Module 2: Word Level Analysis

Q. Write a short note on morphological parsing and Morphology Analysis.

=> Morphological parsing is the task of recognizing the morphemes inside a word.

- Morphemes are the minimal meaning-bearing units in a language.

- Example: Mangoes

Mango es → ①

Here there are two Morphemes

- Morphemes can be Stems (Root word) or an Affix

- Now this Affix is divided into three parts
An affix can be prefix (eg reform) or Suffix (eg. loved) or infix (passersby).

- So here Mango is a Stem and es is a suffix because it is attached after the main word

- Following are the requirments for building a morphological Parser

1. Lexicon: It includes the list of stems and affixes along with the basic information about them.
eg Stem is a noun stem or a verb stem.

Morphotactics:

Morphotactics:

Morphotactics has a set of rules by the help of which it decides the ordering of words.

example: Use able ness

Useableness ✓

able use ness

ableuseness ×

Orthographic rules:

These spelling rules are used to model the changes occuring in a word.

example: lady+s = ladys ×

lady+s = ladies ✓

- The study of formation of words is called morphology.

- Some words are self sufficient they have there own meaning example- camera, pen

- Some words if devided has there own meanings example: Showcase = Show & case both words have there own meanings.

- Some words if ~~combine~~ dont have any meaning but if they are combined with a word it becomes a meaningful word.

  example: ing has no meaning but if combined with love its loving which have there meaning.

- So basically there are different words which if used in a right way we can get a meaningful word.

- So analysis is Studying indetail and Analysis of Morphology is Morphology analysis.

Q1 Write a short note on Inflectional and Derivational morphology.

=> Before Inflectional and derivational morphology we need to understand what are morphemes. Morpheme is a word or a part of a word that has meaning and a morpheme cannot be devided further into meaningfull units.
example of morpheme : cat
If we try to devide morpheme more it will be a meaningless result.

There are two types of morphemes
① Free Morphemes ② Bound morphemes

① Free Morpheme : Free morpheme is a morpheme which has its own meaning or it has its complete meaning example : fan, camera etc

- Free Morphemes are of two types lexical morphemes and grametical morphemes

- Lexical morphemes are the picture words they are noun, adjective, verbs, adverbs
example : black, yello, chair
every year new and new lexical morphemes are added in a language.

- Gramatical morphemes are grammer words which are limited in each and every language.

These morphemes dont change frequently like Lexical morphemes they are preposision, conjunctions, etc

② Bound Morphemes: These morphemes are of two types Inflectional and Derivational.
But before going further we must know what is bound morphemes.

Bound morphemes are those morphemes whose meaning is not complete in themself. And that is the reason why they depend on the free morphemes for meaning

Now Affeexes are bound morphemes and Affiexes are of three types prefix, Postfix, suffix
prefix → Because

Suffix → loveable

Infix → passersby

Inflectional morpheme: Inflectional morpheme is one which when attached to a root word dosent change its class

$$\underset{\text{Noun}}{\underline{\text{Book}}} + \text{S} = \underset{\text{Noun}}{\underline{\text{Books}}}$$

Inflectional morphemes are Infixes and Suffixes and cant Be prefix

Derivational Morphemes:

Derivational Morphemes are one which when added to a word changes its class.

Teach + er = Teacher
Verb Noun

Derivational morphemes are of two types class changing which was the above one and class maintaining which when added to word changes the word but cant change the class.

Child + hood = childhood
common noun abstract noun but the class is same which is Noun

Q3 Design a finite state automata (FSA) for baat!

=> Q: finite set of states
q_0, q_1, q_2, q_3, q_4

Σ: set of input alphabets
: $\{a, b, !\}$

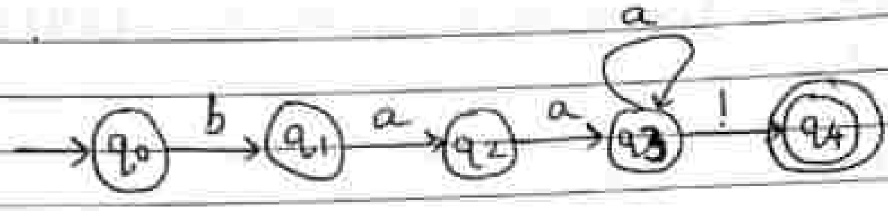
q_0: Start State.

F: Set of final States
: $F \subseteq Q$

$\delta(q,i)$: the transision function or transision matrix between states. Given a state $q \in Q$ and input symbol $i \in \Sigma$, $\delta(q,i)$ returns a new state $q' \in Q$. δ is thus a relation from $Q \times \Sigma$ to Q;

Transition table

States \ Input	a	b	!
q_0	ϕ	q_1	ϕ
q_1	q_2	ϕ	ϕ
q_2	q_3	ϕ	ϕ
q_3	q_3	ϕ	q_4
q_4	ϕ	ϕ	ϕ



Q4 Design a finite State Automata for divisibility by 5 tester for binary number.

