## * UNIT-5 – Uncertain knowledge:

### → Discourse Processing :

Discourse Processing is a subfield of NLP that focuses on understanding the structure & meaning of text at beyond level of individual sentences.

→ The main aim is to extract meaningful information from the Text which can be useful for summerization, information extraction, question answering ....

→ It involves 4 sub tasks,

i) coherence & cohesion analysis

ii) Discourse structure analysis

iii) Rhetorical Parsing

iv) Discourse - level sentiment analysis.

i) coherence & cohesion analysis:

→ coherence refers to overall logical connection blw sentences in Text.

→ cohesion refers to grammatical connections blw sentences (such as: Pronouns, conjunctions ... etc)

eg:- John went to store. He bought some milk

He in 2nd sentence refers to John in 1st sentence.

ii) Discourse structure analysis:

→ It involves in analyzing the hierarchical structure in the Text, identifying main Topics, supporting details & relations ....

eg: 1st we'll discuss Problem, Then we'll find soln, By Discussion only we can find soln.

Structure: Introduction: First, we'll Discuss Prob
main Point: we'll find soln
supporting detail: By Discussion only
we can find soln

iii) Rhetorical Parsing:
↳ (To inform)

→ Rhetorical Parsing involves in analyzing structure of Text to understand how different Parts of a Text are related.

eg: "she was clever because she studys everyday".

because indicating its cause a effect of sentence.

iv) Discourse - Level sentiment Analysis:

Involves in analyzing the sentiment of a Text.

eg: The battery life of mobile is great (Positive)

but camera quality is bad (negative)

* Key components of Discourse Processing:

1) cohesion
2) Resolution Reference
3) Discourse cohension & structure

---

1) cohension :-

→ the grammatical connections blw the sentences

(or)

Features that link different parts of Text.

→ cohension includes;

a) Reference words: These are like Pronouns / articles which refers to entities mentioned in inital stage of Text.

eg: Lucky went to bookstore, He bough a book

he in 2nd sentence refers to Lucky in 1st sentence

b) conjunctions; words that connect sentences

like ÷ and, but, so.....

eg÷ I want to go outside, but it's raining

but is conjuntion

c) Ellipsis ÷ words which are understood & necessary to repeat

eg: "John reading a book, and Mary is too"

ellipse— mary also reading book

d) Lexical cohesion÷

creating a meaning using similar words

eg÷ "The restaurant had a tasty biryani.

The dish was like by many peo

Dish refers → Biryani

9) Reference Resolution:

It is the process of determining which words in the text refers to same entity.

a) Pronoun :- Identify To which entity the Pronoun is refering.

eg: After John finished work, he went outside.

he is a Pronoun refering → John.

---

b) Noun Phrase:-

Identify to which entity the Noun is refering

eg: The Book is on table, I read it yesterday.

Here, book & it refers same object.

---

c) Anaphora :- Using Pronoun to refer Previously mentioned entity

eg: I bought a new Laptop. It has high resolution

3) Discourse cohesion & structure:

Involves in understanding how different parts of Text related to form 1 sentence.

a) Discourse connectives: words which gives hints the relationships.

Eg: "The government should increasig education funds because Education is key to economic growth → {connective}

---

b) Discourse markers:

words which helps to organise words.

Eg: "firstly", we need to complete syllab markers "on the other" hand, we'll Prepar for exam.

---

c) segmentation:

Dividing Texts into segments.

Eg; The cat chased the mouse. The mouse ran away.

# * PART - 2 *

**overview :-**

1) Introduction to Language Modelling
   - overview
   - Applications

2) Types of Language models :-
   a) statistical Language Model
      → NGrams
   b) Neural Language Models
      → Feedforward Neural Networks
      → Recurrent Neural Network
      → Transformers (BERT, GPT)

3) Language Model Evaluation:
   → Perplexity
   → Accuracy

4) Parameter Estimation:
   → Max likelihood
   → Bayesian Parameter Estimation.
   → Large-scale Language Model

5) Language model Adaptation:

6) Language specific modelling

→ Word order
→ character - Based models.

7) multilingual & crosslingual modelling:

* Introduction :

Language modeling is a crucial Task in NLP involves in Predicting next word in a sequence.

Application:

Text Generation

speech Recognition

Machine Translation.

Eg: "The cat sat on the ___"

language model will Predict next word as 'mat'.

