**NLP**

The first generation of AI was ML

The second generation was DL

The third generation of AL is considered to be NLP+TRANSFORMER

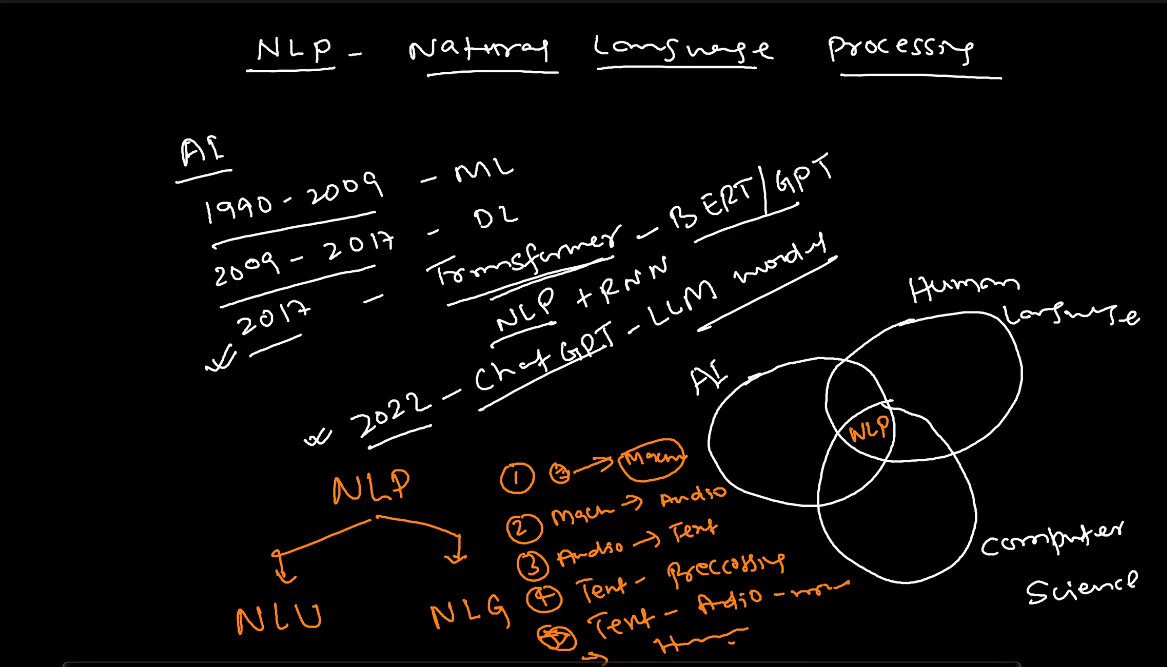
NLP is considered as the intersection between AI, Human Language and Computer Science.

NLP has two components:

* NLU(Natural Language Understanding)
* NLG(Natural Language Generation)

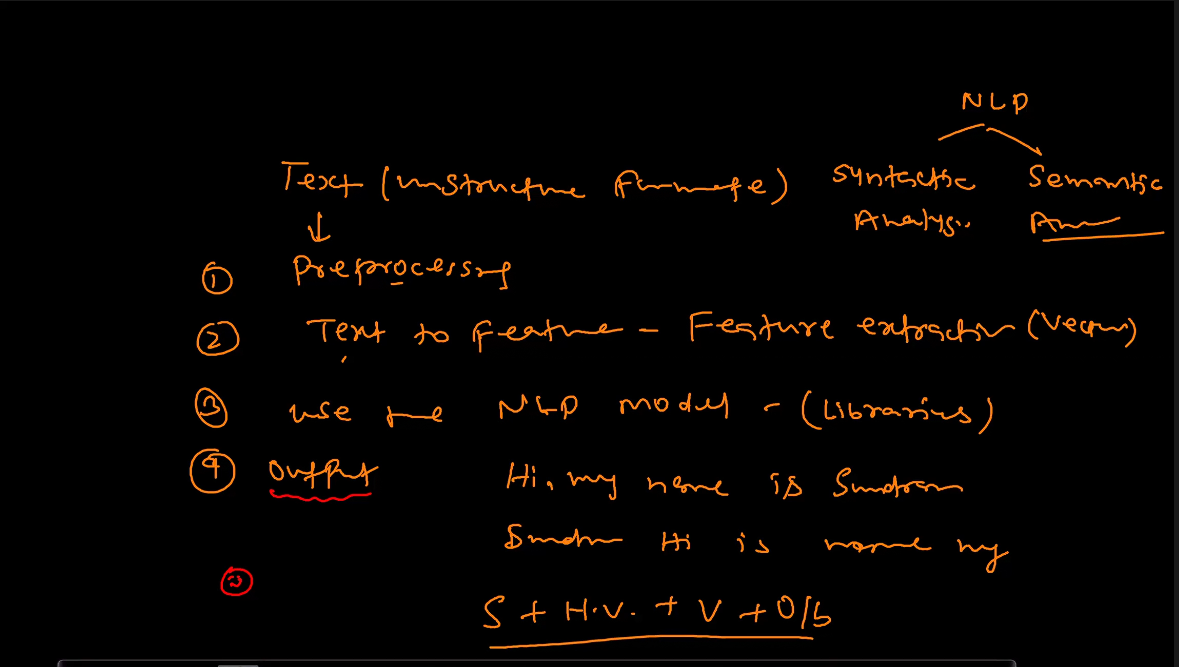
Alan Turning introduced the term NLP in 1955. Alan Turing is often called the **"Father of Computer Science"** and **"Father of Artificial Intelligence"** due to his pioneering contributions in these fields.

