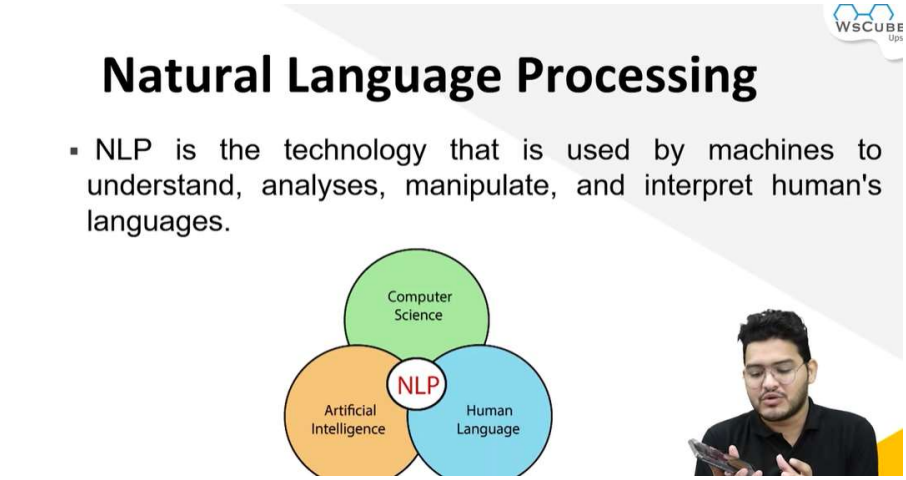
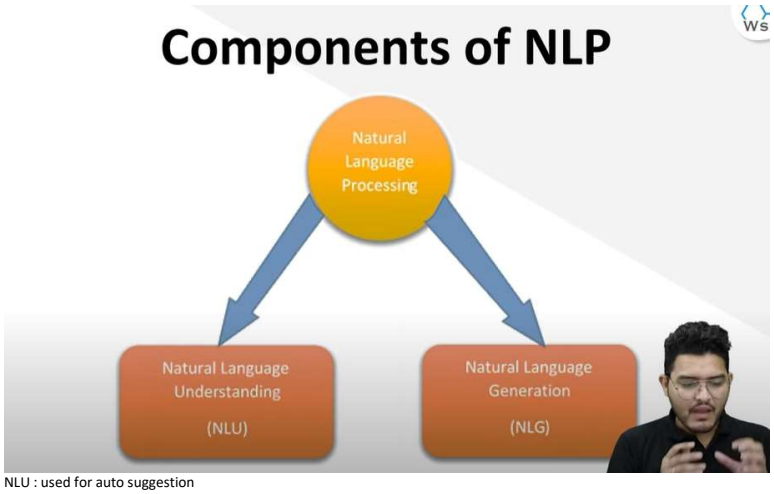
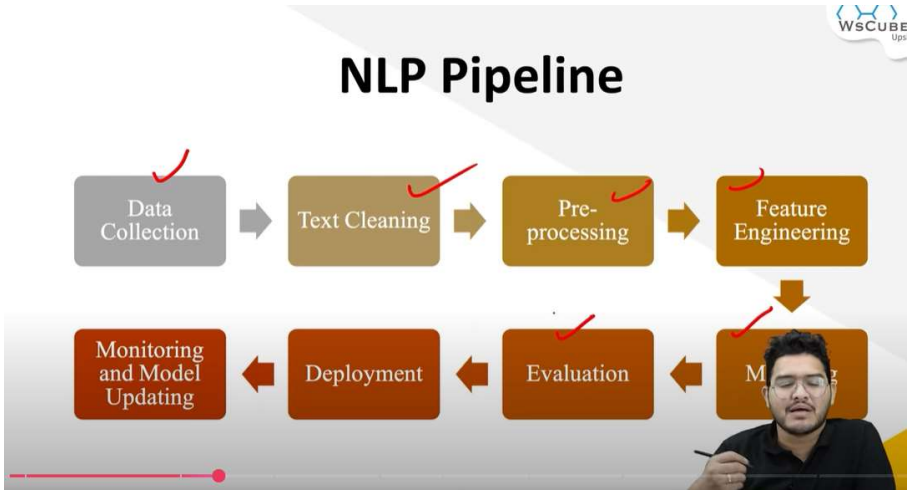
03 April 2025 21:40





