NATURAL LANGUAGE PROCESSING

Start

What

The ability of a computer to understand text and spoken words

- NLP Process — Process to simplify human lang. to make it understandable.
- Data Processing
  - Text Normalisation
  - Sentence Segmentation
  - Tokenisation
  - Removal of Stop word
  - Converting into same case
  - Stemming and Lemmatization
- Bag of word Algorithm
- TFIDF
  - Term Frequency
  - Inverse Document Frequency
  - Applications of TFIDF

Why

Applications of NLP

- Chat Bots — Ex. Mitsuku Bot, Clever Bot, Jabberwacky, and Haptik.
  - Smart Bot
  - Script Bot
- Automatic Summarization
- Sentiment Analysis
- Text classification
- Virtual Assistants

Human VS Computer

Problems in Understanding human languages by computers.

- Human Language
- Computer Language
- Arrangement of words & meanings
  - (Structure) Syntax
  - (Meaning) Semantics
- Multiple Meanings of a word
- Perfect Syntax, no Meaning

Practice

CLICK TEXT TO OPEN THE LINK

- Download Revision Notes Pdf
- Solve Important Questions
- Practice VIP Questions PDF
- Practice Sample Papers
- Ask and Solve Doubts at Aiforkids Doubts corner
- NLP Explanation Video

FINISH

Youtube.com/aiforkids

Aiforkids.in/class-10/nlp

" Learning is not a course, Its a path from passion to profession "

~Lalit Kumar

WHAT IS NLP?

Natural Language Processing (NLP) is the sub-field of AI that focuses on the ability of a computer to understand human language (command) as spoken or written and to give an output by processing it.

APPLICATIONS OF NLP

→ Feel Deta h, NLP KA

Automatic Summarization

- Summarizing the meaning of documents and information
- Extract the key emotional information from the text to understand the reactions (Social Media)

Sentiment Analysis

- Identify sentiments and emotions from one or more posts
- Companies use it to identify opinions and sentiments to get feedback
- Can be Positive, Negative or Neutral

Text classification

- Assign predefined categories to a document and organize it to help you find the information you need or simplify some activities.
- Eg: Spam filtering in email.

Virtual Assistants

- By accessing our data, they can help us in keeping notes of our tasks, making calls for us, sending messages, and a lot more.
- With speech recognition, these assistants can not only detect our speech but can also make sense of it.
- A lot more advancements are expected in this field in the near future
- Eg: Google Assistant, Cortana, Siri, Alexa, etc

REVISING AI PROJECT CYCLE

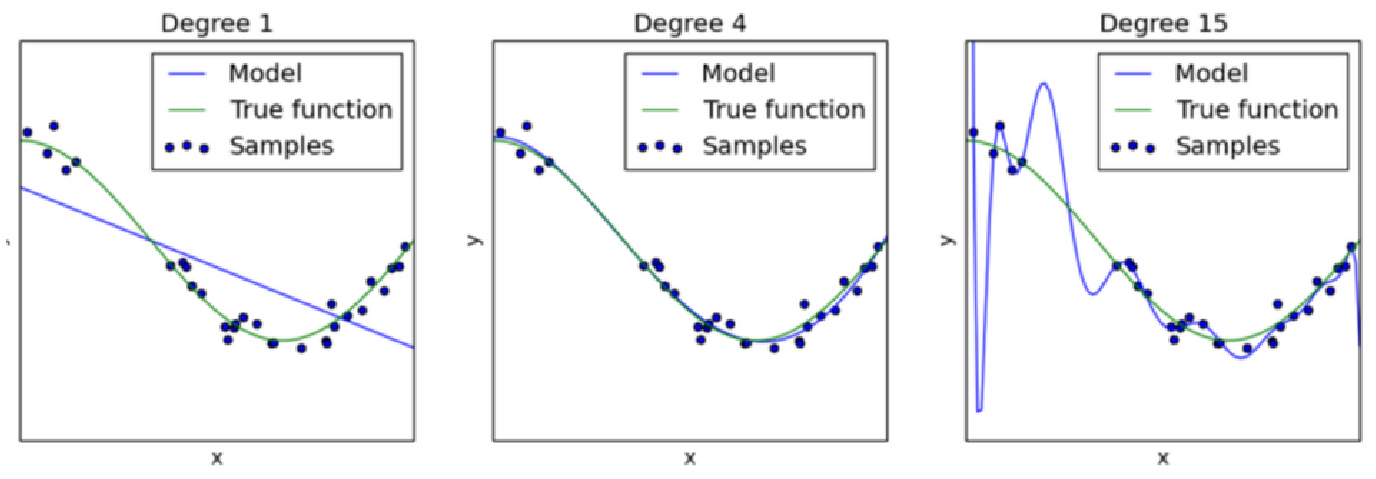
Project Cycle is a step-by-step process to solve problems using proven scientific methods and drawing inferences about them.