The motivation was to decode the Russian language and convert it into English during Cold War between USA and soviet union.



NLP uses unstructured data. The data is extracted through data mining or data analytics.

* Whenever we get text data we do preprocessing.
* Then we convert text into features(numbers) this is called Feature Extraction.
* Use the NLP Model, we can build the algorithms using NLP Libraries(NLTK, Spacy, TextBlob, Spacy, etc.,)
* The output is generated.



NLP can be classified into Rule based and Statistical Based.

NLP has two patterns. – Syntactic Analysis and Semantic Analysis.

Syntactical analysis focus on the structure or the arrangement of words. On the top of it they apply grammatical rules.

Semantic analysis focuses on the actual meaning and interpretation of words.

Tokenising – converting text to vectors(numbers)

**Challenges in NLP:-**

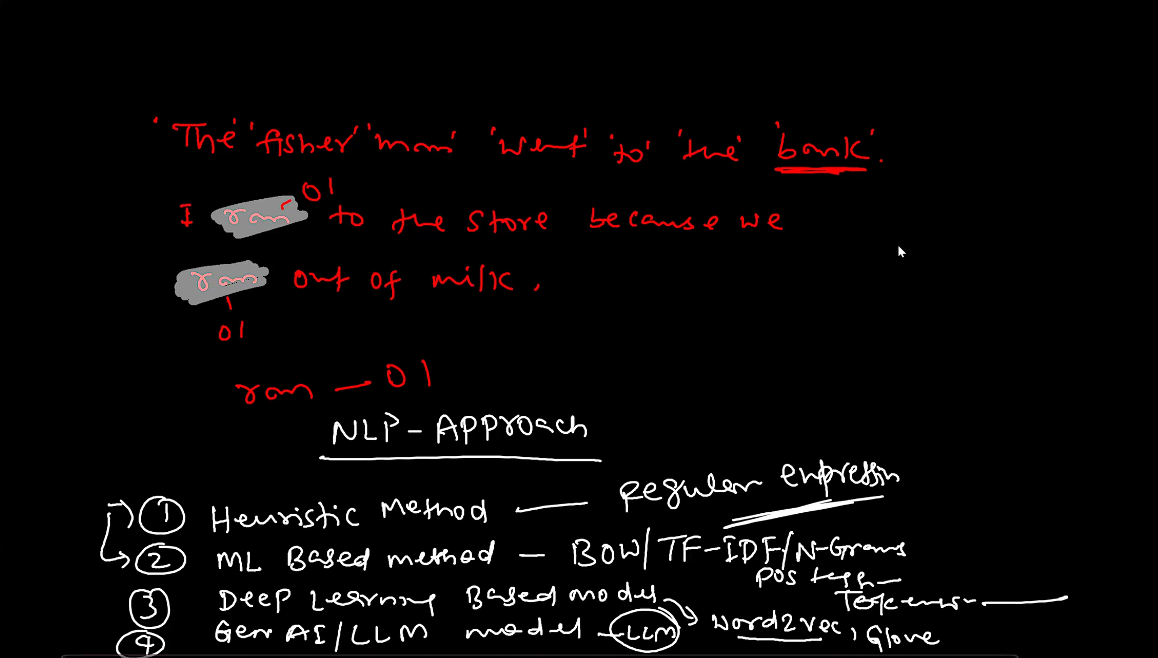
The machine cannot understand the poetry and it just interprets the words.

