Natural Language Processing

Ch-1 Introduction to NLP

Q-1 Explain applications of NLP in brief

Ans NLP is a branch of AI that enables computers to understand, interpret and generate human language

- Applications of NLP

1) Machine Translation
→ Used in systems like Google Translate to automatically translate text or speech from one language to another while preserving meaning and grammar

2) Chatbots and Virtual Assistants
→ NLP powers assistants like Siri, Alexa and ChatGPT enabling them to understand user queries, respond conversationally and perform tasks based on voice or text input

3) Sentiment Analysis
→ Bussiness use NLP to analyze customer reviews or social media posts to determine the sentiment and improve services or products

4) Text Summarization
→ Automatically generating summaries of longer texts. Eg: Used in web browsers to summarize

news articles

5) Information Extraction
→ Extracting specific information from unstructured text data
→ Example: Extracting names, dates, phone numbers, or key facts from documents

Q-2 Explain the Components of NLP

→ For natural ~~proce~~ language communication to take place, following two things are necessary

- Language Understanding
- Language Generation

→ Natural language understanding, it means to understand the context, and Natural language generation relates to sensible response to the context.

1) NLU (Natural Language Understanding)
→ NLU helps the machine to understand and analyze human language
→ NLU used in business applications to understand the customer's problem in both spoken and written language
- NLU involves two tasks:-
  → Used to map the given input into useful representation
  → Used to analyze different aspects of the language

2) Natural Language Generation (NLG)
→ NLG acts as a translator that converts the computerized data into natural language representation
- It involves in
  → Text planning
  → Sentence planning
  → Text realization

Q-3 State various Python libraries of NLP OR Explain NLP API and Libraries

Ans NLP API
→ An NLP API is a ready-made interface provided by AI platforms that allows developers to easily integrate language processing features into their applications without building models from scratch
- Function
  → These APIs are pre-trained NLP models to perform tasks like text translation, summarization, sentiment analysis, entity recognition, speech-to-text and chatbot communication

Eg: 1) Google Cloud NLP API
2) IBM Watson NLP API
3) OpenAI API
4) Microsoft Azure text Analytics API
5) Google speech-to-text
6) OpenAI Whisper

Page No.
Date

○ NLP Libraries
are open-source programming toolkits that provide functions and pre-trained models to perform NLP tasks efficiently

→ These Libraries offer tools for text processing, analysis, and understanding, making them essential for developers working with textual data

Example
1) textblob
2) NLTK
3) Core NLP
4) Gensim
5) SpaCy
6) Scikit learn

Q-4 What is NLP? Why NLP is difficult?

Ans NLP is the field of CS that bridges the gap between human language and computer understanding. It allows machines to process and analyze large volumes of text and speech data, enabling tasks like
- Understanding intent and sentiment
- Extracting Information
- Generating Text
- Translating Language
- Answering Questions

○ Why NLP is difficult
1) Ambiguity
2) Contextual Dependence
3) Variations in language

4) Rule Based Systems Limitations
5) Data Requirements
6) Computational Complexity

Q.5 State Phases of NLP. Explain each step for following statement "GTU is the winner of Robocon 2023"

Ans 1) Lexical Analysis
This initial phase focuses on breaking down the input text into smaller units called tokens (words, punctuation) etc. It also identifies and categorizes these tokens, often involving tasks like stemming and lemmatization

tokens: GTU | is | the | winner | of | | 2023
winner: → "win" + "-ner"
win: B

(2) Syntatic Analysis: Also known as Parsing, this phase analyzes the grammatical structure of sentences. It determines how words relate to each other, identifies phrases, and checks for grammatical correctness
Eg:
- Subject: GTU
- Verb: is
- Predicate: the winner of

(3) Semantic Analysis: This phase delves into the meaning of text, focusing on the literal meaning of words, phrases and sentences. It aims to understand the relationships between words and how they contribute to the overall meaning

Eg: Understands that "GTU" is the organisation and "winner" indicates victory or achievement in "Robocon 2023"

4) Discourse Integration : This phase examines how sentences relate to each other within a larger context. It considers the impact of previous sentences on the understanding of the current sentence, involving resolving pronoun references

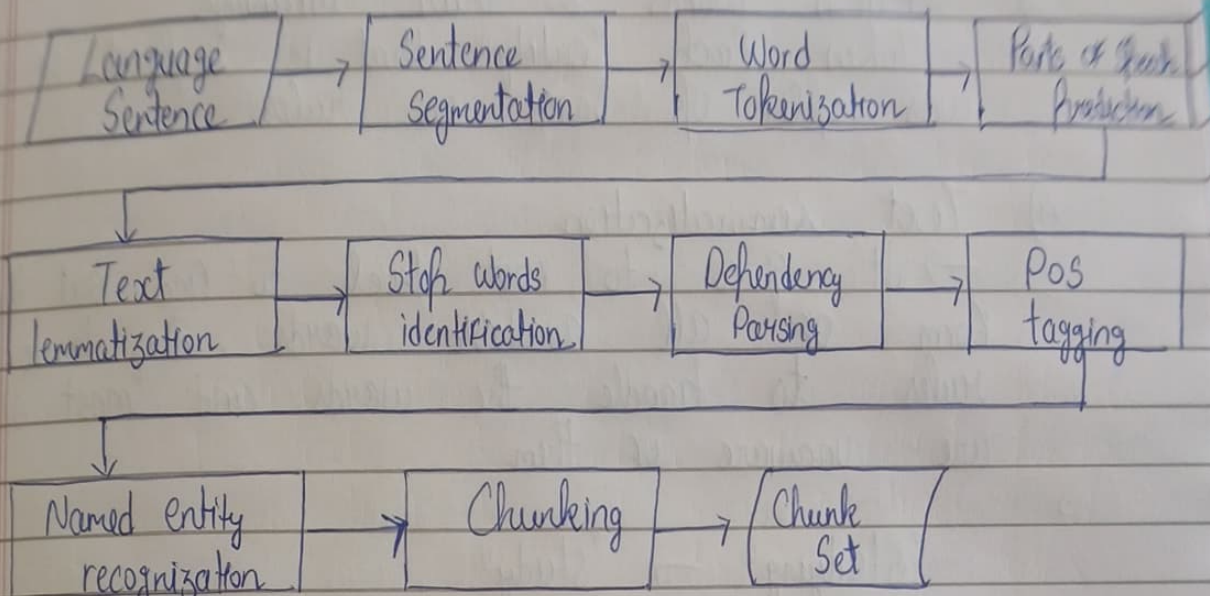
For single sentence
- No previous sentence → treated as independent info.
- "GTU" is recognized as a new entity introduced in discourse

5) Pragmatic Analysis
The final phase focuses on understanding the intended meaning and effect of the text, considering factors like context, speaker intent and real world knowledge. This phase interprets nuances like sarcasm, politeness or the overall tone

Eg:
Understands that "Robocon" is a robotics competition and statement expresses an achievement by GTU in that event

Q-6 Explain a method to build NLP pipeline with suitable example

Ans



1) Sentence Segmentation

→ Paragraph is broken / segmented into sentences

→ It reduces the complexity and simplifies the process even gets you the most accurate results.

2) Word Tokenization

→ Tokenization is process of breaking a phrase, sentence, paragraph or entire documents into smallest unit such as individual words or terms. And each of these small units is known as tokens

→ These tokens could be words, numbers, or punctuation marks.

["I", "love", "A", "I"]

3) Parts of Speech Prediction

→ In a part of the speech, we have to consider each token, whether tokens belong to nouns, pronouns, verbs, adjectives and so on

1) Corpus: Body of text, Singular
2) Lexicon: Words & their meanings
3) Token: Each "entity" that is a part of whatever was split up based on rules

4) Text Lemmatization
→ It goes to root level to find out the base form of all available words. They have ordinary rules to handle the words and most of us are unaware of them

"studying" → "study"
"running" → "run"

5) Identifying Stop words
→ Remove filler words like "is", "the", "a"

6) Dependency Parsing
→ Parsing is divided into three prime categories further. And each class is different from others they are part of speech tagging, dependency parsing and constituency parsing.

