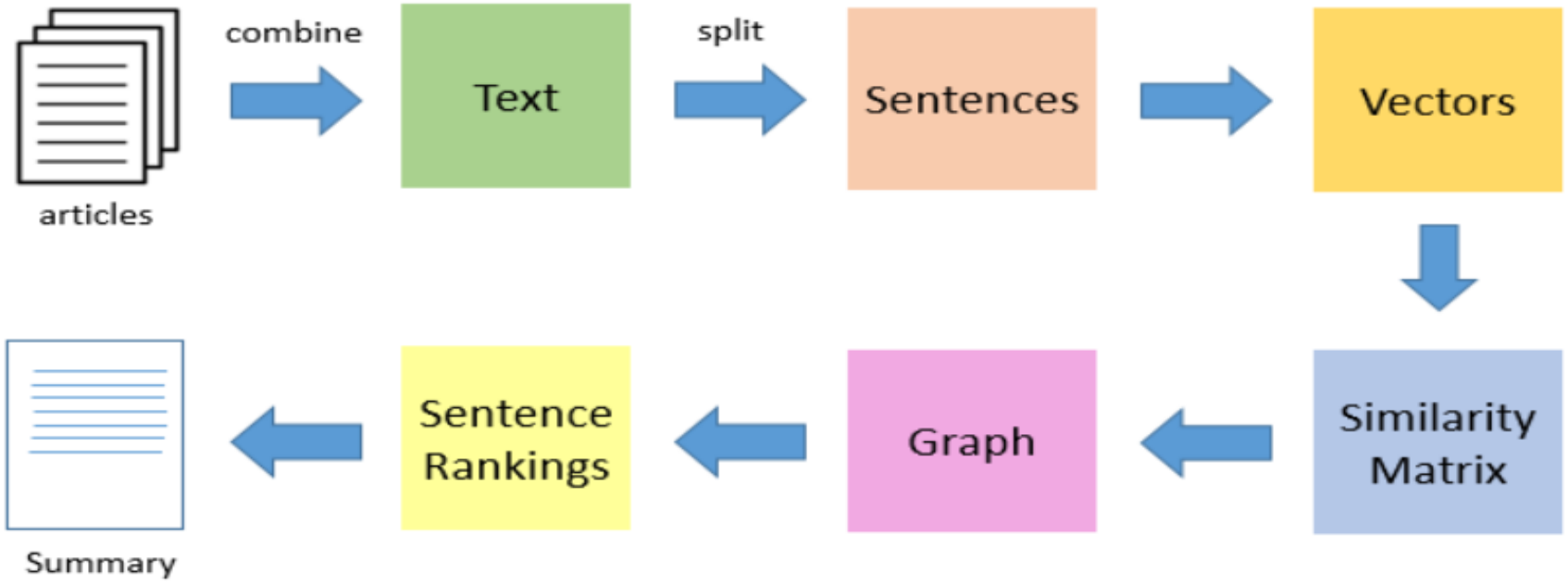
TextRank

- TextRank is a text summarization technique which is used in Natural Language Processing to generate Document Summaries.
- TextRank uses an extractive approach and is an unsupervised graph-based text summarization technique.
- TextRank uses the concept of PageRank algorithm
- PageRank is an algorithm used to calculate rank of web pages, and is used by search engines such as Google.
- This algorithm gets its name from Larry Page, one of the co-founders of Google.

TextRank

Algorithm

- The rank/importance of a page is decided by the number and quality of links to that page.
- It applies several iterations on the pages to arrive at a final value.



Working of TextRank

- Extract all the sentences from the text document, either by splitting at whitespaces or full stops, or any other way in which you wish to define your sentences.
- A Graph is created out of the sentences extracted in Step 1. The nodes represent the sentences, while the weight on the edges between two nodes is found by using a Similarity function, like Cosine Similarity.
- This step involves finding importance (scores) of each node by iterating the algorithm until convergence, i.e. until consistent scores are obtained.
- The sentences are sorted in a descending order based upon their scores. The first *k* sentences are chosen to be a part of the text summary.

Example of TextRank

A programming language comprises a set of instructions . It is used to produce various kinds of output. It is a language which involves a computer performing some kinds of instructions or algorithm to produce some kinds of output

-----Divide into Sentences-----

1 - A programming language comprises a set of instructions.
2 - It is used to produce various kinds of output.
3 - It is a language which involves a computer performing some kinds of instructions or algorithm to produce some kinds of output.