It does not understand sarcasm.

the sentence **"The fisherman went to the bank"** is an example of **semantic ambiguity** in NLP.

There are broadly 3 approaches or Methods to solve NLP problem:-

* Heuristic Method(Hit and Trial) (uses Regular Expression)
* ML Based Method
* DL based Model
* Gen Al/ LLM Model

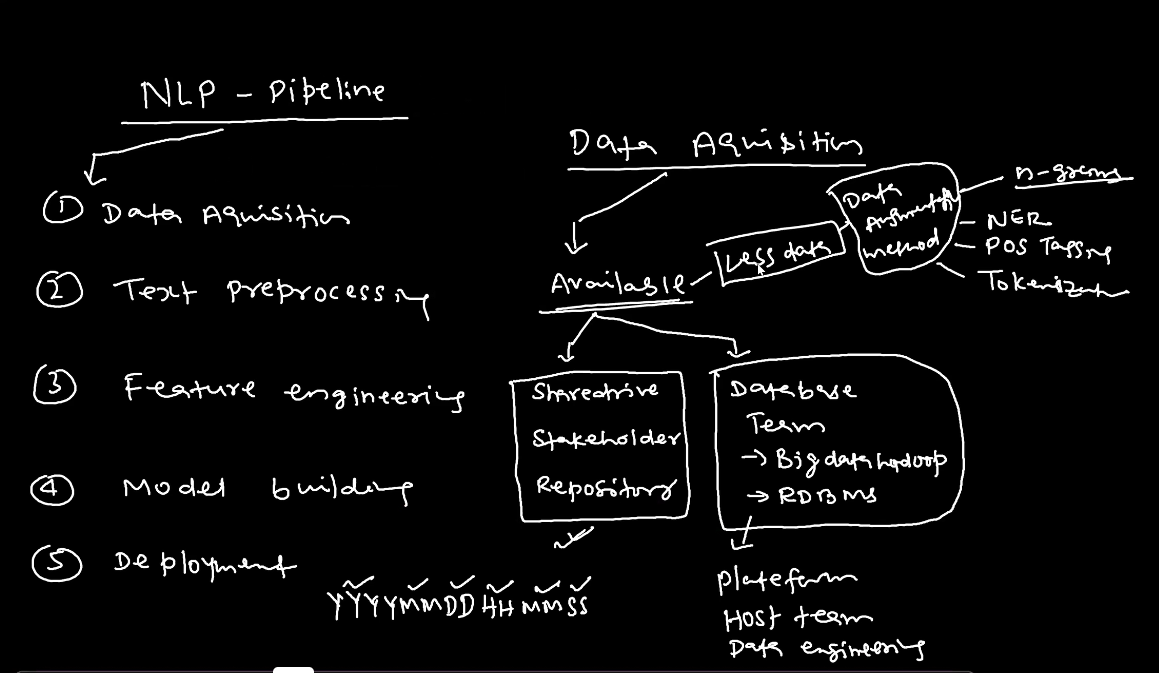


Google(translate) is trained on the BERT model.

**NLP Pipeline:-**

There are 5 broad steps in an NLP pipeline(Machine Learning Architecture):