1 COMPONENTS OF PROJECT CYCLE

- Problem Scoping – Understanding the problem
- Data Acquisition – Collecting accurate and reliable data
- Data Exploration – Arranging the data uniformly
- Modelling – Creating Models from the data
- Evaluation – Evaluating the project

The	Stakeholder	Who
Have a problem	Issue/Problem	What
When/While	Context/Situation/Location	Where
Ideal Solution	How the Solution will help Stakeholders	Why

[Problem Statement Template]



Under Fit | Perfect Fit | Over Fit

CHATBOTS

One of the most common applications of Natural Language Processing is a chatbot.

Example:

- Mitsuku Bot
- Clever Bot
- Jabberwacky
- Haptic
- Rose
- OChatbot

Types of ChatBots

SCRIPT BOTS	SMART BOTS
• Easy to make	• Comparatively difficult to make
• Work on the script of the programmed set.	• Work on bigger databases
• Limited functionality	• Wide functionality
• No or little language processing skills	• Coding is required
• Example: Customer Care Bots.	• Example: Google Assistant, Alexa, Cortana, Siri, etc.

HUMAN LANGUAGE VS COMPUTER LANGUAGE

1 HUMAN LANGUAGE

- Humans communicate through language which we process all the time.
- Our brain keeps on processing the sounds that it hears around itself and tries to make sense out of them all the time.
- Communications made by humans are complex.

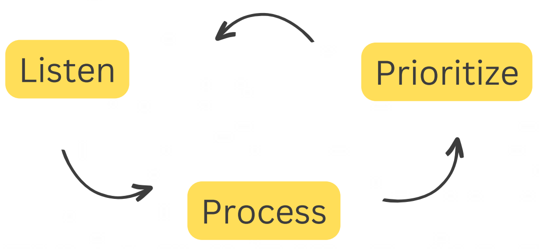
2 COMPUTER LANGUAGE

- The computer understands the language of numbers.
- Everything that is sent to the machine has to be converted to numbers.
- A single mistake is made, the computer throws an error and does not process that part.
- The communications made by the machines are very basic and simple

ERRORS IN PROCESSING HUMAN LANGUAGE

- Arrangement of words and meaning
  - Different Syntax, Same meaning
  - Different Meaning, Same Syntax
- Multiple Meanings of the Word
- Perfect Syntax, No Meaning

Our Brain



ARRANGEMENT OF THE WORDS AND MEANING

- Syntax: Syntax refers to the grammatical structure of a sentence.
- Semantics: It refers to the meaning of the sentence.

Different syntax, same semantics: 2+3 = 3+2

- Here the way these statements are written is different, but their meanings are the same that is 5.

Different semantics, same syntax: 3/2 (Python 2.7) ≠ 3/2 (Python 3).

- Here we have the same syntax but their meanings are different. In Python 2.7, this statement would result in 1 while in Python 3, it would give an output of 1.5.

1 MULTIPLE MEANINGS OF A WORD

To understand let us have an example of the following three sentences:

1. "His face turned red after he found out that he had taken the wrong bag"

- Possibilities: He feels ashamed because he took another person's bag instead of his OR he's feeling angry because he did not manage to steal the bag that he has been targeting.