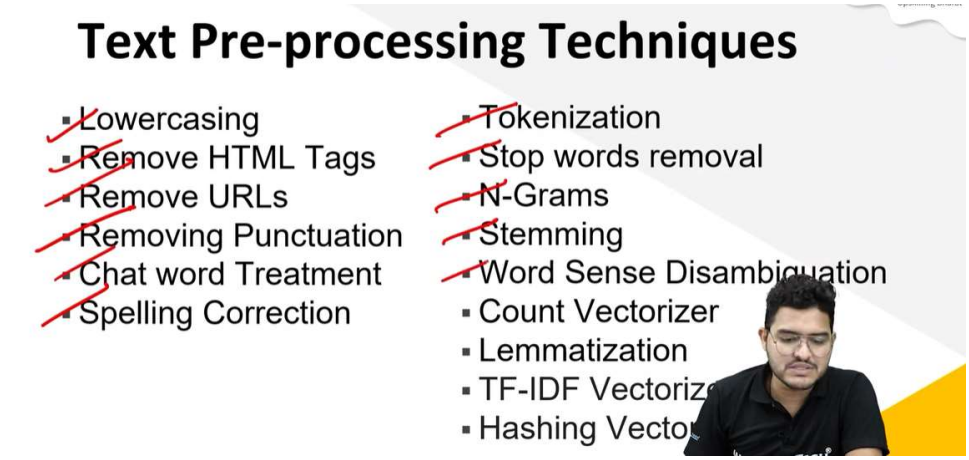
NLU : used for auto suggestion
NLG: Used for translation

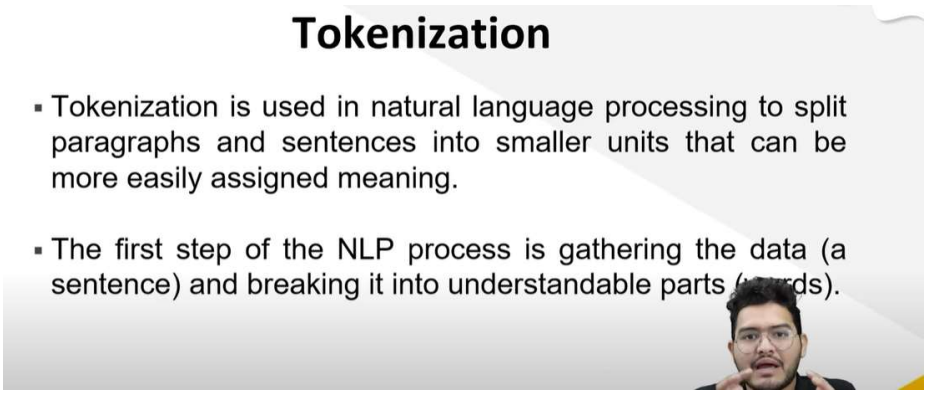


Understanding textual data

Hierarchy of words: gives word with how many times they occurred
Tokenization: separate the sentence into words
Vocabulary: different word with similar meaning
Punctuation : ,./
Part of speech: word which are noun, adjective, pronoun
Root of word: enjoyable --> enjoy
Base of word: similar to root word
Stopwords: and , is , or, the

Text Pre-processing techniques





Sentence and word tokenization

```
[25] from nltk.tokenize import word_tokenize, sent_tokenize

[26] sentences_tokenized  = sent_tokenize(x)

[27] sentences_tokenized

['Natural language processing (NLP) is a subfield of computer science and especially artificial intelligence.',
 'It is primarily concerned with providing computers with the ability to process data encoded in natural language and is thus closely related to information retrieval, knowledge representation and computational linguistics, a subfield of linguistics.']

[29] word_tokenized = word_tokenize(x)

word_tokenized

['Natural',
 'language',
 'processing',
 '(']
```

Stop word Removal

```
[42] from nltk.corpus import stopwords
     from string import punctuation
```

```
[38] nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
[39] stop = stopwords.words('english')
```

```
stop
```

Show hidden output

```
[41] var_new = word_tokenize(x)
```

```
[44] list(punctuation)
```

Show hidden output

```
[45] stop_word_list = stop + list(punctuation)
```

```
[46] for i in var_new:
       if i not in stop_word_list:
         print(i)
```

Stemming

Stemming

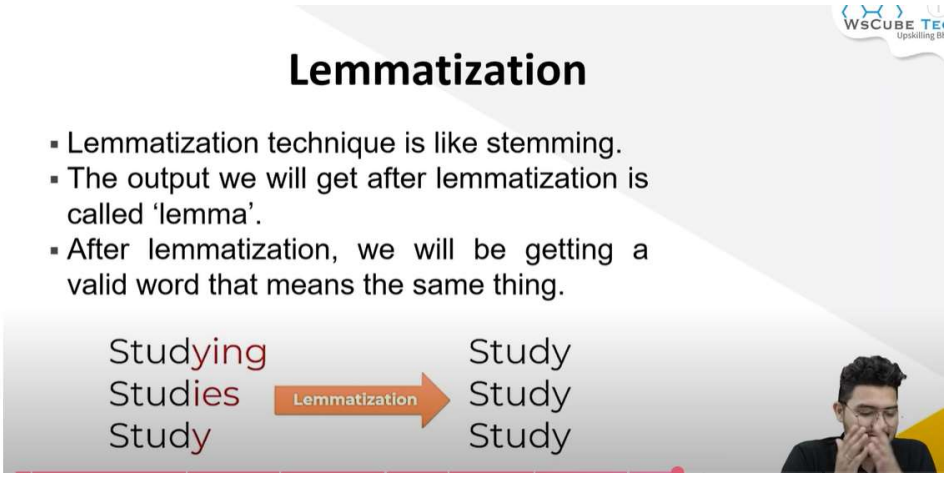
```
[49] from nltk.stem import LancasterStemmer, RegexpStemmer, PorterStemmer, SnowballStemmer
```

```
[52] l = LancasterStemmer()
     r = RegexpStemmer('ing')
     p = PorterStemmer()
     s = SnowballStemmer('english')
```

