Natural Language Processing.

History.

"Can machines think?"

— Alan Turing.

Turing Game

→ 3 Participants

→ 2 People & 1 computer.

→ one is an interrogator & other person → Comp. → Person.

→ Computer → will try to make believe that it's a human response.

Person → will try to make believe that it's a machine.

ELIZA

→ early NLP system capable of carrying a limited Conversation

ELIZA: What's your Name?

RAM: My Name is Ram.

ELIZA: Hi I'm Eliza. what do you want to talk about?

Mid 1950's - 1960's

Birth of NLP & Linguistics

Main Aim ↓

→ Mostly hand Coded Rules

→ No Major Advancements

Machine Translation:

1960 - Mid 70's → Dark Era of NLP.

because of No Proper development in Machine translation in Previous era → (1950-1960's)

- People started to believe it as hype & machine / Comp. to be incapable of language translation.

1970's — 1980's Slow Revival of NLP.

→ Some Research Activities Revived.
→ emphasis was still on linguistics.
→ work on toy problems

(I) 1980's & 1900's Statistical Revolution

→ by this time computational power increases substantially.
→ Data-driven / statistical approach

(II) 2000's

Computational Power ↑↑.

2010's

Evolution of Deep Learning

Neural N/w.
word2vec
Glove
fast Text
BERT

2017 +

Transformers
self attention
GPT ....

NLP.

Natural Language Processing.

This is a NLP class.

↑
which class

NLP

NLP is a field of computer science, Artificial Intelligence & linguistics concerned with the interaction between computers & human languages.

specially, the process of a computer extracting meaningful information from Natural language I/P and/or producing Natural Language O/P.

– wikipedia.

why NLP ?

① Answering Questions

→ what is the next train from the city after 3:00 PM Train?

→ I'm a Masters DS student, which classes do I have today ?

→ who is Alan Turing ?

② Information Extraction:

→ we decided to meet for our class at 10:00 AM tomorrow in Learnbay class.

To do: class Meeting
Time: 10:00 AM
Venue: Learnbay class.

③ Translation

मेरा नाम राम है ⇒ My Name is Ram

Hindi ⟺ English

④ Text Summarization

→ extract keywords
→ create an abstract of the entire article/Book etc.

NotebookLM.

⑤ Sentiment Analysis

Processing reviews of Product/Services & extract sentiment of people and it.

Level of Analysis in NLP.

1.) Morphological Analysis -

→ Breaking down the words into smallest meaningful units → morphemes

→ It analyzes structure of word - Prefix, Suffix, root.

eg. →

cats → "cat" + "s" → Plural

running → "run" + "ing" → Suffix

unhappiness → "un" + "happy" + "ness"
↓ Prefix ↓ Root ↓ Suffix

② Syntactic Analysis

→ This examines the grammatical structure of sentences.

→ how words combine to form a phrase or a sentence.

→ A/C to grammar.

Eg. → ① The Cat sat on the mat.
↓ Subject ↓ verb ↓ Prepositional phrase.

② determining Parts of speech.
POS Tagging

③ Sentence structure.

③ Semantic Analysis

→ focuses on meaning of words, phrases & sentences

→ Going beyond just grammar to understand what the text actually means.

Eg. →
① River bank } different meanings.
Money bank }

② Resolve pronoun References.
Amit wrote the exam. he scored 90%.

(3) word sense disambiguation and understanding context.

(4) Discourse Analysis

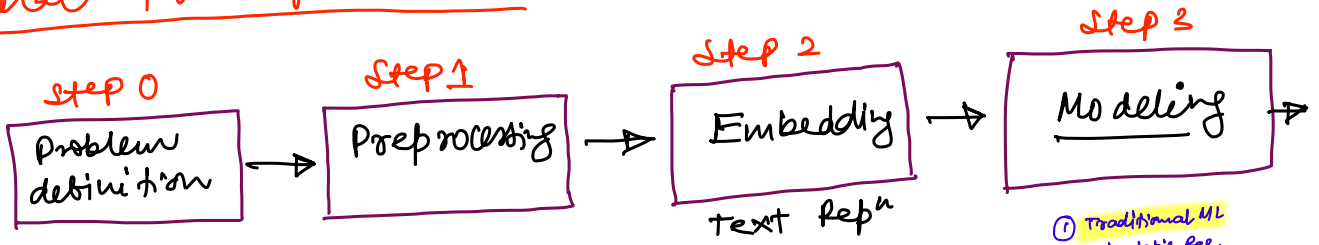
→ Examines how meaning is constructed across larger units of texts.

→ looks at the relationship betn sentences & paragraphs.

→ ↓ to understand overall coherence & context.

eg.→ how sentences connect in a conversation or a document.

General flow of NLP.



- → Lowercasing (A → a)
- → Removing punctuations (!, .)
- → Tokenization (This is NLP) → ("This", "is", "NLP")
- → Stop word removal → removing uninformative words → ("is", "the", "a", "and")
- → Lemmatization / Stemming → runs → run, running → run ← Removing suffix ← Root word.
- → Handling special tokens → Removing URL's, emails etc.
- → Spelling correction → fix typos.

eg.→
- → TF IDF
- → BOW (Bag of words)
- → word2vec
- → Glove
- → BERT
- → GPT etc.