2. "The red car zoomed past his nose"

- Possibilities: Probably talking about the color of the car, that traveled close to him in a flash.

3. "His face turns red after consuming the medicine"

- Possibilities: Is he having an allergic reaction? Or is he not able to bear the taste of that medicine?

2 PERFECT SYNTAX, NO MEANING

1. "Chickens feed extravagantly while the moon drinks tea"

- Meaning: This statement is correct grammatically but makes no sense. In Human language, a perfect balance of syntax and semantics is important for better understanding.

DATA PROCESSING

- Since we all know that the language of computers is Numerical, the very first step that comes to our mind is to convert our language to numbers.
- This conversion takes a few steps to happen. The first step to it is Text Normalisation.

TEXT NORMALISATION

In Text Normalization, we undergo several steps to normalize the text to a lower level. That is, we will be working on text from multiple documents and the term used for the whole textual data from all the documents altogether is known as "Corpus".

1 SENTENCE SEGMENTATION

Under sentence segmentation, the whole corpus is divided into sentences. Each sentence is taken as a different data so now the whole corpus gets reduced to sentences.

Example:

BEFORE SENTENCE SEGMENTATION	AFTER SENTENCE SEGMENTATION
• "You want to see the dreams with close eyes and achieve them? They'll remain dreams, look for AIMs and your eyes have to stay open for a change to be seen."	• You want to see the dreams with close eyes and achieve them?
	• They'll remain dreams, look for AIMs and your eyes have to stay open for a change to be seen

2 TOKENISATION

A "Token" is a term used for any word or number or special character occurring in a sentence.

Under Tokenisation, every word, number, and special character is considered separately and each of them is now a separate token.

Corpus: A corpus can be defined as a collection of text documents.

Example: You want to see the dreams with close eyes and achieve them?

↓

You | want | to | see | the | dreams | with | close

eyes | and | acheive | them | ?

4 REMOVAL OF STOPWORDS

Stopwords: Stopwords are the words that occur very frequently in the corpus but do not add any value to it.

Examples: a, an, and, are, as, for, it, is, into, in, if, on, or, such, the, there, to.

In this step, the tokens which are not necessary are removed from the token list. To make it easier for the computer to focus on meaningful terms, these words are removed.

Along with these words, a lot of times our corpus might have special characters and/or numbers.

> if you are working on a document containing email IDs, then you might not want to remove the special characters and numbers

Example: You want to see the dreams with close eyes and achieve them?

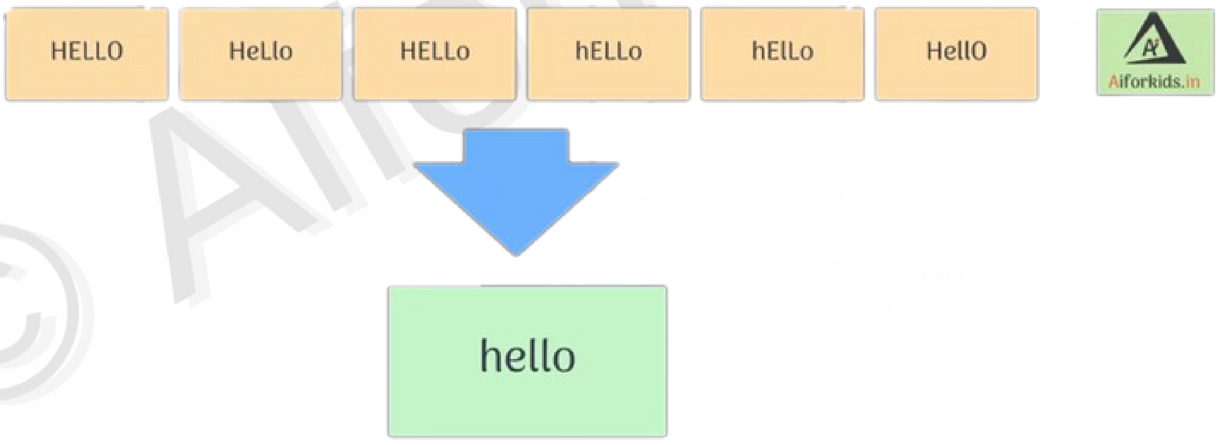
- the removed words would be
- **to, the, and, ?**

The outcome would be:

-> <u>You want see dreams with close eyes achieve them</u>

5 CONVERTING TEXT TO A COMMON CASE

We convert the whole text into a similar case, preferably lower case. This ensures that the case sensitivity of the machine does not consider the same words as different just because of different cases.