→ Dependency Phrasing Case: Analyses the grammatical structure of the sentence. Based on the dependences in words of sentences. Whereas in constituency parsing: the sentence breakdown into sub-phrases And these belong to a specific category like Noun Phrase (NP) and Verb Phrase (VP)

7) Pos tags
→ Pos stands for Parts Of Speech, which includes noun, verb, adverb and adjective. It indicates that how a word functions with its meaning as well as grammatically within the sentences
→ A word has one or more parts of speech based on the context in which it is used

8) Named Entity Recognition (NER)
→ It is process of detecting the named entity such as person name, organization name or location

9) Chunking
→ Chunking is used to collect the individual piece of information and grouping them into bigger pieces of sentences

Q-7 Give advantages and disadvantages of NLP

Ans Advantages:
1) Improved Communication
2) Enhanced Data Processing
3) Automation of tasks
4) Sentiment Analysis
5) Personalized experiences

Disadvantages
1) Bias
2) Complexity
3) Limited Contextual Understanding
4) Language Dependence
5) Maintenance and Updates

Date

Ch-2

Language Modelling and Pos Tagging

Q-1 Explain Unigram, Bi-gram and Tri-gram concepts using an example sentence:
"The purpose of our life is to happy"

Ans 1) Unigram Language Model
→ A unigram model can be treated as the combination of several one-state finite automata
It splits the probabilities of different terms in a context eg from

$$P(t_1\ t_2\ t_3) = P(t_1)\ P(t_2|t_1)\ P(t_3|t_1 t_2)$$

to

$$P_{uni}(t_1 t_2 t_3) = P(t_1)\ P(t_2)\ P(t_3)$$

→ In unigram language model, the probability of each word only depends on ~~is~~ that word's own probability in the document, so we only have one-state finite automata as units

Eg: "The", "Purpose", "of", "our", "life", "is", "to", "happy"

(2) Bigram Language Model
→ It approximates the probability of a word given all the previous words by using only the conditional probability of one preceding ~~details~~ word.

$$P(w_n | w_{n-1}) = \frac{P(w_{n-1}, w_n)}{(w_{n-1})}$$

→ A bigram is a sequence of two adjacent elements in a string is commonly used for examples

→ A bigram is a sequence of two adjacent elements from a string of tokens, which are typically letters, syllables or words. A bigram is an n-gram for n=2

Eg: "The purpose", "purpose of", "of our", "our life", "life is", "is to", "to happy".

3) Trigram

→ Trigram are a special case of n-gram, where n is 3 Trigram models, conditions on the previous two words preceding it. rather than only the previous word.

Probability of Trigram

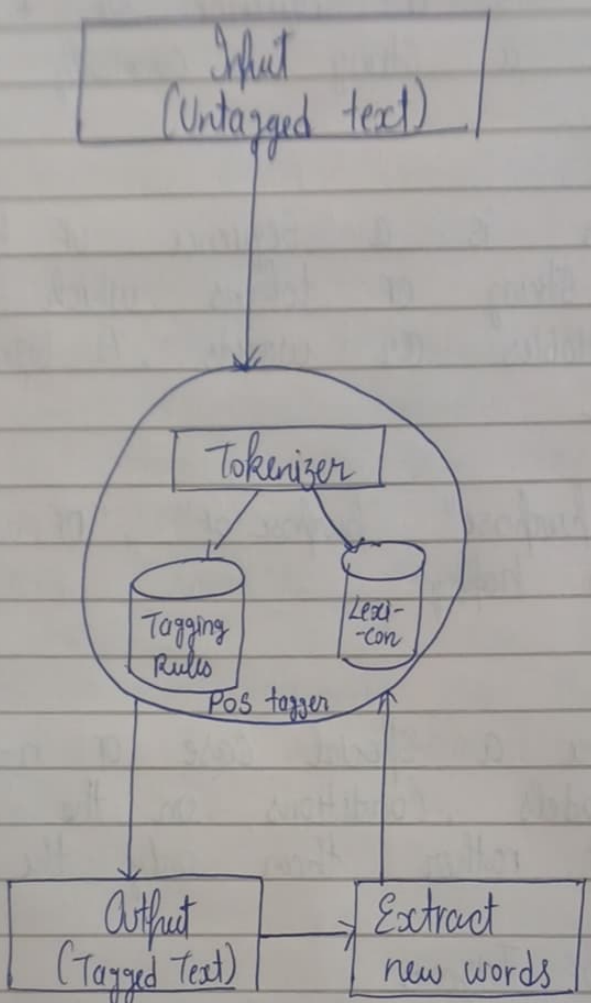
$$P^{\wedge}ML\{w_3|w_2|w_1) = \frac{C(w_1\ w_2\ w_3)}{C(w_1, w_2)}$$

Eg: [The purpose of], [purpose of our], [of our life], [our life is], [life is to], [is to happy].

Q-2 Describe Part-of-Speech Tagging with example

Ans POS is an NLP technique that involves assigning a grammatical category (like noun, verb, adjective etc) to each word in a text.

→ It's a fundamental step in understanding the structure and meaning of sentences, aiding various NLP tasks



PoS tagging Process

There are two types of taggers
1) Rule-Based Taggers
2) Stochastic Taggers

1) Rule Based Taggers
Input
↓
Tag or Dictionary or Lexicon
↓
Handwritten rules
↓
Output (word, tag)

→ Rule Based taggers generally involve a large database of hand-written disambiguation rule which specify, for example, that an ambiguous word should have a given is a noun rather than a verb if it follows a determiner

I Want to Perform a Play

determiner ~~Verb~~ Noun

o We can also understand rule-based POS tagging by its two-stage architecture

1) First Stage: In first stage, it uses a dictionary to assign each word a list of potential POS.
2) Second Stage: In second stage, it uses large lists of hand-written disambiguation rules to sort down the list to a single POS for each word

Properties of Rule Based Taggers
1) Are Knowledge-driven taggers
2) Rules are built manually
3) Information is coded in form of rules
4) There are limited no. of rules
5) Smoothing and language modelling is defined explicitly in rule-based taggers

2) Stochastic taggers
→ It generally resolve tagging ambiguities by using a trainning corpus to compute the probability of a given word having a given tag in a given context

Stochastic tagger applies the following approaches for POS tagging

1) Word Frequency approach
2) Tag Sequence Probabilities

1) Word Frequency approach

→ The tag encountered most frequently in the training corpus is the one assigned to the ambigous instance of the word.

→ The main issue with this approach is that it may yield inadmissible sequence of tags.

2) Tag Sequence Probabilities

→ It is another approach of stochastic tagging, where the tagger calculates the probability of a given sequence of tags occuring

→ n-gram approach called.

→ It is called so because the best tag for a given word is determined by probability at which it occurs with the n previous tags

Properties

1) POS tagging based on probability of tag occuring
2) It requires training corpus
3) There would be no probability for the words that do not exist in the corpus
4) It uses different testing corpus.
5) It is simplest POS tagging because it chooses most frequent tags associated with a word in training corpus

Q-3 List and Explain Smoothing techniques used in NLP

Ans
1) Additive / Laplace Smoothing
2) Good - Turing estimate
3) Kneser Key Smoothing
4) Katz Smoothing (back - off)
5) Absolute Discounting

1) Additive / Laplace Smoothing
→ Not used in modern n-gram models.
→ Add 1 to every word so no probability becomes zero.

Before $P(w_i) = \frac{C_i}{N}$

After Laplace Smoothing

$$P_{Laplace}(w_i) = \frac{C_i + 1}{N + V}$$

2) Good - Turing estimate
→ Good turing Smoothing techniques uses the frequencies of the count of occurence of N-grams for calculating the maximum likelihood estimate

$$P_{unknown}\left(\frac{w_i}{w_{i-1}}\right) = \frac{N_1}{N}$$

$$P\left(\frac{w_i}{w_{i-1}}\right) = \frac{N_1}{N}$$

where $C^* = (C+1) \times \frac{N_{t+1}}{N_c}$

c = original count of an event
c* = adjusted count
N(c) = no. of events that occur exactly c times
N(c+1) = " " " " c+1 times

3) Kneser-ney Smoothing
→ In Good turing it is observed that the count of n-grams is discounted by a constant/ absolute value such as 0.75
Absolute-discounting is applied

$$P_{Kneser\text{-}Ney}\left(\frac{w_i}{w_{t-1}}\right) = \frac{\max(c, w_{t-1}, w_i - d, 0)}{c(w_{t-1})} + \lambda(w_{i-1}) \times P_{Continuation}(w_i)$$

where λ = normalizing constant

$$\lambda(w_{i-1}) = \frac{d \times |c(w_{t-1}, w_i)|}{c(w_{i-1})}$$

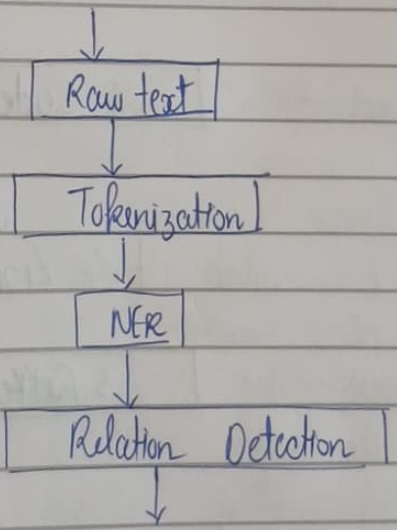
4) Katz Smoothing
→ In Katz smoothing technique, good turing technique is combined with interpolation. It outperforms good-turing by redistributing different probabilities to different unseen units