Example of TextRank

WORD	TF *1*	TF *2*	TF *3*	IDF	TF-IDF *1*	TF-IDF *2*	TF-IDF *3*
a	0.25	0	0.1	0.176	0.044	0	0.0176
programming	0.125	0	0	0.477	0.06	0	0
language	0.125	0	0.0476	0.176	0.022	0	0.0084
comprises	0.125	0	0	0.477	0.06	0	0
set	0.125	0	0	0.477	0.06	0	0
of	0.125	0.11	0.1	0	0	0	0
instructions	0.125	0	0.0476	0.176	0.022	0	0.0084
it	0	0.11	0.0476	0.176	0	0.005	0.0084
is	0	0.11	0.0476	0.176	0	0.005	0.0084
used	0	0.11	0	0.477	0	0.05	0
to	0	0.11	0	0.477	0	0.05	0
produce	0	0.11	0.0476	0.176	0	0.005	0.0084
various	0	0.11	0	0.477	0	0.05	0
kinds	0	0.11	0.0476	0.176	0	0.005	0.0084
output	0	0.11	0.0476	0.176	0	0.005	0.0084
which	0	0	0.0476	0.477	0	0	0.023
involves	0	0	0.0476	0.477	0	0	0.023
computer	0	0	0.0476	0.477	0	0	0.023
performing	0	0	0.0476	0.477	0	0	0.023
some	0	0	0.1	0.477	0	0	0.0477
or	0	0	0.0476	0.477	0	0	0.023
algorithm	0	0	0.0476	0.477	0	0	0.023

Example of TextRank

---------Make a Word Vector----------

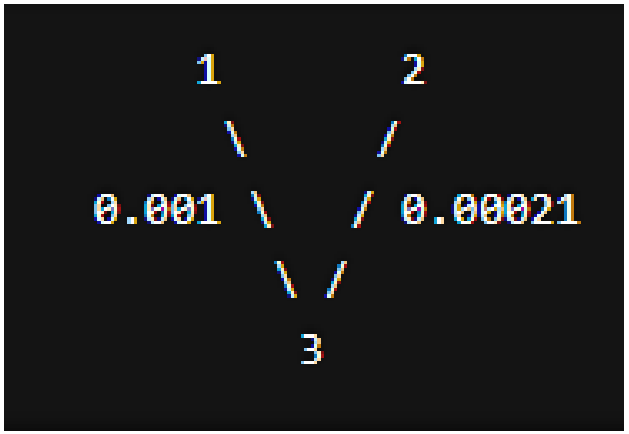
SENTENCE	VECTOR
1	[0.044, 0.06, 0.022, 0.06, 0.06, 0, 0.022, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
2	[0, 0, 0, 0, 0, 0, 0, 0.005, 0.005, 0.005, 0.005, 0.005, 0.005, 0.005, 0.005, 0, 0, 0, 0, 0, 0, 0]
3	[0.0176, 0, 0.0084, 0, 0, 0, 0.0084, 0.0084, 0.0084, 0, 0, 0.0084, 0, 0.0084, 0.0084, 0.023, 0.023, 0.023, 0.023, 0.0477, 0.023, 0.023]

Example of TextRank

---Compute Cosine Similarity---

SENTENCE	1	2	3
1	1	0	0.0011
2	0	1	0.00021
3	0.0011	0.00021	1

---Draw a graph for textrank---

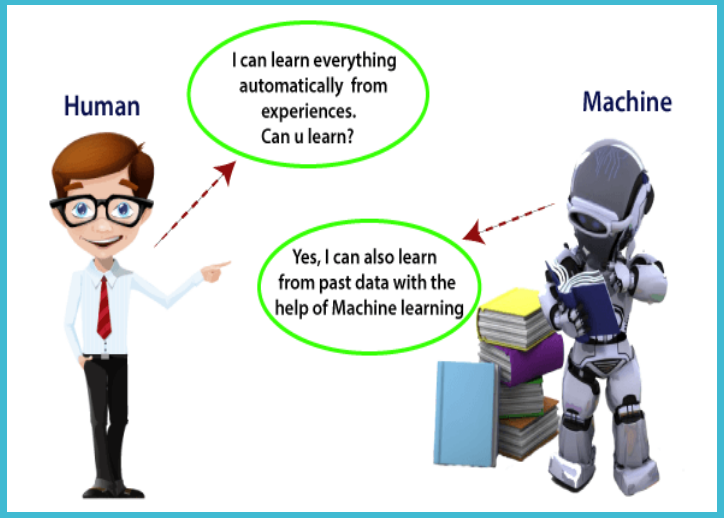


Example of TextRank

According to the PageRank algorithm, we can treat the sentences (nodes) as webpages and the edges as links to the webpages.

Applying PageRank to the above graph concludes that node 3 is the most important/highly ranked sentence since it contains most links to other nodes.