Most frequently PorterStemmer is used

```
p.stem("changing")
```

```
'chang'
```



Lemmatization

```
[54] from nltk.stem import WordNetLemmatizer
```

```
[55] wl = WordNetLemmatizer()
```

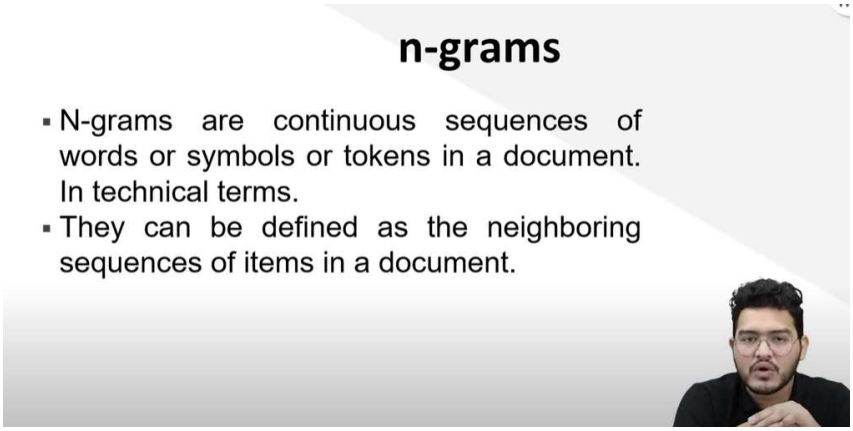
```
[57] nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
True
```

```
wl.lemmatize("mice")
```

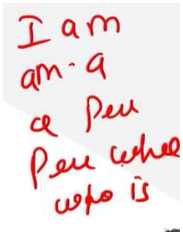
```
'mouse'
```

n- grams



Give auto suggection for sentence after writing one word
Suppose if I write I, then probably it'll give me next word love

For bi-gram
Ex:




```
[68] from nltk.collocations import BigramCollocationFinder, TrigramCollocationFinder, ngrams
```

```
[70] b = BigramCollocationFinder.from_words(tokenized_word)
```

```
b.ngram_fd
```

```
FreqDist({('I', 'am'): 3, ('am', 'Kunal'): 1, ('Kunal', 'Patil'): 1, ('Patil', 'I'): 1, ('am', 'a'): 1, ('a', 'Product'): 1, ('Product', 'Analyst'): 1, ('Analyst', 'I'): 1, ('am', 'sincere'): 1, ('sincere', 'and'): 1, ...})
```

```
[72] t = TrigramCollocationFinder.from_words(tokenized_word)
```

```
[76] t.ngram_fd
```

```
FreqDist({('I', 'am', 'Kunal'): 1, ('am', 'Kunal', 'Patil'): 1, ('Kunal', 'Patil', 'I'): 1, ('Patil', 'I', 'am'): 1, ('I', 'am', 'a'): 1, ('am', 'a', 'Product'): 1, ('a', 'Product', 'Analyst'): 1, ('Product', 'Analyst', 'I'): 1, ('Analyst', 'I', 'am'): 1, ('I', 'am', 'sincere'): 1, ...})
```

```
[79] n = ngrams(tokenized_word, 2)
```

```
[80] for i in n:
       print(i)
```

```
('I', 'am')
```

Vectorizer similar to Tfidfvectorizer in sklearn

Word disambiguation

Mouse is running
Mouse is working

Word Disambiguation

```
[81] x = "Sunrise (or sunup) is the moment when the upper rim of the Sun appears on the horizon in the morning,[1] at the start of the Sun path. The term can also refer to the entire proce
```

```
[82] from nltk.wsd import lesk
     from nltk.tokenize import word_tokenize
```

```
[83] l = lesk(word_tokenize(x), 'sun')
```

```
[86] l.definition()
```

```
'the star that is the source of light and heat for the planets in the solar system'
```

```
[93] z = "A computer mouse (plural mice, also mouses)[nb 1] is a hand-held pointing device that detects two-dimensional motion relative to a surface. This motion is typically translated in
```

```
[92] from nltk.wsd import lesk
     from nltk.tokenize import word_tokenize
```

```
[99] l = lesk(z, "computer")
```

```
l.definition()
```

```
'a machine for performing calculations automatically'
```