5) Absolute Discounting
Absolute Discounting involves subtracting a fixed discount D, from each nonzero count, an redistributing this probability mass to N-grams with zero counts

$$P_{new}(w_i | w_{i-n+1} \dots w_{i-1}) = \frac{\max\{c(w_{i-n+1} \dots w_i) - D, 0\}}{X}$$

Q-4 Write a short note on : Named Entity Recognition

Ans → NER is a core component of NLP that identifies and classifies named entities within text into predefined categories like person, organization, location, date etc



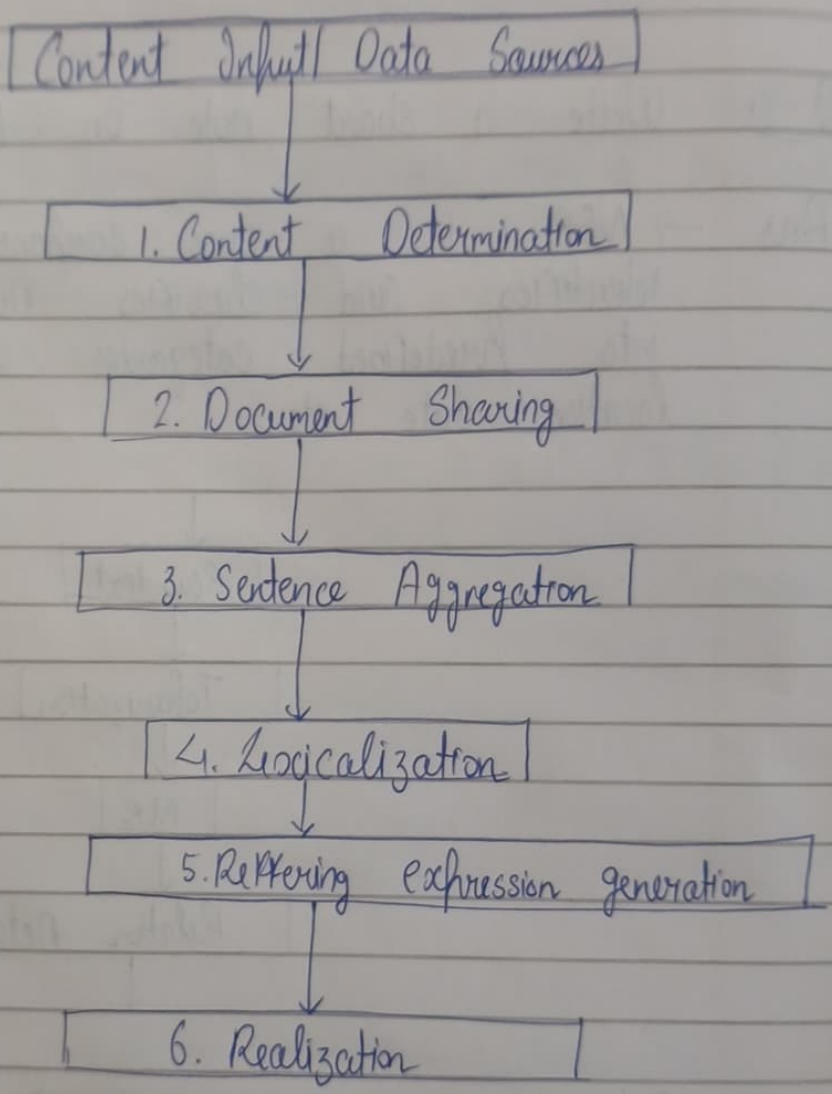
* Entities
→ Person
→ Location
→ Organization
→ Date

* Working
→ Text Preprocessing
→ Entity Identification
→ Entity classification/categorization
→ Contextual Analysis

Eg: 'Sujal won first prize in hackathon in Surat'
Sujal: PERSON, hackathon: EVENT, SURAT: LOCATION

Q.5 Draw the architecture of NLG system and explain its steps in detail

Ans



1) Content Determination : Deciding what information to mention in the text
2) Document Sharing : Overall organisation of the info. to convey. For eg, deciding to describe the areas with high ~~level~~ pollen levels first, instead of the areas with low pollen levels
3) Aggregation : Merging of similar sentences to improve readability and naturalness. For instance, merging the two following sentences

4) Lexical Choice : Putting words to the concepts. For eg. deciding whether medium or moderate should be used when describing a rollen level of 4.

5) Referring Expression Generation : Creating referring expressing that identify objects and regions. This tasks also includes making decisions about pronouns and other types of anaphora

6) Realization : Creating the actual text, which should be correct according to the rules of syntax.

Q-6 Explain Deleted Interpolation Smoothing

Ans → Advanced version of Interpolation Smoothing

→ In n-gram models, some word combinations may not appear in training data
→ If we rely on trigram or bigram model, such unseen sequences get zero probability
→ To fix this, deleted interpolation uses all n-gram models together with suitable weights that are learned from data

Formula: $P(w_i | w_{i-2}, w_{i-1}) = \lambda_1 P_{trigram} + \lambda_2 P_{bigram} + \lambda_3 P_{unigram}$

$\lambda = \lambda_1 + \lambda_2 + \lambda_3 = 1$

These weights are estimated automatically by testing the model on "deleted" data

Page No.
Date

Advantages
1) Avoids zero probabilities
2) Uses all models (unigram, bigram, trigram) together
3) Weights are automatically optimized - no manual tuning
4) Performs better than normal interpolation
5) Avoids overfitting using held-out data

Q-7 Difference between backoff & interpolation

Ans

Backoff	Interpolation
1) Uses lower-order models only when higher-order ones fail	1) Combines multiple models together
2) Works in a step-by-step fallback manner	2) Works in weighted combination manner
3) Uses one model at time	3) Uses all models simultaneously
4) Simpler & faster	4) More accurate but computationally heavier
5) Probability from lower model only is higher is zero	5) Probability = weighted sum of all models
6) Eg: Katz Backoff	6) Jelinek-Mercer Interpolation

Q-8 What is Perplexity ?

Ans Perplexity is a measure used to evaluate how well a language model predicts a sentence
- Low Perplexity : good model (less confused)
- High Perplexity : bad model (more confused)

$$\text{Perplexity} = P(w_1, w_2, \ldots w_n)^{-1/N}$$

N = no of words.

Q-9 What is Morphology ? Explain the approaches to morphology

Ans Morphology is a branch of linguistics (and NLP) that studies the internal structure of words and how words are formed from smaller units called morphemes

Approaches :-
1) Morpheme - Based Approach
2) Word - Based Approach
3) Paradigm - based Approach

(1) un + kind → unkind
teach + er → teacher

Words are built by joining morphemes in a sequence

2) Word Based Approach
Words are transformed into other words through processes, not by breaking into morphemes

Example:
run → ran
mouse → mice
go → went

3) Paradigm - Based Approach
→ Words are grouped into paradigms - sets of word forms belonging to same root

Eg: Paradigm of write
write, writes, wrote, written, writting

Q-10 Explain POS and its importance

Ans POS check Q-2

POS Importance:-
1) Helps in Sentence Understanding
2) Improves Parsing
3) Essential for Machine Translation
4) Important for Information Extraction
5) Helps in Lemmatization
6) Useful in Text-to-Speech

Q-11 Give the formula to find perplexity in Unigram and Bigram Language Model

Ans o Unigram

$$PP_{unigram} = \left(\frac{1}{P(w_1) \times P(w_2) \times \ldots \times P(w_N)} \right)^{\frac{1}{N}}$$

Page No.
Date

- Perplexity in Bigram

$$PP_{bigram} = \left(\frac{1}{P(w_1|<s>) \times P(w_2|w_1) \times \ldots \times P(w_n|w_{n-1})} \right)^{\frac{1}{N}}$$

<s> is start symbol of sentence.