→ Types of Model:

* statistical lang model:

These model uses statistical methods to estimate Probability of a word sequence.

* N-Gram :

N gram can be defined as contiguous sequence of n items from a given Text.

→ The items can be letters, words, (or) pairs

→ N grams can be; times, let's say

unigram – (consider each word as independe

bigram – (considers Prob of previou word)

Tri gram – (considers Prob of previous two words

N Gram modeling:

Predicts the Probability of given N-gram with sequence of previous words.

Eg: the cat sat on the ___

→ It contains 5 words

"The", "cat", "sat", "on", "the"

→ To find the 6th word, find conditional probability of first 5 words

$P\left(\frac{w_6}{w_1, w_2, w_3, \ldots}\right)$

→ It can be calculated using chain rule

$P(A/B) = \frac{P(A \cap B)}{P(B)}$ ; Hence, we can find 6th word.

Example:

S1 = "I am Happy" Predict the likelihood of S1

S2 = "I am sad" with S2.

step-1; Let's Take Bigram (2 word/pair)

S1 → "I am", "am Happ"

S2 → "I am" ; "am sad"

step-2; count Frequency

"I am" → appeared 2 times

"am happy" → 1

Step 3: calc Probability

$$P\left(\frac{am}{g}\right) = \frac{\text{count (gam)}}{\text{count (g)}} = \frac{2}{2}$$

$$= 1$$

$$P\left(\frac{\text{happy}}{am}\right) = \frac{\text{count (happy)}}{\text{count (am)}}$$

$$= \frac{1}{2} = 0.5$$

$$P\left(\frac{sad}{am}\right) = \frac{\text{count (sad)}}{\text{count am)}} = \frac{1}{2}$$

$$= 0.5$$

Step - 4: Predict Prob

$$P(\text{"gam happy}) = P\left(\frac{am}{g}\right) \times P\left(\frac{happy}{am}\right)$$

$$= 1.0 \times 0.5$$

$$= 0.5$$

gm happy has 0.5 Probability.

* Neural Languge models :

a) feedforward :

uses a simple feedforward neural network where each word is represented by embeddings.
↓ (vector representation)

Architetuse :

Input : Fixed no.of Previous words given

Process : word enbeding given to Neural network

output : network Predicts Probability of next word.

eg: Input : "Sun rises in _____"

Process : each word represented as vector

Prediction : The neural network outputs the Probability of Next word.

eg : east, west, North, south.

If east has higher Probability

b) Recurrent Neural Network:

designed to handle the sequences, it maintains hidden layer / state, updated for every word sequence.

eg :- I love my ——

step - 1) "I" → update hidden state
2) "love" → update hidden state
3) "my" → update hidden state

it uses final state (hidden state) to predict next word it might be [dog, cat]

c)

c) Transformer based (BERT, GPT):

BERT: (Bidirectional encoder Representati from Transfor

it uses transformer to understand meaning of words from both

It mask some words wantedly (mask)

Eg: "The [mask] barked at cat".

It predicts masked word as "dog"

*GPT: (Generative Pretrained Transformer)

It generats text by predicting next word in a sequence.

→ It takes the input prompt & generates output.

Eg: "The weather today is"

It takes the prompt and predicts "The weather today is warm".

## * Language model evaluation *

1) Perplexity: used to measure how well a language model is predicting sequence of words.

low perplexit = better prediction of next word.

Eg: "The cat sat on the mat"

model A

$P(\text{The}) = 0.4$

$P(\text{cat}/\text{The}) = 0.2$

$P(\text{sat}/\text{cat}) = 0.2$

$P\left(\frac{\text{on}}{\text{cat sat}}\right) = 0.35$

$P\left(\frac{\text{mat}}{\text{cat saton}}\right) = 0.1$

model B

$P(\text{The}) = 0.5$

$P(\text{cat}/\text{The}) = 0.4$

$P(\text{sat}/\text{cat}) = 0.3$

$P\left(\frac{\text{on}}{\text{cat sat}}\right) = 0.1$

$P\left(\frac{\text{mat}}{\text{cat saton}}\right) = 0.2$

Perplexity =

$$PP(w) = \left[\frac{1}{P(w)}\right]^{\frac{1}{N}}$$

model A

$$PP(w) = \frac{1}{0.4 \times 0.2 \times 0.2 \times 0.3 \times 0.1}$$

$$= 43.84$$

model B

$$\frac{1}{0.5 \times 0.4 \times 0.3 \times 0.1 \times 0.2}$$

$$= 16.67$$

↓

low perplexity

mean better Prediction

---

2) Accuracy: measures % of correct Prediction made by a model

$$\text{Accuracy} = \frac{\text{correct Prediction}}{\text{Total Prediction}}$$

eg: "I love my Dog"