Q => finite set of State
qS => Start State (*Might or Might not include You yourself)
q0 => rem 0 State
q1 => rem 1 State
q2 => rem 2 State
q3 => rem 3 State
q4 => rem 4 State

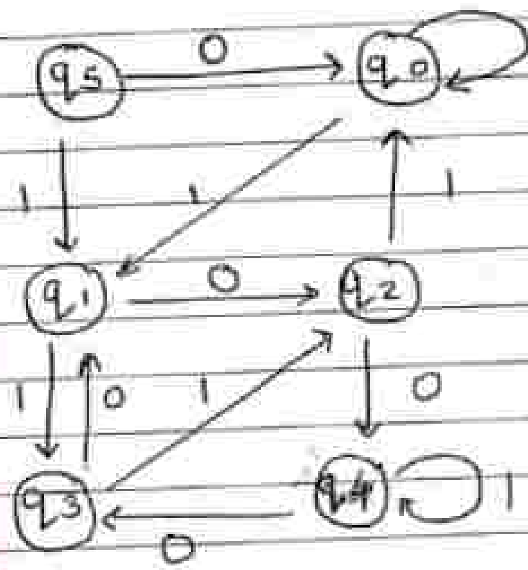
Σ => set of input alphabets
= {0, 1}

rem = reminder

The question can also be written as design a FSA to check wether given binary no is divisible by 5 or not

δ = Transition function
$\delta : Q \times \Sigma \to Q$
q0 = Start State = qS
F = final State
$q_0, F \le Q$
[or you can write like previous Numerical]

δ \ Σ	0	1
q_5	q_0	q_1
* q_0	q_0	q_1
q_1	q_2	q_3
q_2	q_4	q_0
q_3	q_1	q_2
q_4	q_3	q_4



Q5= Design a DFA of a string that should end with 100

=> $M = \{Q, \Sigma, \delta, q_0, F\}$

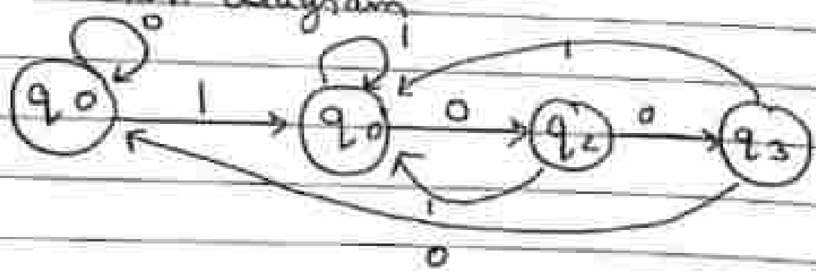
q_0 = initial state
q_1 = String ending with 1
q_2 = String ending with 10
q_3 = String ending with 100

$\Sigma = \{0, 1\}$
$q_0 = \{q_0\}$
F = final state = q_3

Transition Table

	0	1
q_0	q_0	q_1
q_1	q_2	q_1
q_2	q_3	q_1
q_3	q_0	q_1

Transition diagram



Date :

Q6. Differentiate between Inflectional & Derivational morphology.

	Inflectional Morphology	Derivational Morphology
1.	It is a morphological process that adapts existing words so that they function efficisively in sentences without changing POS of base morpheme/m	It is concerned with the way morphemes are connected to existing lexical forms as affixes.
2.	Regular: It is more Regular	It is very less regular
3.	Use: Can only be suffix or infix and not prefix	can be both prefix & suffix
4.	Change in Part of Speech: Never change the gramatical category or POS	It can change the gramatical category or POS
5.	Example: Cat + S = cats (Noun → Noun)	danger + ous = dangerous (Noun → Adjective)

Q7. Write a short note on language model

Q1. Write a short note on language model

=> The goal of a language model is to assign probability to a sentence.

With this it also decides which sentence is more accurate at the moment.

example: She is a tall girl is more accurate than She is a long girl

Statistical Language Modelling, or language Modelling is the development of probabilistic models that are able to predict the next word in the sequence given the words that ~~predict~~ precede it. It is a probability distribution over sequences of words.

Given such a sequence, say of length m, it assigns a probability $P(w_1, \ldots, w_m)$ to the whole sequence.