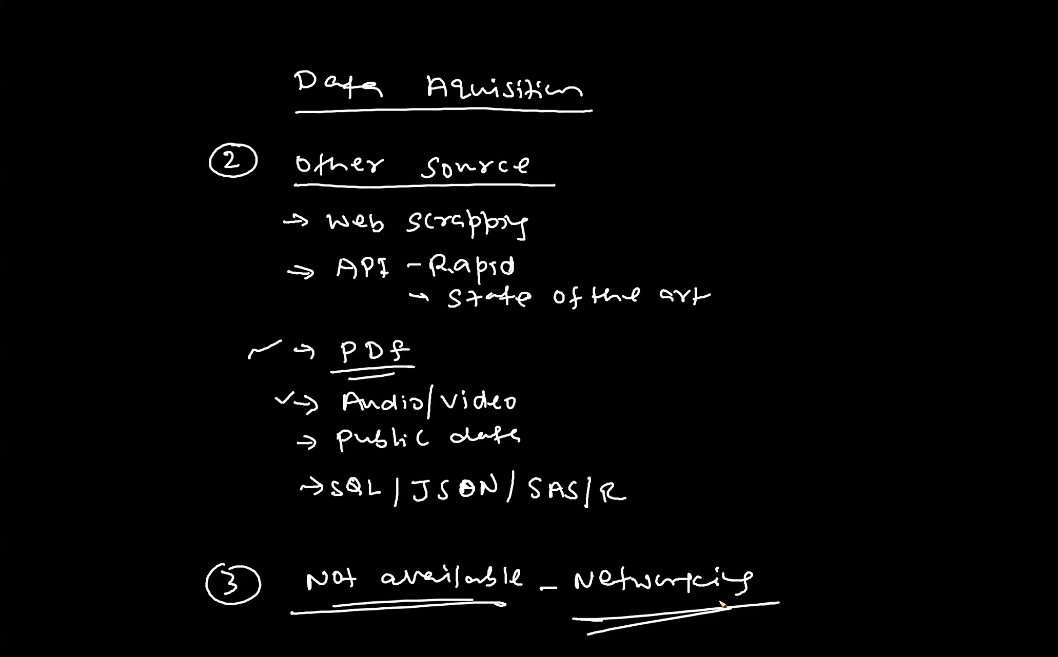
1. Data Acquisition
2. Text Preprocessing
3. Feature Engineering
4. Model Building
5. Deployment



**Data Augmentation:-** when we have very less data we create multiple columns or insights from the existing data.

**Data acquisition:-** we can retrive the data either from

* Within the organization
  + Available to us for access
    - Sharedrive
    - Stakeholders
    - Repository
  + Available with the data team or some other relevant team, for which we have to follow certain organizational procedures to access it.
  + It is available to access but is very less. And thus we have to apply data augmentation to use it effectively.
* Other sources(Not within the organization):- Data is present on other websites and social networking sites. We can use Web scrapping to fetch the data. Other alternatives are **Rapid API** to fetch the data which is use commonly in MNCs.
* Data Not Available or is not visible to us(Problem faced usually in startups):- here we can use our Networking to get the data.



**Text Preprocessing:-**

* Cleaning the data:- html tags, https, email, grammar corrections, spelling mistakes.
* Tokenization:- word, sentence, whitespace, punctuations, stop words.
* Stemming/Lemmatization(Reducing the vocab so that multiple tokens does not gets created):- Converting to lowercase, language detection.
* Handling symbols, numbers, digits.
* POS Tagging.
* Name Entity Recognition (NER)

**Feature Engineering:-** Process of converting Text into Vector. This can be achieved through,

* OHE(can be done but very complex computationally)
* Bag of Words (BOW)
* N-Grams(uni-gram, bi-gram, tri-gram, tetra-gram, etc.,)
* Term Frequency – Inverse Document Frequency (TF-IDF)(Very popular in ML approach). This method creates a sparse matrix and does not gives the Semantic meaning. Unlike, DL methods like Word2Vec and Glove methods. This serves as a main difference between ML and DL feature engineering concept.

**Model Building:-** We then procced to the model building step which is of two types, ML model like (Random Forest, XGBoost, Logistc Regression, Decision Tree, SVM, KNN, Naïve Bayes Theorem, etc.,). NB was actually developed for tasks like Sentiment Analysis.

DL Methods:- RNN, LSTM, GRU, Transformers(BERT and GPT)(before that we need to know seq 2 seq and Encoder Decoder).

**Deployment**:-

Deployment is done to create chatbots, APIs and can be achieved with methods like Flask, Heroku, AWS, Azure, GCP, etc.,

**Stopwords:-**

In Natural Language Processing (NLP), *stopwords* are common words in a language that are usually filtered out before processing text data. These words generally don't carry significant meaning and don't add much value to the analysis of the text. Examples of stopwords in English include words like "the," "is," "in," "at," "and," "to," "of," etc.

The reason for removing stopwords is to simplify the text, focusing on the more meaningful words that contribute more directly to understanding or classifying the content. Since stopwords appear frequently in almost any text, they can add noise to tasks such as text classification, sentiment analysis, or information retrieval.

Here’s a quick example of text before and after removing stopwords:

* **Original sentence**: "The cat is sitting on the mat."
* **After stopword removal**: "cat sitting mat."

Stopword removal helps algorithms process and analyze text data more efficiently by focusing only on important terms. Most NLP libraries, such as NLTK and SpaCy, provide built-in lists of stopwords that you can customize for your needs.

