CIE-I

SAQ's

Q1) Define speech & NLP?

Ans) Speech:

Basic meaning of speech, means of communication used by people.
speech refered in NLP means,
giving computers the ability to understand text & spoken words in much the same way human beings can.
Translation of spoken language to text is called as speech recognition.

Natural Language processing (NLP):

NLP is a ML technology that giver computer. the ability to learn, understand, analyze, manipulate, interpret natural human language.
The Input & output of an NLP system can be speech or text.
To make interaction between computers & humans, computers need to understand natural languages used by humans.

Q2) Define Stemming?

Ans Stemming is reducing words to their root form by removing suffixes & prefixes, such as "running" becoming "run". This method is helpful when working with text data that has many different versions of the same word.

Example: Input: "running, runner, ran"
output: "run, run, run"

Q3) What are Regular Expressions?

Ans Regular Expressions(RE) are an algebraic way to describe Formal Languages.

→ RE represents a set of strings having certain pattern.

→ A regular expression is built up of simpler RE's using defining rules.

• simple Definition for RE over alphabet "Σ"

- Σ is a RE
- If a ∈ Σ, a is a RE
- Or If E_1 & E_2 are RE's then $E_1 | E_2$ is a RE

concatenation: E_1 & E_2 = E_1E_2

Kleen closure: $E = E^*$

Positive Closure: $E = E^+$

LAQ1

Q1)
(a) What is Tokenization? (b) Challenges of NLP?

Ans Tokenization is the process of breaking text into individual words or phrases, also known as "tokens".
This technique is useful when working with text data that needs to be analyzed at the word level, such as Text classification.

Example: Text Normalization

Input: "The quick BROWN Fox Jumps OVER the Lazy dog."

```
text : "The quick BROWN Fox Jumps OVER the Lazy dog."
# split text by whitespace
tokens = text.split()
print (tokens)
```

Output: ["The", "quick", "BROWN", "Fox", "Jumps", "OVER", "the", "LAZY", "dog"]

Advantages:

1) It allows for analysing & manipulating individual words in the text data.
2) It can improve the performance of NLP algorithms that rely on word level analysis.

Disadvantages:

•) It can lead to loss of information, as the meaning of a sentence or text can change based on the context of words.

Types of Tokenization:

As we know, Tokenization helps split the original text into characters, words, sentences. etc. depending upon the problem at hand.

→ If you split text data into words, it's called Word Tokenization.

Consider the following sentence/raw text.

"Let us learn tokenization."

A word based tokenization algorithm will break the sentence into words.

["Let", "us", "learn", "tokenization."]

→ If the document is split into sentences, then it is called Sentence Tokenization.

→ splitting the document into individual characters is known as character Tokenization.

→ A character based tokenization algorithm will break the sentence into characters.

["L", "e", "t", " ", "u", "s", " ", "l", "e", "a", "r", "n", " ", "t", "o", "k", "e", "n", "i", "z", "a", "t", "i", "o", "n"]

Q2) (a) Explain N-gram?

(b) What is Stochastic Based Tagging?

Ans (a) N-Gram:

The is one of the simplest approaches to language modelling. Here a probability distribution for a sequence of 'n' is created, where 'n' can be any number and defines the size of the gram.

If n = a, a gram may look like:

"can you help me".

There are different types of N-Gram models such as unigrams, bigrams, trigrams etc.

- A model that simply relies on how often a word occurs without looking at previous words is called unigram.
- If a model considers only the previous word to predict the current word, then it's called Bigram.
- If two previous words are considered, then it's a trigram model.

Text	N-gram.
Data.	1-gram
Great Idea.	2-gram
I am Fine.	3-gram
Nice to meet you.	4-gram.

For example:

For the sentence "The cow jumps over the grass". If N=2 (known as bigrams). Then the n-grams would be:

- The cow
- cow jumps
- jumps over.
- over the
- the grass.

If N = 3, the n-grams would be:

- The cow jumps
- cow jumps over.
- jumps over the.
- over the grass.

So, you have 4 n-grams in this case.
when N=1, this is called as unigrams.
when N=2, this is called as Bigrams
when N=3, this is called as trigrams.
when N>3, this is usually refered to as Four grams or five grams & so on.

So how many n-grams can we have in a sentence?
If X = number of words in a given sentence K, the number of n-grams for sentence K be,

$$N\text{-grams } K = X - (N-1)$$

(b) Stochastic POS Tagging :

Another technique of tagging is Stochastic POS Tagging.

The model that includes frequency (or) probability can be called stochastic. Any number of different approaches to the problem of part-of-speech tagging can be referred to a Stochastic tagger.

The simplest stochastic tagger applies the following approaches for POS tagging

word Frequency Approach –

The stochastic taggers disambiguate the words based on the probability that a word occurs with a particular tag. we can also say that the tag encountered most frequently with the word in the training set is the one assigned to an ambiguous instance of that word.

Tag sequence probabilities –

It is another approach of Stochastic tagging, where the tagger calculates the probability of a given sequence of tags occurring. It is also called n-gram approach.

Here is an example of how a statistical pos tagger might work.

- collect a large annotated text & divide it into training & testing sets.
- Train a statistical model on the training data. using techniques such as maximum likelihood or hidden markov models.
- use the trained model to predict the pos tags of the words in the testing data.
- Evaluate the performance of the model by comparing the predicted tags to the true tags in the testing data
- use the trained model to perform pos tagging on new, unseen text.

Disjunction of characters:

The string of characters inside the braces [ ] specifies a disjunction of characters to match. The RE [wW] matches pattern containing either w or W

RE	Matches.
[wW] ood check	Woodcheck, woodcheck.
[abc]	'a', 'b', 'c'

Morphology

→ It is the study of the way words are build up from smaller meaning bearing units, morphemes.

→ A morpheme is often defined as the minimal meaning bearing unit in a language.

→ For example - The word fox consists of a single morpheme (the morpheme fox) while the words cats consists of 2. The morpheme cat & The morpheme cats/eat-s

→ The two broad classes of morpheme. The stems

(i) The stems - The main morpheme of word suppling the main meaning.

(ii) Affixes - It The additional meaning of various kinds. There are further divided into prefixes & suffixes.

Suffix examples: eats: eat-s

prefix " : un-buckle.

Concatinative morphology:

prefixes & suffixes are often called as concatinative morphology.

The two broad classes of ways to form words from morphemes

1) Inflection – The combination of word stem with Grammatical morpheme usually resulting in a word of the same class as the original term & usually filling some syntactic function like agreement

2) Derivation – The combination of word stem with Grammatical morpheme usually resulting in a word of different class often with a meaning hard to predict exactly

Inflection of Noun in English:

An affix making plural cat (-s)
box (-es)