(I) Traditional ML
- → Logistic Reg.
- → SVM
- → RF

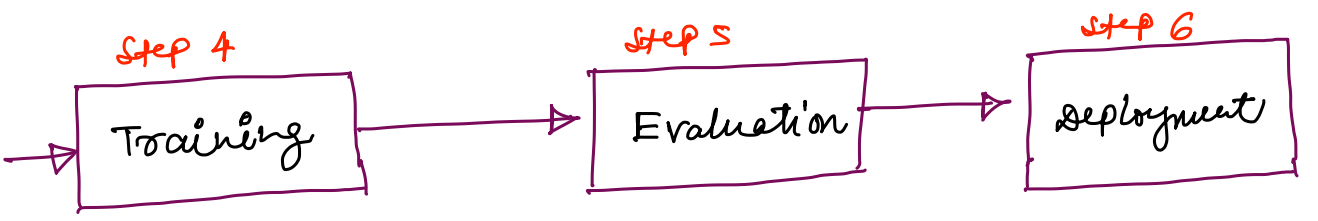
I/P → TF IDF, BOW, eg. word vectors

(II) Deep Learning

Embedding layer → LSTM/GRU → Dense → O/P Layer

(III) Transformer Models

Pre-trained models BERT, RoBERTa et



optimiser → Adam SGD etc.

loss → binary_crossentropy etc.

Based on the task I'll test my model.
e.g.→
Classification → Accuracy, f1 score
Sentiment Recall
NER → span level f1.
Generation → BLEU, ROUGE

* Wrap model in API → flask FastAPI
* use cloud → GCP, AWS, Az
* Hugging face spaces

Preprocessing Techniques

1. Lowercasing

→ each character in your text is converted to lower case.

→ "Apple → "apple"

"ORANGE" → "orange"

A B C D
↓ ↓ ↓ ↓
a b c d

→ .lower() → uses unicode mapping.

This step normalizes text because most NLP models treat "APPLE" and "apple" as different tokens unless explicitely train otherwise.

2. Tokenization

→ Breaking text into smaller units → tokens.

→ tokens can be words / subwords / sentences.

I don't like Java.

"I" "do" "n't" "like" "Java" "."

subwords

→ we can use RegEx or libraries like SpaCy, Huggingface etc.

↓

include langage rules, abbreviation handling etc.

3. Removing Punctuation

→ Removing punctuation characters (. , ! ? ; :)

→ often using regex.

→ it checks for each character's unicode category to determine if it falls in punctuation category.

→ can help in bag of words (BOW) models but we can skip this in tasks which require sentence structure.

4. Removing Stopwords

→ Removing words that don't contribute significantly to the semantic meaning of the sentence.

→ NLTK & SpaCy provide a list of stopwords & it matches each token against these list & Remove.

→ goal → Remove noise & Reduce dimensionality.

5. Stemming

→ Reduces the word to its root form by removing prefixes or suffixes.

"ing".

→ Rule based & doesn't care about grammer.

Root

Relational → Relate

Arguing → Argu → ~~ize~~

↓

not grammatically correct.

6. Lemmatization

→ It uses a vocabulary & morphological analysis to return the dictionary form of a word

↓

Lemma

→ better —lemma→ good

→ It also looks at the POS (Parts of Speech) to convert word into lemma.

eg. →

		Lemma
meeting (Noun)	→	meeting
meeting (Verb)	→	meet

7. Removing Numbers

→ Stripping out digits using Regex.

→ Context sensitive.

→ It match numerical characters (unicode) & replaces them with an empty string.

8. Removing extra spaces

→ Removing extra spaces introduced in the text.

→ mostly prevalent in text obtained from HTML (web) scraping etc.

→ `' '.join(text.split())`

9.) Removing special characters

→ Removing characters like #, @, &, * etc.

→ We do it using Regex.

10.) Handling Contractions

→ Contractions like don't are expanded to "do not", "you're" → "you are", I'm → "I am".

→ use Python libraries → contractions, textsearch.

→ Internal logic checks each token for expands by looking for Pre defined mappings.

11.) POS (Parts of Speech) Tagging.

→ POS tagging assign grammatical tags
↓
Parts of speech (Noun, Pron etc)

→ In background libraries uses different models to label each word based on features
↓
suffix, surrounding words, capitalization etc.

NLTK → Perceptron
spacy } bi-directional LSTM
stanza } Transformers

12. Name Entity Recognition (NER)

→ NER identifies & classifies entities like people, organisation & location etc.

→ under the hood

(a) Rule Based → Pattern of POS chunks.

(b) Context.

(c) Deep learning models → Bi LSTM, Transformers.

13.) Handling missing data.

→ Predict next word / missing word

→ Replace → mode.

14.) spelling correction

→ Pretrained model on Corpus of words

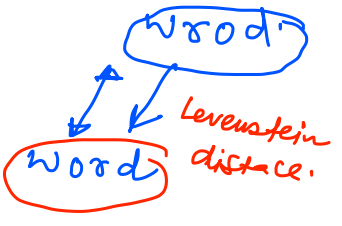
→ Textblob

→ probabilistic spelling model.

↓

trained on large corpus of words.

word is a 4 lettered word



15.) Handling emoji's / emoticons

 → :slightly_smiling_face:

→ unicode mapping

→ emoji

Libp
↓
emjix

16.) Handling URL's

→ Regex Patterns like https://
http://

Can be detected & removed.

www.
.com

17.) Removing HTML Tags

→ HTML is parsed with libraries
like BeautifulSoup.

→ DOM tree

→ get_text()

↳ recursively strips tags()

↓

& Left with textual Content