**Stemming:-** It takes the root of the word. Wherever, machine find the words like, player, played, playing, plays etc., it replaces them with the root word that is Play. The benefit of this is that it reduces the tokenization part. Or whenever these words are found there is no need to create 5 tokens for these and only 1 token is suffice.

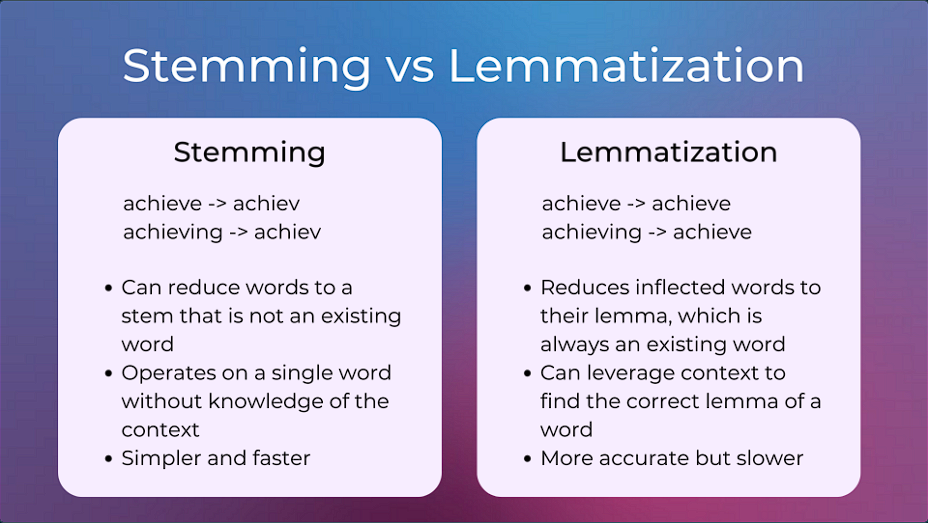
It removes the last few words or suffix. But it can also lead to misspelled or incorrect word.

**Note:-** Stemming should not be done in translation tasks as context in important in those. But can be done in tasks like Sentiment Analysis.

Stemming is very fast as it does not care about forming the correct words.

Lemmatization:- It converts the text to meaningful base form by considering its context. It always gives meaningful word basis the context.

The practical distinction between stemming and lemmatization is that, where stemming merely removes common suffixes from the end of word tokens, lemmatization ensures the output word is an existing normalized form of the word (for example, lemma) that can be found in the dictionary.



Feature Engineering or Feature Extraction:- The 3rd step in our NLP pipeline is the feature engineering or Feature Extraction. We have 3 questions that we may ask:-

1. What is feature engineering and feature extraction and why we need it?
2. Why is it difficult and what is the core idea behind it?
3. What are the techniques required for this?

Converting text into vectors(numbers) is called Feature Extraction. As machine only understands numbers.

Core idea – The sequence of the sentence should remain the same or we should do the encoding decoding.

Techniques Required – OHE, n-grams, BOW(Bag of Words), Tf-IDF, Custom Feature, Word2Vec, Embedding, Glove, etc.,

We have some Common terms to remember:-

Corpus – Consolidated Documents

Documents – No. of sentences/Reviews

Vocabulary – No. of unique words in a Document/Sentence.

Words – No. of words in a Document/Sentence.

Suppose we have a corpus like – I like NLP. NLP is in demand. LearnBay teaches NLP.

Here, we have 3 documents and 8 vocab which is – I like NLP is in demand Learnbay teaches.