Predictions; after "I":

after I → love ✓

after I → my ✓

after I → cat ✗

After I → (love, my) ✓

Accuracy is

$$= \frac{2}{3} \times 100$$

$$= 66\ \%$$

## * Parameter Estimations *

1) max likelihood Estimation:

used to estimate Parameters of a model by likelihood function.

max likelihood = $\arg\max P\left(\frac{D}{\theta}\right) \to \theta$

$$\hat{\theta}_{MLE} = \arg\max P\left(\frac{D}{\theta}\right)$$

eg: Training Data

I love dogs ; I like dogs
I love cats ; I like cats
I love coding ; I like coding

To find Probability of next word after love

$$P(\text{love}) = \frac{\text{count(love)}}{\text{Total no. of words}}$$

$$= \frac{3}{6} = 0.5$$

50% of chance is love to be appear in given context.

## 2) Bayesian Parameter Estimation

In compares prior belief of parameter & update these bilief on data.

$$P\left(\frac{\theta}{D}\right) = \frac{P\left(\frac{D}{\theta}\right) \cdot P(\theta)}{P(D)}$$

Eg: 1) Flipping coin

$\alpha \rightarrow$ Head
$\beta \rightarrow$ Tail

} and prior belief is

$\alpha = 2 \; ; \; \beta = 2$

2) Now, After Fliping 10 times and observes 7 head ; 3 tails

3) update : $\alpha = \alpha_{prior} + \text{no. of H} = (2+7)$

$= 9$

$\beta = \beta_{prior} + \text{no. of T} = (2+3)$

$= 5$

distribution (9,5)

Prob of head $= \frac{\alpha}{\alpha+\beta}$

$= \frac{9}{9+5} = 0.6$

3) Large-scale Language model:

It consist of large amount of data where the Prediction occurs iteratively by assigning weights on previous work

Eg: step-1: Assing weights

step-2: model Predicts if correct ✓

, if wrong (calculate loss)

Re-assign weight - - - again Predict.

Eg; "The sun is shining"

step-1; Initialize weights (random weights)

step-2; model Predicts

"The sun is dull" ✗

calc loss; let's assume loss = 0.8

Re-Assign weights,

smoothing Technique; It is also used for Probability

Add-on smoothing; Add something to avoid 'zero' if scenario never happen

eg: 'The bird barked' → Not Possible but, Probablty > 0 (small non zero)

---

* Language model Adaptation *

Process of adjusting Pre-trai -ned language model to perform better on a specific Task.

→ models like GPT sometimes may not give Proper answers so, Adaptation makes them to Provide better information.

Benefits of Adaptation;

- Improved performance
- Efficiency

## Techniques of Adaptation:

1) Fine-tuning:

- Adjust the pretrained model Parameters by training on a smaller domains first, then increment further.

---

2) Prompt engineering:

Providing a Proper Prompt / instructions to guide model's Behavior.

Eg: "summerize about NLP" can give a full summery of NLP

---

3) Few-short Learning:

Training on a small amount of labeled datasets.

4) zero shot learning:

using Pretrained model without any additional Training

## * Language - specific modelling

It is a fundamental Task in NLP involves in Predicting next word / sequence of words in a Text by word order / character.

---

i) Word order:

Different languages have different word orders.

→ English have fixed word order
German have flexible word order.

Eg: The Dog, The cat (english)
Der Hund Der Katze (german)

---

ii) Character - Based:

Breaking the words into smaller units (characters / syllables)

Eg: "running" → "r" "u" "n" "n" "i" "n" "g"

## * Multi Lingual & Cross Lingual *

→ Multi Lingual:

Focuses on building models that generates text in multiple Languages.

→ used for machine Translation, Text summerization

→ eg: Translating Text from English to spanish.

---

→ cross Lingual:

This involves in training model in one language to Perform Task in another language.

→ The trained data will be limited & The languge where it perform will be high level

low trained data → high lang-uage