6 STEMMING

Stemming is a technique used to extract the base form of the words by removing affixes from them. It is just like cutting down the branches of a tree to its stems.

Might not be meaningful.

Example:

Words	Affixes	Stem
healing	ing	heal
dreams	s	dream
studies	es	studi

7 LEMMATIZATION

In lemmatization, the word we get after affix removal (also known as lemma) is a meaningful one and it takes a longer time to execute than stemming.

Lemmatization makes sure that a lemma is a word with meaning

Example:

Words	Affixes	lemma
healing	ing	heal
dreams	s	dream
studies	es	study

DIFFERENCE BETWEEN STEMMING AND LEMMATIZATION

Stemming	lemmatization
• The stemmed words might not be meaningful. • Caring ➔ Car	• The lemma word is a meaningful one. • Caring ➔ Care

BAG OF WORD ALGORITHM

Bag of Words just creates a set of vectors containing the count of word occurrences in the document (reviews). Bag of Words vectors is easy to interpret.

The bag of words gives us two things:

- A vocabulary of words for the corpus
- The frequency of these words (number of times it has occurred in the whole corpus).

> Here calling this algorithm a "bag" of words symbolizes that the sequence of sentences or tokens does not matter in this case as all we need are the unique words and their frequency in it.

STEPS OF THE BAG OF WORDS ALGORITHM

1. Text Normalisation: Collecting data and pre-processing it
2. Create Dictionary: Making a list of all the unique words occurring in the corpus. (Vocabulary)
3. Create document vectors: For each document in the corpus, find out how many times the word from the unique list of words has occurred.
4. Create document vectors for all the documents.

Example:

Step 1: Collecting data and pre-processing it.

Raw Data	Processed Data
• Document 1: Aman and Anil are stressed	• Document 1: [aman, and, anil, are, stressed]
• Document 2: Aman went to a therapist	• Document 2: [aman, went, to, a, therapist]
• Document 3: Anil went to download a health chatbot	• Document 3: [anil, went, to, download, a, health, chatbot]

Step 2: Create Dictionary

Dictionary in NLP means a list of all the unique words occurring in the corpus. If some words are repeated in different documents, they are all written just once while creating the dictionary.

aman	and	anil	are	stressed	went
download	health	chatbot	therapist	a	to

Some words are repeated in different documents, they are all written just once, while creating the dictionary, we create a list of unique words.

Step 3: Create a document vector

The document Vector contains the frequency of each word of the vocabulary in a particular document.

In the document, vector vocabulary is written in the top row.

- Now, for each word in the document, if it matches the vocabulary, put a 1 under it.
- If the same word appears again, increment the previous value by 1.
- And if the word does not occur in that document, put a 0 under it.

aman	and	anil	are	stressed	went	to	a	therapist	download	health	chatbot
1	1	1	1	1	0	0	0	0	0	0	0

Step 4: Creating a document vector table for all documents

aman	and	anil	are	stressed	went	to	a	therapist	download	health	chatbot
1	1	1	1	1	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1	1	0	0	0
0	0	1	0	0	1	1	1	0	1	1	1

In this table, the header row contains the vocabulary of the corpus and three rows correspond to three different documents.

Finally, this gives us the document vector table for our corpus. But the tokens have still not converted to numbers. This leads us to the final steps of our algorithm: TFIDF.

TFIDF

TFIDF stands for Term Frequency & Inverse Document Frequency.