Difference Between Smoothing Techniques in NLP

No.	Technique	Working Principle	Key Idea / Formula	Pros	Cons
1	**Additive / Laplace Smoothing**	Adds 1 (or α) to every count to avoid zero probabilities	`P = (count + 1) / (N + V)`	Simple & easy to implement	Over-smooths (changes probabilities too much)
2	**Good-Turing Smoothing**	Adjusts counts based on how many events occur once, twice, etc.	Replaces count `c` with `c* = (c+1) * N(c+1)/N(c)`	Handles unseen events well	Difficult for large datasets
3	**Kneser-Ney Smoothing**	Improves backoff by considering *how likely a word appears in new contexts*	Uses discounted counts + continuation probability	Most accurate for real-world language models	Complex implementation
4	**Katz Smoothing (Back-off)**	Uses higher-order probabilities if available, else backs off to lower ones	Applies discount `d` and backs off recursively	Efficient and widely used	Slightly less smooth than interpolation
5	**Absolute Discounting**	Subtracts a fixed discount `δ` from nonzero counts and redistributes probability mass	`P = (count - δ)/N + δ * P_lower`	Simple and effective	Needs good tuning of δ value

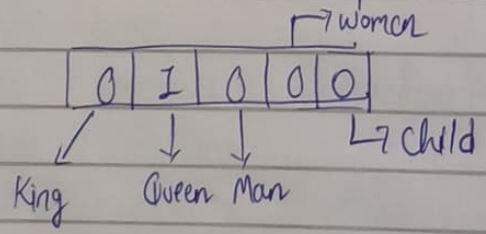
Ch-3 Word and Word Forms

Q-1 Explain Word embedding in brief

Ans Word Embedding is a technique in NLP that represents words as dense numerical vectors so that a computer can understand their meaning

Two types:-
1) Frequency based embedding
2) Prediction based embedding

1) Frequency based embedding
→ One-hot encoding
→ Creating numerical representations



'Queen' encoding all other words Zero

2) Prediction based embeddings
→ Word2Vec
→
→ Fixed vector

	the	dog	sat	in	hat
the dog sat	1	1	1	0	0
the dog sat in	1	1	1	1	0

Q-2 Describe Embedding representations for words Lexical Semantics with example

Ans Lexical Semantics is a subfield of linguistics that studies the meaning of words and how words are related to each other in terms of meaning

Terms used:

1) Lexeme: It is basic unit of meaning
→ It can be considered as a word in its most basic form
Eg:
- "is", "was", "will be" → same lexeme ("be")
- "come", "came", "coming" → same lexeme ("come")

2) Lexicon
A lexicon is collection of lexemes

3) Lemma
→ A Lemma is a collection of lexemes ×
→ You can think of it like a dictionary of all basic meaningful units in a language ×

→ It is dictionary form of lexeme
→ It is the standard/base form used to represent the entire lexeme

Eg: Lexeme forms:
- sing, sang, sung → Lemma = "sing"
- better, best → Lemma = "good"

4) Lemmatization

Lemmatization is process of converting a word to its lemma

Examples

- "running" → run
- "was" → be
- "mice" → mouse
- "children" → child

Eg:

"The kids were running quickly"

- "kids" → lemma : kid, lexeme : {kid, kids}
- "were" → lemma: be, lexeme : {is, am, are, was, were}
- "running" → lemma: run, lexeme : {run, ran, running}

Q-3 Explain Word Sense Disambiguation with Lesk Algorithm

Ans WSD identifies the correct meaning of an ambiguous word in a given context, such as the difference between "bank" as a financial institution and "bank" as the side of river

→ WSD is the task of selecting the correct sense for a word

→ WSD is essential part of many important NLP applications like question answering, information retrieval and text classification where the use of wrong senses of words can create disasters

Many english words are ambiguous:-

- bank
  - → river bank (land)
  - → money bank (Financial Insti.)

- bat
  - → a flying animal
  - → a cricket bat

WSD helps machines understand the correct sense based on the context

- Lesk Algorithm

→ It is simplest method for WSD

- Steps:-
  1) Identify the target word
  2) For each possible sense of that word:-
     - Take the dictionary definition
  3) Compare the definition with the context words around the target word
  4) Count the overlapping words
  5) The sense with the highest overlap is chosen as the meaning.

Example

"He sat on the bank of the river"

Ambiguous word = bank

Possible senses

1) bank (financial)
2) bank (river side)

Context words:-

- sat, on, the, river

- Compute Overlaps

Sense 1: Financial bank

Words: Institution, money, savings

Context words: sat, on, the, river

Overlap: 0

Sense 2: River bank
Words: sloping, ground, beside, river, lake
Context words: sat, on, the, river
Overlap = 1 (word "river" matches)

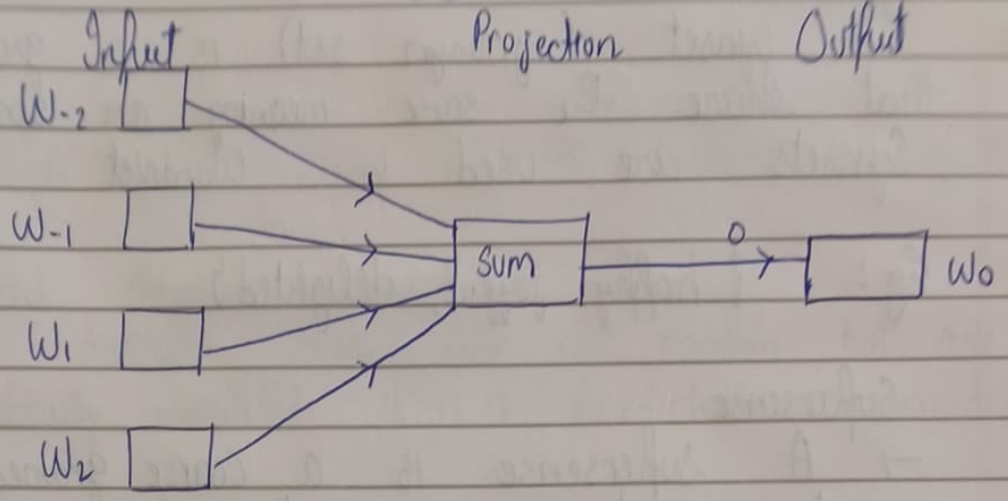
Lesk chooses the sense with maximum overlap → River Bank

Q-4 Explain CBOW with an example

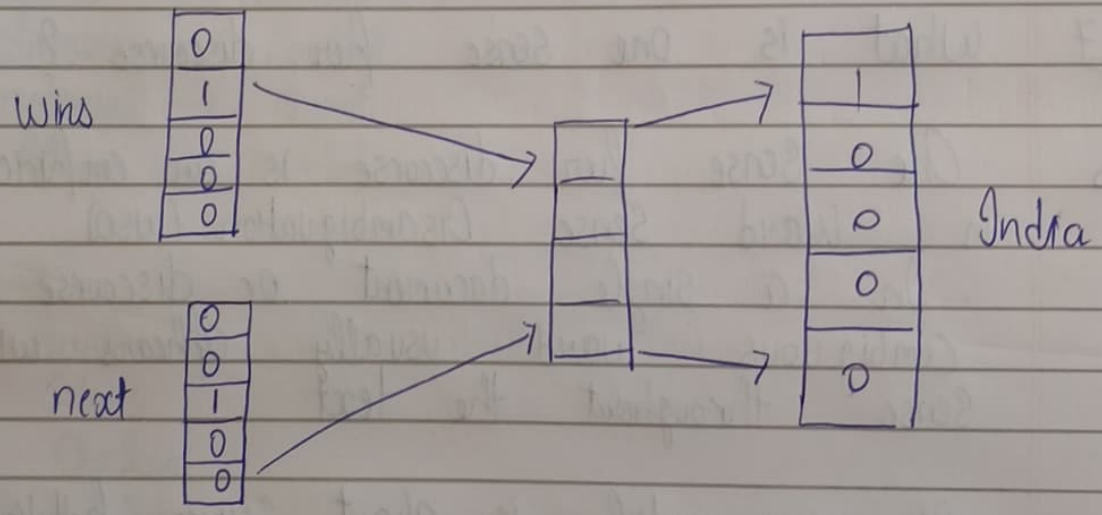
Ans It belongs to family of neural networks architectures called Word2Vec, which aims to represent words in a continuous vector space
→ In CBOW, the model predicts the current word based on the context of surrounding words
→ the architecture consists of an input layer, a hidden layer, and an output layer

1) Input layer: It represents the context words encoded as one hot-vectors
2) Hidden Layer: This layer processes the input and performs non-linear transformations to capture the semantic relationships between words
3) Output Layer: It produces a probability distribution over the vocabulary, with each word assigned a probability of being the target word given its context

CBOW



India wins next world cup



Q-5 Describe Lesk Algorithm in Detail

Ans Q-3

Q-6 Define Following terms : gloss, synset, Supersense

Ans 1) gloss: A gloss is the dictionary defination of a word or word sense
It explains the meaning of word in simple textual form

2) Synset

→ A synset (synonym set) is a group of words that share the same meaning or sense
Synsets are used in WordNet

Eg: {happy, joyful, delighted}

3) Supersense

→ A supersense is a coarse grained (high-level) semantic category used to classify words or synsets these are broader than individual senses

Q-7 What is one sense per discourse ?