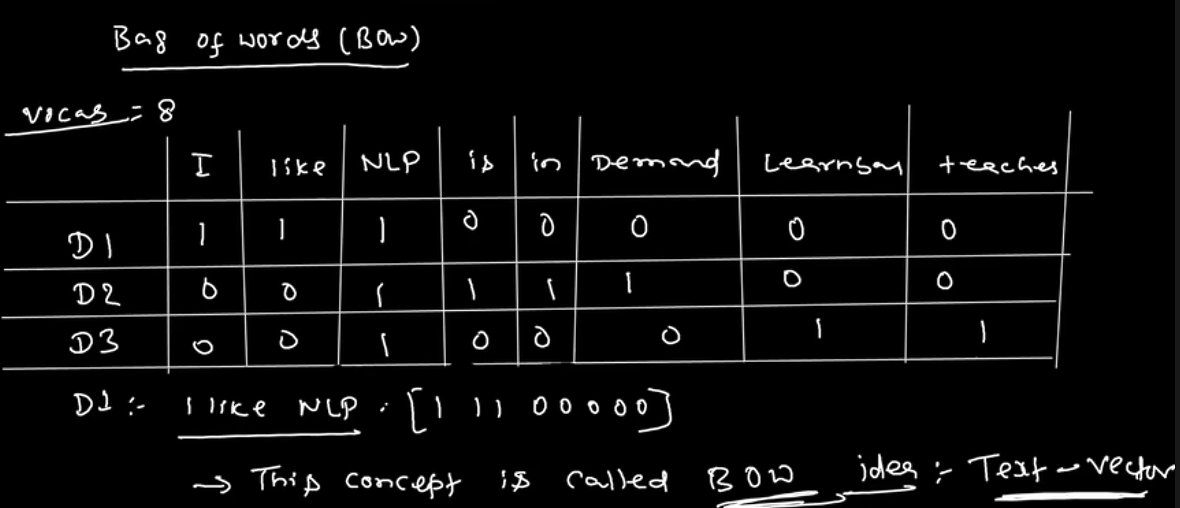
Our task is to convert these vocab into vectors or number. Here, are some approaches we can take –

Approach 1 – OHE:-

This approach is very complex as with the real word data we have thousands of vocab which will create complexity. Also, it creates the sparse Matrix. Also, Ohe only works when all the sentences are of same length which is not efficient.

Approach 2 – BOW:-

In this we arrange the vocab by sentence, which is easier then converting it into OHE.

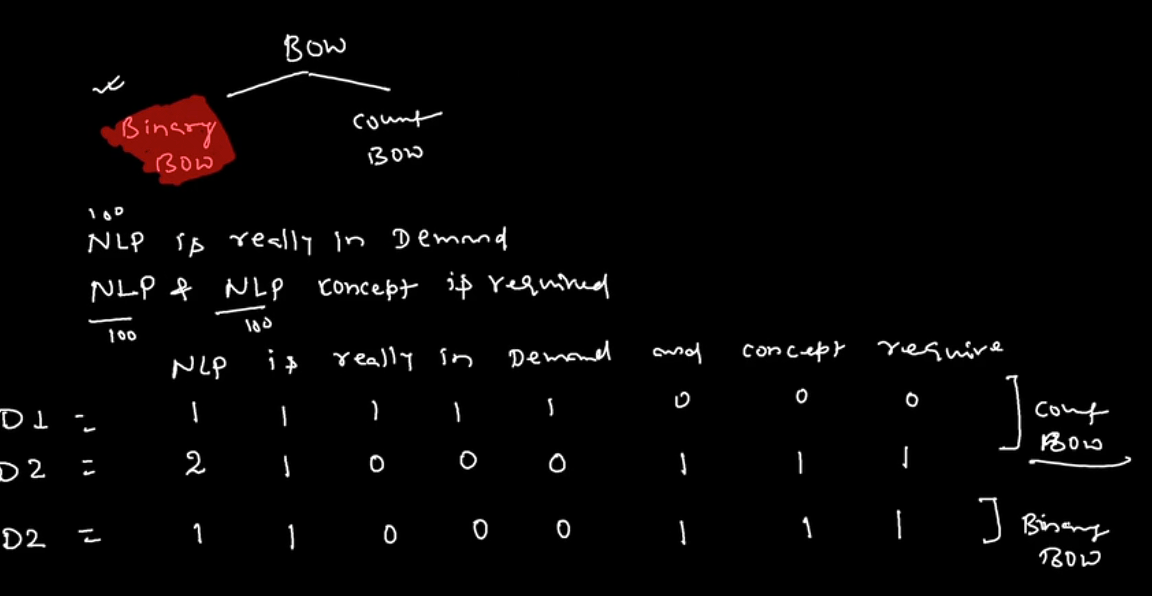


BOW has two types:-

* Binary BOW
* Count BOW

In binary BOW the count doesn’t matter only the occurance does. Reverse is try for Count BOW in which frequency of the words matter.

Binary BOW is much more popular than Count BOW as it helps in effective tokenization.



Pros:-

* BOW is extremely simple and intuitive.

Cons:-

* We have sparsity.
* OOV(Out of Vocab) - It ignore any new words if they are not a part of the vocab.
* With large datasets, applying stop words in BOW results in sentence loss their meaning.
* Also, we are not able to figure out the significant words in BOW.

These problems can be solved by using n-grams.