1 TERM FREQUENCY

1. Term frequency is the <u>frequency of a word in one document</u>.
2. Term frequency can easily be found in the document vector table

Example:

aman	and	anil	are	stressed	went	to	a	therapist	download	health	chatbot
1	1	1	1	1	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1	1	0	0	0
0	0	1	0	0	1	1	1	0	1	1	1

Here, as we can see that the frequency of each word for each document has been recorded in the table. These numbers are nothing but the Term Frequencies!

2 DOCUMENT FREQUENCY

Document Frequency is the number of documents in which the word occurs irrespective of how many times it has occurred in those documents.

aman	and	anil	are	stressed	went	to	a	therapist	download	health	chatbot
2	1	2	1	1	2	2	2	1	1	1	1

We can observe from the table is:

1. Document frequency of 'aman', 'anil', 'went', 'to' and 'a' is 2 as they have occurred in two documents.
2. Rest of them occurred in just one document hence the document frequency for them is one.

3 INVERSE DOCUMENT FREQUENCY

In the case of inverse document frequency, we need to put the document frequency in the denominator while the total number of documents is the numerator.

aman	and	anil	are	stressed	went	to	a	therapist	download	health	chatbot
3/2	3/1	3/2	3/1	3/1	3/2	3/2	3/2	3/1	3/1	3/1	3/1

FORMULA OF TFIDF

The formula of TFIDF for any word W becomes:
TFIDF(W) = TF(W) * log(IDF(W))

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

We don't need to calculate the log values by ourselves. We simply have to use the log function in the calculator and find out!

aman	and	anil	are	stressed	went	to	a	therapist	download	health	chatbot
1*log(3/2)	1*log(3)	1*log (3/2)	1*log(3)	1*log(3)	0*log (3/2)	0*log (3/2)	0*log (3/2)	0*log(3)	0*log(3)	0*log(3)	0*log(3)
1*log(3/2)	0*log(3)	0*log (3/2)	0*log(3)	0*log(3)	1*log (3/2)	1*log (3/2)	1*log (3/2)	1*log(3)	0*log(3)	0*log(3)	0*log(3)
0*log(3/2)	0*log(3)	1*log (3/2)	0*log(3)	0*log(3)	1*log (3/2)	1*log (3/2)	1*log (3/2)	0*log(3)	1*log(3)	1*log(3)	1*log(3)

After calculating all the values, we get:

aman	and	anil	are	stressed	went	to	a	therapist	download	health	chatbot
0.176	.477	0.176	0.477	0.477	0	0	0	0	0	0	0
0.176	0	0	0	0	0.176	0.176	0.176	0.477	0	0	0
0	0	0.176	0	0	0.176	0.176	0.176	0	0.477	0.477	0.477

Finally, the words have been converted to numbers. These numbers are the values of each document.

Here, we can see that since we have less amount of data, words like 'are' and 'and' also have a high value. But as the IDF value increases, the value of that word decreases.

That is, for example:

- Total Number of documents: 10
- Number of documents in which 'and' occurs: 10

Therefore, IDF(and) = 10/10 = 1

Which means: log(1) = 0. Hence, the value of 'and' becomes 0.

On the other hand, the number of documents in which 'pollution' occurs: 3

IDF(pollution) = 10/3 = 3.3333...

This means log(3.3333) = 0.522; which shows that the word 'pollution' has considerable value in the corpus.

Important concepts to remember:

1. Words that occur in all the documents with high term frequencies have the least values and are considered to be the stopwords
2. For a word to have a high TFIDF value, the word needs to have a high term frequency but less document frequency which shows that the word is important for one document but is not a common word for all documents.
3. These values help the computer understand which words are to be considered while processing the natural language. The higher the value, the more important the word is for a given corpus.

APPLICATIONS OF TFIDF

TFIDF is commonly used in the Natural Language Processing domain. Some of its applications are:

1. Document Classification – Helps in classifying the type and genre of a document.
2. Topic Modelling – It helps in predicting the topic for a corpus.
3. Information Retrieval System – To extract the important information out of a corpus.
4. Stop word filtering – Helps in removing the unnecessary words from a text body.