Ans One sense per discourse is an empirical rule used in Word Sense Disambiguation (WSD)
In a single document or discourse, an ambiguous word usually appears with the same sense throughout the text

If news article is about "river pollution", the word "bank" will almost always mean:
o river bank, not a financial bank

o Importance
→ Helps simplify WSD algorithms
→ Reduces ambiguity in long texts
→ Improves accuracy of NLP tasks (eg. tagging parsing, info. extraction)

Q-8 Explain Structured Polysemy

Ans Polysemy means:
A single word has multiple related meanings (senses)

Structured Polysemy means:
These multiple senses are not random, but are systematically related in a predictable, organized pattern

That is, the senses arise from consistent semantic relationships, such as
- Physical → abstract
- Container → Content
- material → object
- Place → institution

- Examples
  1) Bank
  2) Glass
  3) Chicken
  4) School

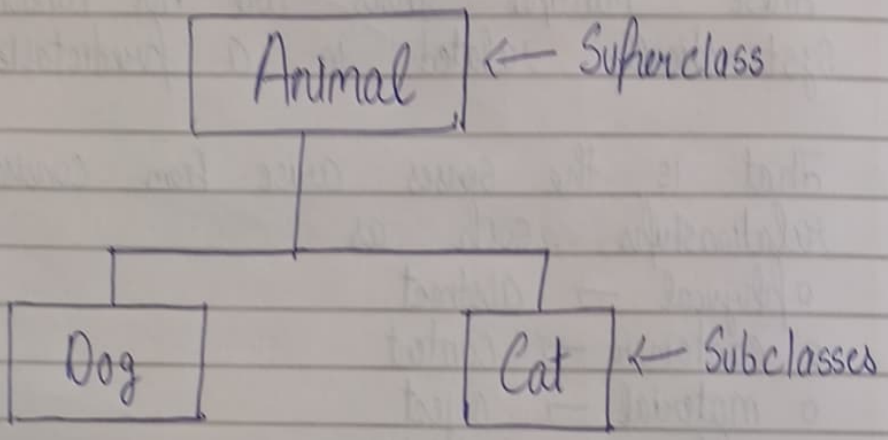
Q-9 Explain IS-A hierarchy

Ans IS-A hierarchy represents a relationship between classes (or concepts) in which a child (subclass) inherits properties and behaviour from a parent (superclass)

ISA means "is a type of"
It shows that one concept is a specialized version of another concept

Example:
- A Dog IS-A Animal
- A CAR IS-A Vehicle

This means Dog inherits the general properties of Animal and Car inherits from Vehicle



Explanation:
Both Dog and Cat are types of Animal.
→ They inherit properties like "can move", "can eat" etc from Animal

Q-10 Write a short note on
a) CBOW
b) Skip-Gram

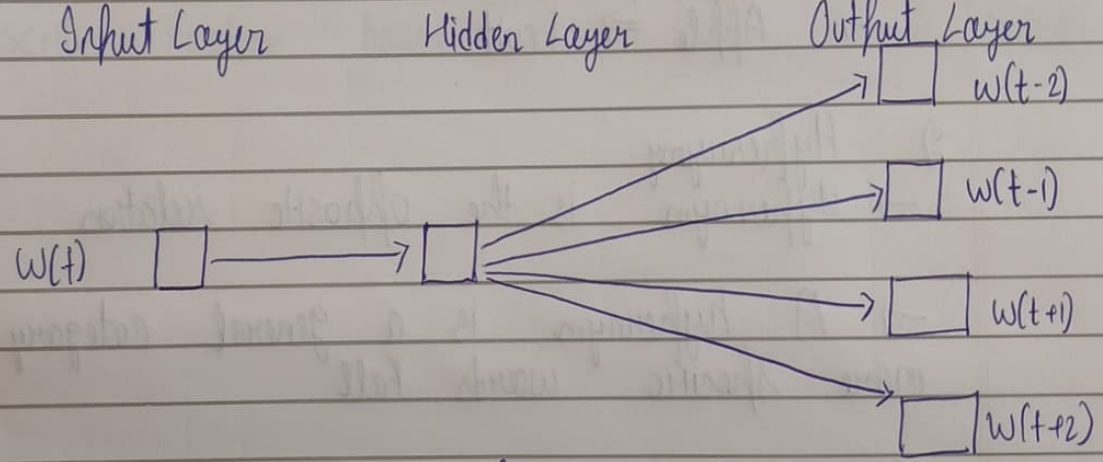
Ans a) CBOW
See Q-4

b) Skip-Gram
→ Unlike CBOW, ~~Skipgram~~ skipgram predicts context words given a target word
→ It's designed to learn the representation of a word by predicting the surrounding words in its context

Eg:
"The dog barked loudly at night"
- Center word = "dog"
- Context window = 2

- Skip-Gram Pairs
→ ("dog", "The")
→ ("dog", "barked")
→ ("dog", "loudly")

SKIP-GRAM Architecture



Input layer: single word (target word) encoded as one-hot vector
Hidden layer: transforms input word into distributed representation in hidden layer
Output layer: It predicts context words (surrounding words) based on representation learned in hidden layer

(Q-11) Q// Give Role of Skip-Grams

Ans (1) Capture Long Distance Dependencies
2) Reduce Data Sparsity
3) Improve Word Embeddings (Word2vec Skip-Gram model)
4) Useful For Free Word Order Languages

Page No.
Date

Q-12 Hyponymy & Hypernymy

Ans 1) Hyponymy is a semantic relation where a word represents a more specific concept within a broader category

A hyponym is a subtype of another word

Eg:
→ Dog is hyponym of Animal
→ Rose ⇒ Flower
→ Apple ⇒ Fruit

2) Hypernymy
→ Hypernym is the opposite relation

→ A hypernym is a general category under which more specific words fall

A hypernym is supertype or general class

Eg:
→ Animal is hypernym of Dog
→ Flower ⇒ Rose
→ Fruit ⇒ Apple

Q-13 Define collocation Feature of word

Ans Collocation refers to the tendency of certain words to occur together frequently more often than would be expected by chance

It represents a natural association or habitual co-occurence of words in a language

Examples of Collocations
- strong tea
- make a decision
- heavy rain
- fast food
- commit a crime

These word pairs are statistically significant and sound natural to native speakers

Q-14 Describe Feature based WSD with suitable example

Ans

Q-15 Explain Word-in-Context task with example

Ans The Word-in-Context (WiC) task is an NLP task used to check whether a word has the same sense or a different sense in two difference sentence contexts

It is commonly used to evaluate Word Sense Disambiguation and contextual word embeddings

WiC is a binary classification task
Given a target word and two sentences, ~~meaning~~ the system must decide:
- Same → if the word has the same meaning in both sentences
- Different → if the word has different meanings

Example:
Target word: "charge"
Sentence 1:
"He was arrested on charge of theft"
- charge = accusation (legal meaning)

Sentence 2:
"Please charge your phone before the trip"
- charge: give electrical power (electronics meaning)

Explain structured polysemy.

Here is a **clear, simple, exam-friendly explanation**:

✅ Structured Polysemy

Polysemy means:

> A single word has **multiple related meanings** (senses).

Structured polysemy means:

> These multiple senses are **not random**, but are **systematically related** in a predictable, organized pattern.

That is, the senses arise from **consistent semantic relationships**, such as:

- physical → abstract
- container → content
- material → object
- place → institution

So the polysemy is **structured** because the senses follow a **regular semantic pattern**.

✅ Examples of Structured Polysemy

1. "Bank"

- **Place:** "I am going to the bank"

✅ Examples of Structured Polysemy

1. "Bank"

- **Place:** "I am going to the bank."
- **Institution:** "The bank approved my loan."

Here, *bank* (place) and *bank* (institution) are **systematically linked** — the institution operates at the place.

---

2. "Glass"

- **Material:** "Glass is transparent."
- **Object made from material:** "I drank water from a glass."

Material → Object is a **regular polysemy pattern**.

---

3. "School"

- **Building:** "The school is closed today."
- **Organization / institution:** "The school announced a holiday."

Place → Organization is a **structured pattern**.

---

4. "Chicken"

- **Animal:** "A chicken is running in the yard."
- **Food:** "We had chicken for dinner."

Animal → Food is a **common structured semantic shift**.

✅ Role and Importance of Skip-grams in NLP

✅ 1. Captures long-distance dependencies

Normal bigrams only capture adjacent words.
Skip-grams capture meaningful relations across gaps.

Example:
"I love Indian **food."**
Skip-gram → (love, food)

✅ 2. Improves semantic learning in Word2Vec

Skip-gram is the **core model** in Word2Vec.

It predicts:

> **Given a target word, predict surrounding context words.**

Helps create high-quality word embeddings.

✅ 3. Helps in sparse data problems

In small corpora, exact adjacent n-grams may not appear often.
Skip-grams increase context combinations → improves training.

✅ 4. Better for free-word-order languages

Languages like Hindi, Gujarati, Marathi have flexible word order.
Skip-grams capture relationships even when words are not adjacent.

Example (Hindi):
"मैं कल मुंबई गया"
Skip-grams catch links between "मैं → गया" even wit' ↓ ords between.

✅ 5. Useful in text classification and sentiment analysis

Skip-grams capture important relationships like:

- (not, good)
- (very, happy)

Even if they are not consecutive.