**N-Grams:-**

Uni-gram – it is by default BOW only. Ngrams.range = (1,1) – BOW

Bi-gram – n = 2, so in any problem which includes word pairs like positive-negative, we use bi-gram.

Tri-gram – n= 3, three consecutive words.

n-grams also helps us to perform Data Augmentation.

Even after we remove stop words after applying ngram. The meaning of the sentence remains intact.

**Problem with n-gram:-**

Ngrams solves the problem with stop words.

But fails to solve the sparsity and word significance problems.

It also faces OOV problem.

Hence, we shall not use either BOW or n-grams. We shall use TF-IDF

**TF-IDF(Term Frequency – Inverse document Frequency):-**

We can easily use BOW or ngrams to convert words to numbers but our purpose is to identify the significant words(or documents) too.

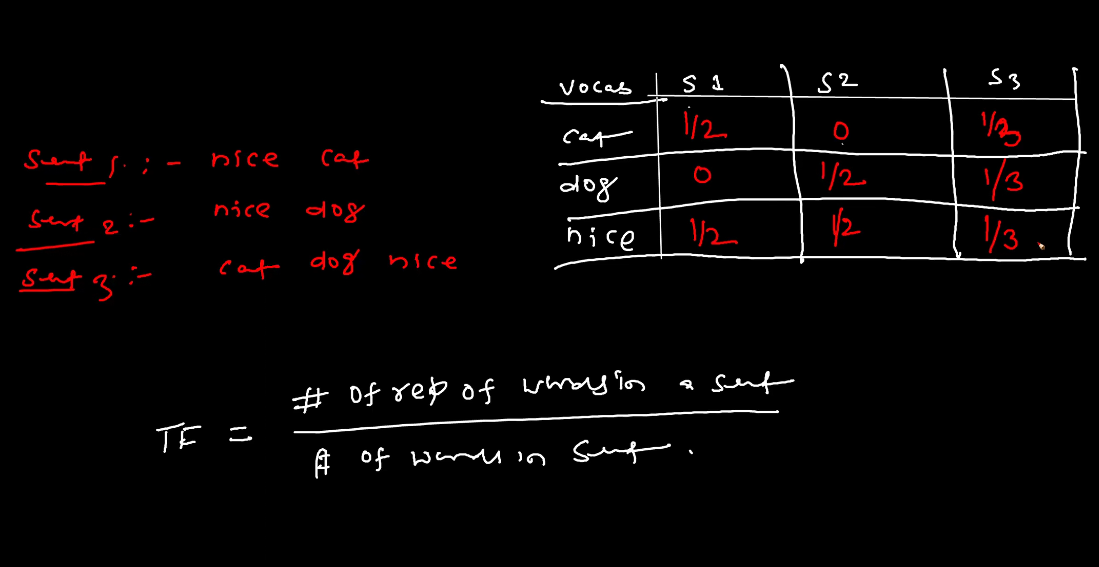
Term frequency means the output of BOW. The formula is:-

Tf = Number of representation of words in sentence.

Number of words in sentence

We can say that TF is the probability value. Hence, 0<=TF<=1

## Term frequency measures the local importance of words or how frequently it is coming. If a word is coming frequently it must be important.