LDA for Topic Modelling



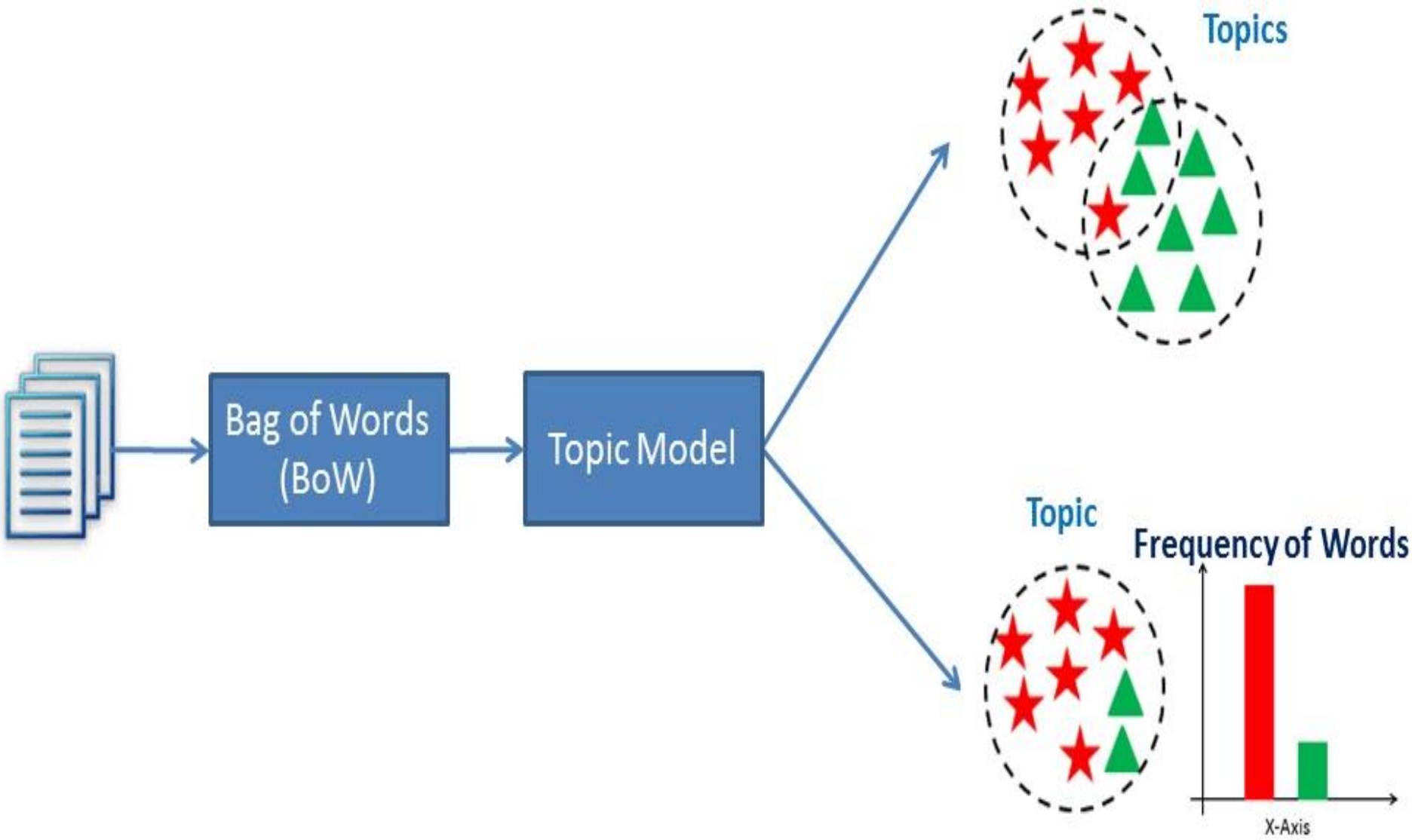
Marwadi University

Department of Information and Communication Technology

Unit 8: NLP

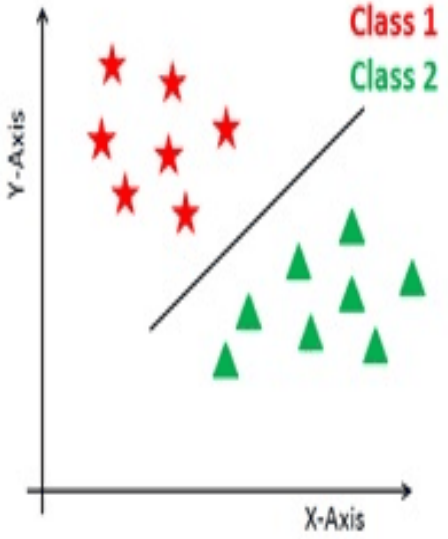
Artificial Intelligence (01CT0703)

Prof. Nishith Kotak


Topics
Bag of Words
(BoW)
Topic Model
Topic
Frequency of Words
X-Axis

Bag of Words (BoW)

Text Classification



Topic Modeling

	Doc-1	Doc-2	Doc-3	Doc-4
Topic-1	0.425	0.032	0.015	0.553
Topic-2	0.052	0.512	0.623	0.121

Dirichlet Allocation