✅ Short 2–3 mark answer

Skip-grams allow words with skips in between to form n-grams. They capture long-distance relationships between words, improve context learning in models like Word2Vec, help reduce sparsity, and are especially useful for languages with free word order.

If you want, I can also give:
✅ Examples for 2-skip, 3-skip
✅ Difference between n-gram and skip-gram
✅ Diagram for exam notes

Q-16 Difference between Knowledge Based and Supervised approaches for WSD

Ans

	Aspect	Knowledge-based	Supervised
1)	Data Requirement	Uses dictionaries, Wordnet	Used labelled data
2)	Training needed	No	Yes
3)	Accuracy	Moderate	High
4)	Resource Dependence	Lexical Resources	Annotated Corpora
5)	Algorithms	Lesk, semantic similarity	ML models, neural networks
6)	Explainability	High	Medium
7)	Best Used when	No labelled data	Large annotated corpus available

Ch-4 Text Summerization

Q-1 Explain Sentiment mining in brief

Ans → Sentiment mining is the process of extracting opinions, emotions, and attitudes from text (or speech) to determine whether the expressed sentiment is positive, negative or neutral
→ It's widely used in social media monitoring, product reviews, customer feedback, political opinion mining etc

o Steps in Sentiment Mining
1) Data Collection
→ Collect textual data (eg tweets, reviews, news)
→ Eg: "The movie was amazing, I loved it".

2) Text Preprocessing
→ Clean the text: remove punctuation, stopwords, URL's emojis
→ Tokenization: split text into words
→ Lemmatization / Stemming: reduce words to base form
→ Eg: "movie amazing loved".

3) Feature Extraction
→ Bag-of-words (BoW) or TF-IDF → numerical features from text
→ Word embeddings (Word2Vec, GloVe, FastText)
→ Contextual embeddings with transformers (BERT, ROBERTa)

Page No.
Date

q) Sentiment Classification
→ Lexicon-based approach : uses pre-defined dictionaries
→ ML models
→ Deep Learning
→ Transformers : BERT, GPT

Q-2 Write a detailed note on Semisupervised Relation Extraction via Boot strapping

Ans Relation Extraction (RE) is the task of identifying semantic relations (like person-born-in-place), company-founded-by) between entity pairs in text

When large labeled datasets are not available, semi-surpervised boot strapping is used as an effective method

Bootstrapping helps the system learn from a few seed examples and expand its knowledge automatically

Bootstrapping is a semi-supervised learning technique where:
→ The system starts with a small set of seed relations instances or patterns
→ It uses unlabeled data to discover new patterns and new relation instances
→ Newly discovered instances are again used to create better patterns
→ The process repeats iteratively, gradually improving accuracy

Page No.
Date

Bootstrapping Process
Step: 1 Provide Seed Instances (Input)
Step: 2 Pattern Extraction From Corpus
Step: 3 Use Patterns to Find New Instances
Step: 4 Score and Filter Candidates
Step: 5 Iteration

Q-3 Explain Macroaveraging and Microaveraging in brief

Ans Macro averaging
Macroaveraging gives equal weight to each class, regardless of how many examples are in each class

$$\text{Macro-Precision} = \frac{P_A + P_B + P_C}{3}$$

Microaveraging
Microaveraging gives equal weight to each individual sample, not each class

$$\text{Micro-Precision} = \frac{\Sigma TP}{\Sigma(TP+FP)}$$

Q-4 Write a detailed note on RE via Supervised Learning

Ans Supervised RE
Supervised learning is one of the most widely used techniques for Relation Extraction (RE). It uses a labeled Corpus where relation between entity pairs are manually annotated

The approach consists of two major subtasks
1) Relation detection between two entities
2) Classification of detected relation

→ Positive and negative samples are created
(annotated) (Non-annotated)

→ In second phase, training of classifier is done for labelling relations between candidate entity pairs.

→ Techniques like decision trees, naive Bayes, MaxEnt etc are used for labelling task

→ Set of classifiers are trained, one for each label as positive class and other labels are classified as negative class.

```
function FINDRELATIONS(word) return relations
    relations ← nil
    entities ← FINDENTITIES(words)

    For each pair(e1, e2) in entities:
        IF RELATED?(e1,e2):
            relation ← CLASSIFYRELATION(e1,e2)
            relation ← relations + relation

    return relations
```

Q-5 How is Text Summarization and Text Classification works in NLP ? Explain in detail

Ans Text Summarization is the process of automatically generating a shorter version of a document while preserving its key information and meaning

There are two main types:
1) Extractive Summarization : Picks the most important sentences / phrases directly from text
Example: Highlighting key lines from an article
Algorithms: TF-IDF ranking, TextRank (graph-based), LSA (Latent Semantic Analysis)

2) Abstractive Summarization : Generates new sentences that capture the meaning of the original text
→ Similar to how humans summarize
→ Uses Deep Learning models: Seq2Seq, LSTMs, Transformers

Steps:
1) Data Collection
2) Preprocessing
3) Feature Extraction
4) Apply Summarization Algorithm
5) Generate Final Summary

Text Classification is process of assigning predefined categories / labels to text documents based on their content
- Examples:
  1) Spam Filtering → Spam / Not Spam
  2) Sentiment analysis → (Positive / Negative / Neutral)

3) Topic labeling
4) Language Detection

Steps
1) Data Collection
2) Preprocessing
3) Feature Extraction
4) Model Training
5) Evaluation

Q-6 Explain Basic Algorithm of Text Classification

Ans
1) Data Collection
2) Preprocessing
3) Feature Extraction
4) Model Training
5) Evaluation

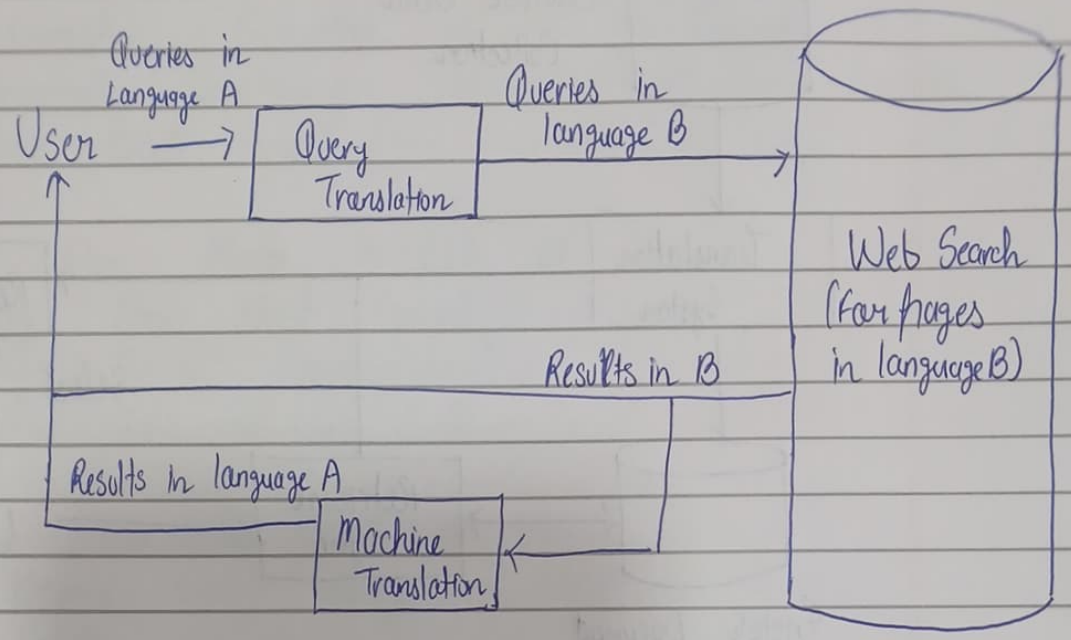
See image for more

Q-7 Describe Cross-Lingual Information Retrieval in NLP [W-23, S-24, S-25, W-22]

Ans Cross Lingual Information Retrieval (CLIR) is retrieving information in one language based on the query written in another language