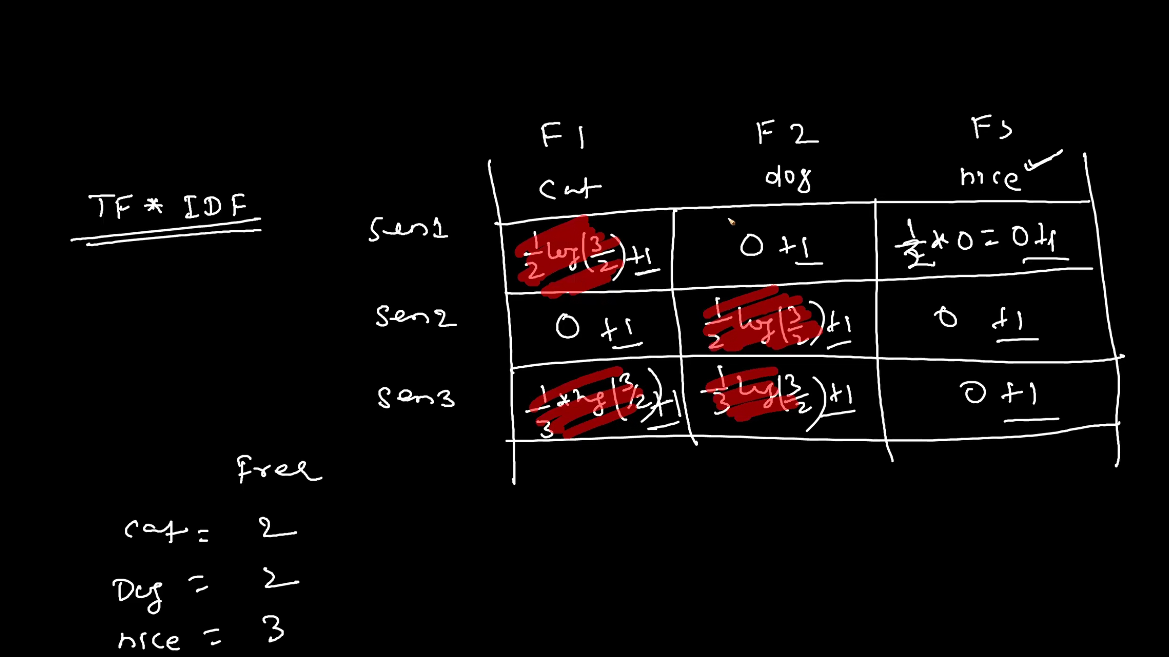
IDF:-

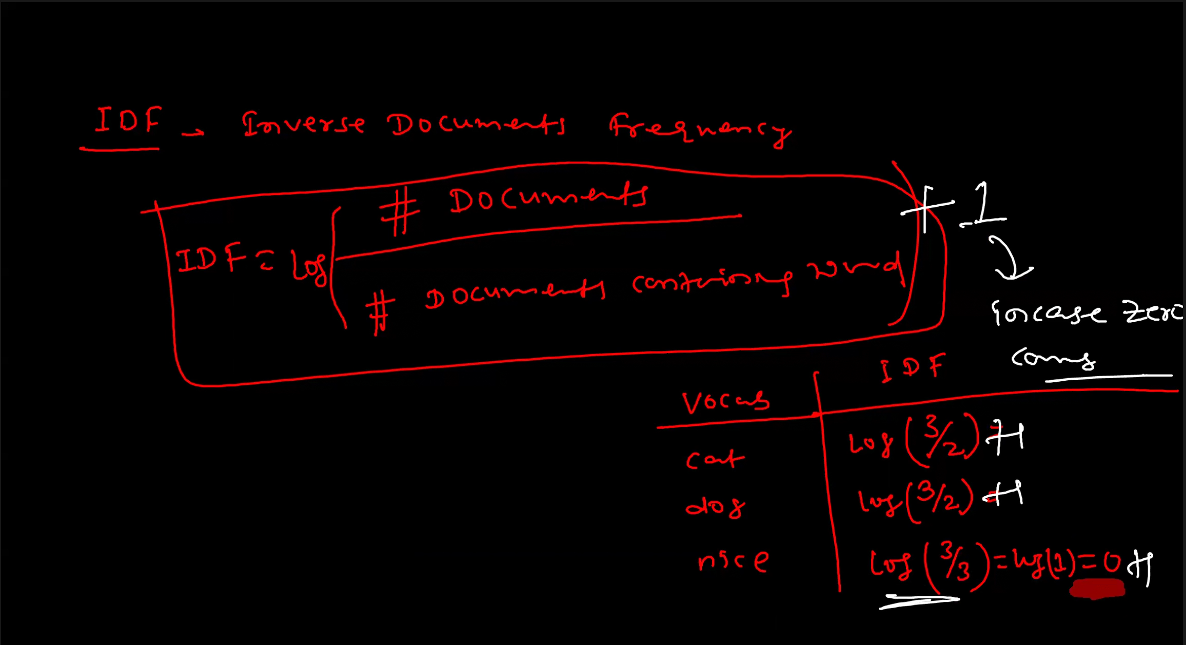
IDF = Total number of documents .

Total number of documents containing word



The denominator should always be less than or equal to Numerator. Hence, we only consider the number of documents that contain the desired word and not the total occurrence of the word in a particular document.





Advantages of using TF-IDF:-

Gives better results/information then BOW.

Disadvantages:-

We still face sparsity issue here.

OOV is still present

Semantic meaning cannot be found out. (If we give joy instead of happy(on which machine is trained, it will not treat it as similar.)

It is not possible to deal with sparsity using ML techniques. We have to deep dive in DL techniques to deal with sparsity like Word 2 Vec and Glove.