The goal of probabilistic language modelling is to calculate the probability of a sentence of sequence of words: $P(W) = P(w_1, w_2, w_3 \ldots w_n)$ and can be used to find the probability of the next word in the sequence: $P(w_5 | w_1, w_2, w_3, w_4)$

A model that computes either of these is called language model

Method of calculating Probability:

Conditional probability:

let A and B be two events with $P(B) \neq 0$, the conditional probability of A given B is:

$$P(A/B) = \frac{P(A,B)}{P(B)}$$

$$P(x_1, x_2, \ldots x_n) = P(x_1)\, P(x_2 | x_1) \ldots P(x_n | x_1 \ldots x_{n-1})$$

for example: P("its water is so transparent") = P(its) * P(water | its) * P(is | its water) * P(so | its water is) * P(transparent | its water is so)

we can estimate this by simply counting and deviding the results.

$$P(\text{transparent} / \text{its water is so}) = \frac{\text{count}(\text{its water is so transparent})}{\text{count}(\text{its water is so})}$$

Markov Property: It says that the probability of the next word can be estimated given only the previous k number of words.

for example, if $k = 1$:

P(transparent / its water is so) $\approx$ P(transparent / so)

or if $k = 2$:

P(transparent / its water is so) $\approx$ P(transparent / is so) general equation for the Markov Assumption, $K = i$:

$$P(w_i | w_1 w_2 \dots w_{i-1}) \approx P(w_i | w_{i-K} \dots w_{i-1})$$

Q8 Write a short note on N-Gram model.

=> Note: This answer is in continuation with the previous answer. You have to decide the length depending the marks allotted to question.

- The Simplest case of markov model is a unigram model, In this model we we simply estimate the probability of the whole sequence of words by the product of probabilities of individual words - unigrams. and if we generated sentences by randomly picking words, It would be Sixth, the, rupees, abduction

  It would be just a random sequence of words

$$P(w_1 w_2 \dots w_n) \approx \prod_i P(w_i)$$

- Slightly more intelligent is the bigram model where we condition on the single previous word.

$$P(w_i | w_1 w_2 \dots w_{i-1}) \approx P(w_i / w_{i-1})$$

We can extend this to trigrams, 4-grams, 5-grams.
But in general this is an insufficient model Of language.
o because language has long distance dependencies
example:
"The computer which I had just put into the machine room on the fifth floor crashed."

And If we say predict the next word after floor So It is very unlikely to predict crash but if we compare with or bring the main subject "computer" in the picture then we are more likely to guess crashed or predict crash as a next word.

Q9 corpus:

<s> I am a human </s>
<s> I am not a stone </s>
<s> I I live in Mumbai </s>

Check the probability of <s> I I am not </s> using bigram

=> P(I I am not)

$= P(I|\langle s\rangle)\, P(I|I)\, P(am|I)\, P(not|am)\, P(\langle /s\rangle|not)$

$= \dfrac{C(\langle s\rangle | I)}{C(\langle s\rangle)} \; \dfrac{C(I|I)}{C(I)} \; \dfrac{C(I|am)}{I(I)} \; \dfrac{C(am|not)}{C(am)}$

$\dfrac{C(not|\langle /s\rangle)}{C(not)}$

$= \dfrac{3}{3} \times \dfrac{1}{4} \times \dfrac{2}{4} \times \dfrac{1}{2} \times \dfrac{0}{1}$

$= 0$

Q10 consider following training data.

<S> I am Jack </S>
<S> Jack I am </S>
<S> Jack I like </S>
<S> Jack I do like </S>
<S> do I like Jack </S>

Assume that we use a biagram language model based on above data.
what is most probable next word predicted by the model?

1. <S> Jack .... 2. <S> Jack I do ....
3. <S> Jack I am Jack ....
4. <S> do I like ....

=> $P(I|<s>) = C(<s>|I)/C(<s>) = 1/5$

$P(Jack|<s>) = C(<s>|Jack)/C(<s>) = 3/5$

$P(do/<s>) = C(<s>/do)/C(<s>) = 1/5$

$P(am/I) = C(I/am)/C(I) = 2/5$

$P(like/I) = C(I/like)/C(I) = 2/5$

$P(do/I) = C(I/do)/C(I) = 1/5$

$P(</s>/Jack) = C(Jack/</s>)/C(Jack) = 2/5$

$P(</s>/like) = C(like/</s>)/C(like) = 2/3$

$P(</s>/am) = C(am/</s>)/C(am) = 1/2$

$P(I/Jack) = C(Jack/I)/C(Jack) = 3/5$

$P(like/do) = C(do/like)/C(do) = 1/2$

$P(I/do) = C(do/I)/C(do) = 1/2$

$P(Jack/like) = C(like/Jack)/C(like) = 1/3$

$P(Jack/am) = C(am/Jack)/C(am) = 1/2$

1. <s> Jack I   2. <s> Jack I do like & I

3. <s> Jack I am Jack I

4. <s> do I like </s>