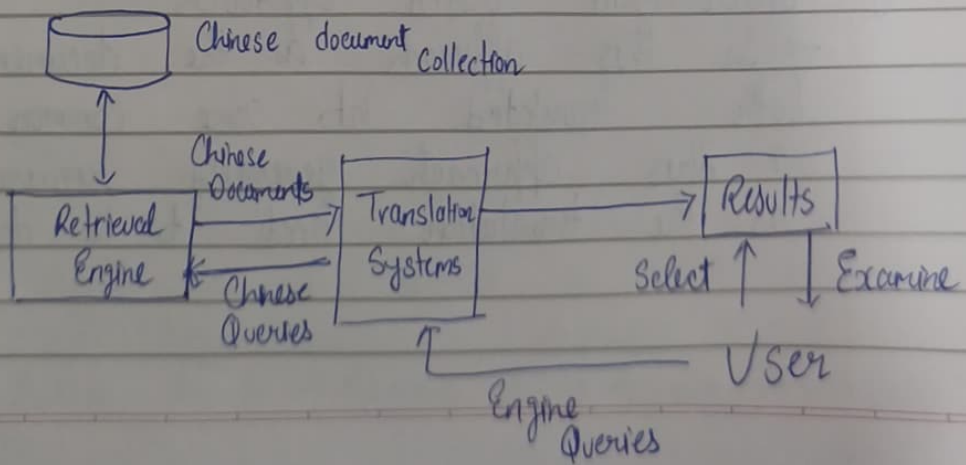
→ The users can search document databases in multiple languages and retrieve information in a form that is useful to them, even though they don't have linguistic competance in the target languages

→ Searching distributed, unstructured, heterogeneous, multilingual data is the goal of CLIR system

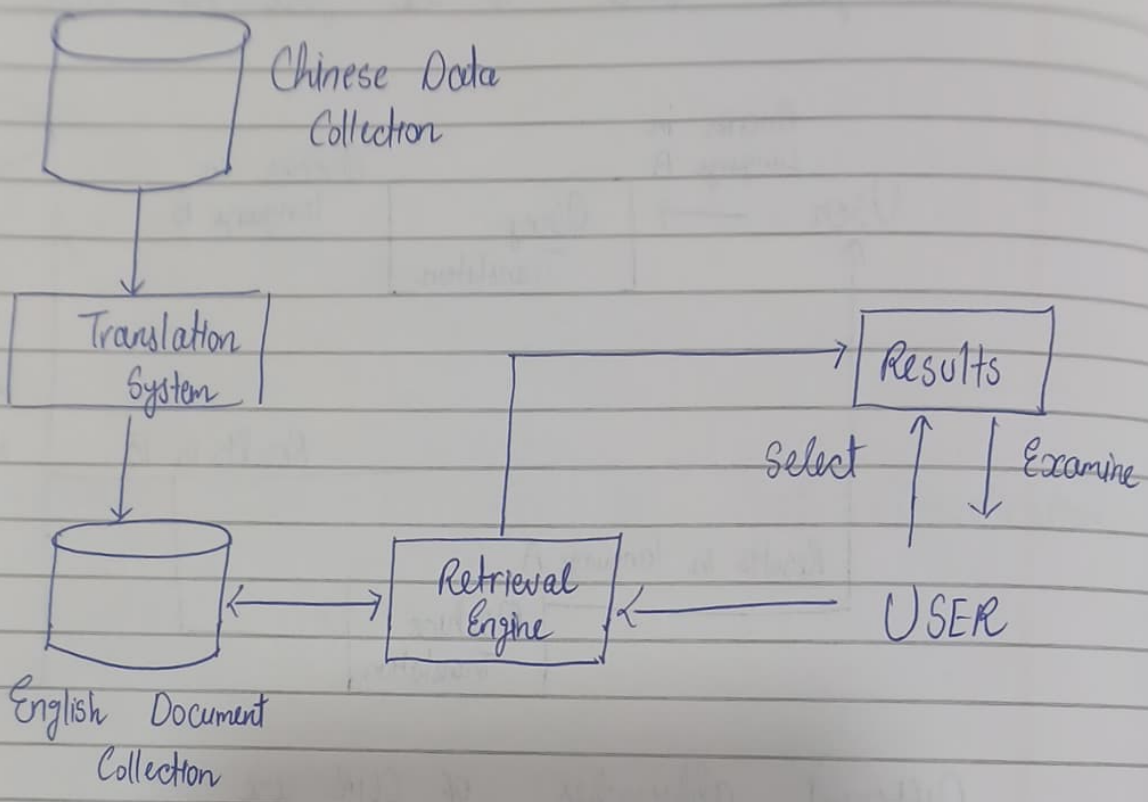


Different approaches of CLIR are

1) Query translation Approach
→ As shown in figure, the query is translated to target document language
→ Most appropriate approach, as the query is shorter and fast to translate than a complete document
→ One disadvantage can be, query translation can suffer from translation ambiguity due to limited context



2) Document translation Approach



→ As shown in Fig., the documents in target language are translated to source language
→ Document translation can provide more accurate translation due to richer contexts
→ Advantage of document translation is, it's easy for user to understand the document easily once it is retrieved

3) Interlingua based approach
→ In this approach of the documents and query are both translated into some common third Interlingua
→ This approach generally requires huge resources as the translation needs to be done online

Q-3 What is Information Extraction ? How does Information Extraction works ?

Ans Information Extraction (IE) is the process of automatically identifying structured information (entities, relations, events, facts) from unstructured text

Eg: Elon Musk is CEO of Tesla
- Entity: Elon Musk (Person)
- Entity: Tesla (Organization)
- Relation: CEO-of (Elon Musk → Tesla)

Main sub-tasks of IE
1) Named Entity Recognition (NER)
2) Relation Extraction
3) Event Extraction
4) Coreference Resolution

1) Named Entity Recognization
→ Identify entities (Person, organization, location, date, money etc)
→ Eg: "Barack Obama was born in Hawaii"
  - Barack Obama → PERSON
  - HAWAII → LOCATION

2) Relation Extraction
→ Find Relation between entities
Eg: ("Elon Musk", CEO-of, "Tesla")

3) Event Extraction
→ Identify events and participants
→ Eg: "Google acquired Youtube in 2006"
  - Event: Acquisition
  - Entities: (Google → Acquirer, YT → Acquiree, 2006 → Date)

A) Coreference Resolution
- Link mentions referring to the same entity
- Example: "Sundar Pichai is the CEO of Google. He joined in 2004" → He: Sundar Pichai.

Q-9 Explain IR-based question answering model with different phases

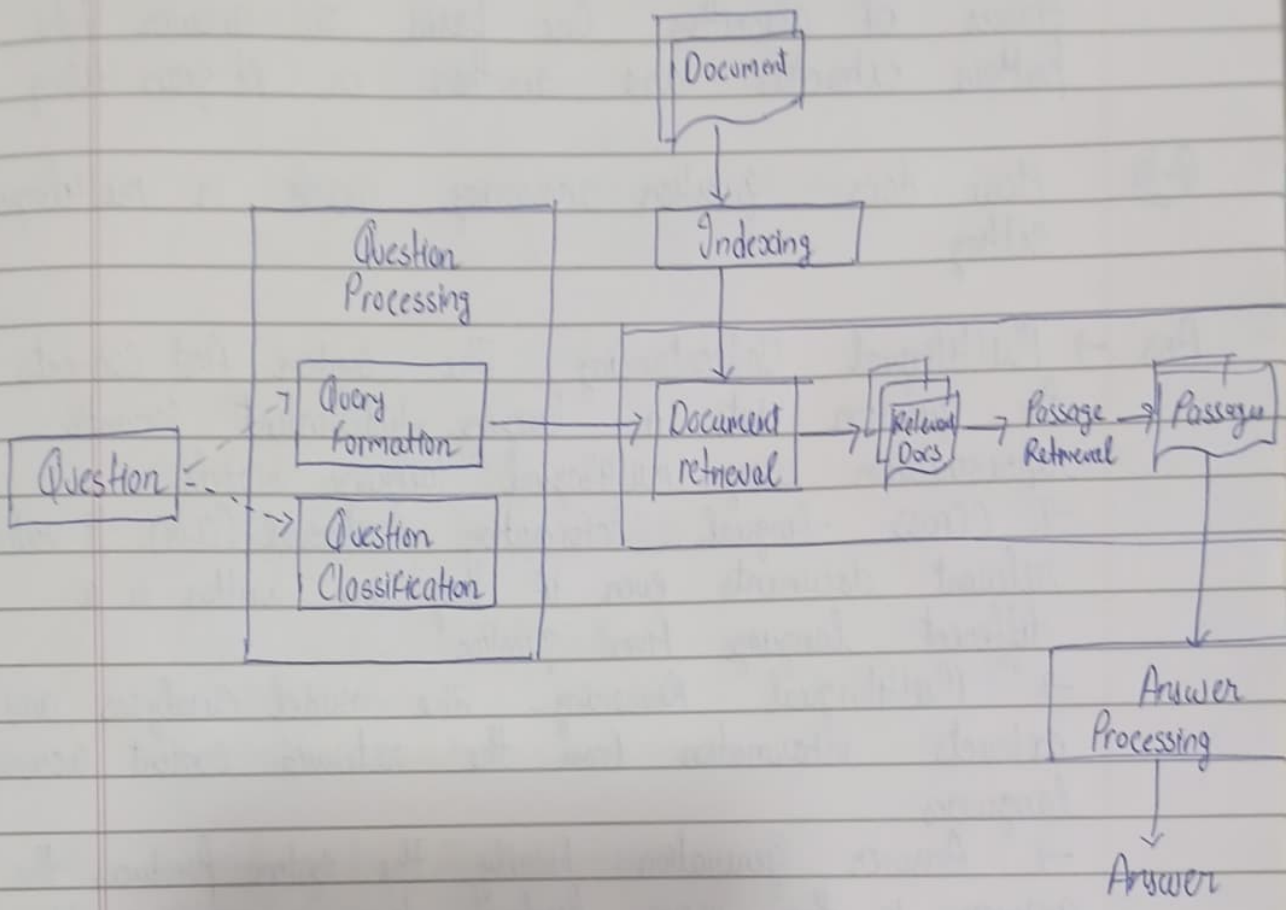
Ans An Information Retrieval (IR) based Question Answering system answers a user's question by retrieving relevant documents from a large collection and then extracting the answer from those documents.

It relies mainly on
- Document Search (IR)
- Ranking
- Answer Extraction

This model is used in search engines, FAQ systems, and domain-specific QA.

Phases of IR-based QA model
1) Question Processing
2) Passage Retrieval
3) Answer Processing

1) Question Processing
→ Two things are to be extracted from question:-
- A keyword query which will be appropriate as an input to the IR system
- An answer type, a specification which will act as a reasonable answer



2) Passage Retrieval

→ The generated query from the previous phase is then used in IR System

o IR system can be,

→ Web Search Engine

→ These passages are filtered and ranked based on how likely they are to contain answers to the question

→ This ranking is done either through hand crafted rules or by supervised training with machine learning techniques

3) Answer Processing

→ In this phase a specific answer is retrieved from the passage

→ This answer extraction can be done by using two classes of algorithm : One based on answer type pattern extraction and another on N gram tiling

Q-10 How does question-answering work in multilingual setting

Ans → Multilingual Understanding : The system first converts the question into a language-independent semantic representation using multilingual language models
→ CCross-Lingual Information retrieval (CLIR): It retrieves relevant documents even if they are written in a different language from question
→ Multilingual Reasoning: The model analyses and extracts information from the retrieved content across languages
→ Answer Generation: Finally, the system produces the answer in the user's prefered language using multilingual generation or translation

Q-11 State different algorithms used for relation extraction

Ans Relation Extraction is the task of identifying semantic relations (eg works for, born-in, part of) between pairs of entities in text

1) Supervised Learning Algorithms
2) Deep Learning Algorithms
3) Semi-Supervised Algo
4) Unsupervised Algo
5) Distant Supervision Algo

1) Supervised Learning Algorithm
   a) Decision Trees
   b) Naive Bayes Classifier
   c) Maximum Entropy model
   d) Support Vector Machines (SVM)
   e) Neural Network Models

2) Deep Learning Algorithm
   a) CNN based
   b) RNN/LSTM based
   c) Transformer-based Models

3) Semi-Supervised Algorithm
   a) Bootstrapping
   b) Pattern-based

4) Unsupervised Algorithm
   a) Clustering Algo
   b) Open IE

5) Distant Supervision Algorithms
   a) Multi-Instance Learning (MIL)
   b) PCNN with Distant Supervision

✨ Application of NLP using Cross-Lingual Information Retrieval (CLIR)

🔹 1️⃣ What is Cross-Lingual IR?

Cross-Lingual Information Retrieval (CLIR) means a user can **ask a question or search in one language** and the system can **retrieve documents or answers written in another language**.

👉 In simple words:

> You search in *Hindi*, but get results from *English documents* (or any other language).

🔹 2️⃣ Concept in NLP:

NLP helps in **translation**, **language detection**, and **meaning matching** between different languages.
CLIR uses these NLP techniques to **bridge the language gap** between the *query* and the *documents*.

🔹 3️⃣ Example Application:

🌍 Multilingual Question Answering System

- **User's Question**: "ભારતના વડા પ્રધાન કોણ છે?" (Gujarati)
- **Step 1**: NLP detects the language (Gujarati).
- **Step 2**: Translates the question into English → "Who is the Prime Minister of India?"
- **Step 3**: IR system searches English documents.
- **Step 4**: Extracts answer "Narendra Modi."
- **Step 5**: Translates it back into Gujarati → "નરેન્દ્ર મોદી."
- ✅ **Final Output**: Narendra Modi (in Gujarati)

🔹 4️⃣ Other Real-Life Examples:

Application	Description
Google Search (Multilingual)	You can search in any language and get results from multiple languages.
Wikipedia Cross-Language Search	Lets users find information in different language versions of Wikipedia.
Cross-Lingual Chatbots	Understands user messages in one language and replies using data in another language.

🔹 5️⃣ Advantages:

- ✅ Removes language barriers.
- ✅ Useful for global information access.
- ✅ Helps in multilingual countries like India.

✅ In short (for 1-line summary):

> Cross-Lingual IR allows users to ask in one language and get information from another language using NLP techniques like translation, language detection, and retrieval — for example, multilingual QA or search systems.

What is the role of Named Entity Recognition in Information Extraction?

Here is a clear and exam-ready answer:

Role of Named Entity Recognition (NER) in Information Extraction (IE)

Named Entity Recognition (NER) is **one of the core components** of Information Extraction. Its main role is to **identify and classify important real-world entities from text**, such as:

- PERSON (e.g., "Sujal", "Narendra Modi")
- ORGANIZATION (e.g., "Google", "AIIMS")
- LOCATION (e.g., "Surat", "India")
- DATE, TIME, MONEY, PERCENTAGE, etc.

★ Role of NER in Information Extraction

1. Identifying key entities that participate in relationships

IE systems extract relations such as *works-for, located-in, born-in*.
NER helps detect the **entities** between which such relations hold.

Example:
Sentence: *"Sujal won first prize in a hackathon in Surat."*
NER finds:

- Sujal → PERSON
- Surat → LOCATION

These entities are needed to detect relations:
(Sujal, won-prize, hackathon), (hackathon, located-in, Surat)

---

2. Acts as the first step in Relation Extraction

Before extracting relations, the system must know **what the entities are**.
NER acts as the **input** for relation classifiers.

---

3. Helps in building knowledge bases

Extracted entities are stored in knowledge graphs (like DBpedia, Wikidata).
NER identifies the nodes of the graph.

---

4. Improves accuracy of downstream IE tasks

NER helps reduce noise by focusing only on relevant named entities.
This improves performance in:

- Event extraction
- Opinion extraction
- Question answering
- Information retrieval

---

5. Disambiguates similar-looking text

NER helps distinguish between common words and entity names.

Example:
"Apple released a new phone." → Apple = ORGANIZATION
"I ate an apple." → apple = FRUIT

This clarification is essential for correct extraction.

---

6. Helps in indexing and searching information

NER allows indexing documents based on entities.
This makes large-scale IE systems faster and more accurate.

State the advantages of Cross-Lingual IR.

Here is a **clear, short, exam-ready answer** for:

Advantages of Cross-Lingual Information Retrieval (CLIR)

Cross-Lingual IR allows users to **search in one language and retrieve documents in another.**
Its advantages include:

✅ Advantages of CLIR (4–5 Marks Answer)

1. Access to More Information

Users can retrieve documents written in many languages, increasing the amount of available knowledge.
Example: Asking in English but getting relevant Hindi or Japanese documents.

---

2. Removes Language Barriers

Users do not need to know the language of the documents.
CLIR automatically translates queries or documents.

---

3. Useful for Multilingual Countries & Domains

Countries like India, Switzerland, Canada benefit as data exists in many languages.
Also useful for multilingual websites, global companies, and research communities.

---

4. Improves Search Quality

Sometimes the best information is available in another language.
CLIR increases relevance by searching across multiple languages.

---

4. Improves Search Quality

Sometimes the best information is available in another language.
CLIR increases relevance by searching across multiple languages.

---

5. Helps in global applications

CLIR is essential for:

- International news search
- Multilingual digital libraries
- Cross-border business intelligence
- Multilingual question-answering systems

---

6. Cost Effective

No need to hire human translators for searching documents.
Machine translation + IR handles the process.

---

7. Supports Cultural and Linguistic Diversity

Encourages access to content from different cultures, improving global communication and understanding.

Ch-5 Machine translation

Q-1 Describe Knowledge Based Machine Translation System [S-24, W-22,23]

Ans KBMT is a traditional approach that uses deep linguistic knowledge and world knowledge to translate text from one language to another

Stages of KBMT

1) Source Language Analysis (Understanding Input)
→ Performs syntatic, semantic and morphological analysis
→ It converts the source sentence into an intermediate meaning representation (like conceptual graph, predicate logic)

Eg: Input: "Sujal ate an apple"
Internal meaning: EAT (Sujal, apple)

2) Mapping between two languages
o The meaning representation is mapped to the target language concepts

3) Target Language Generation (Producing Output)
→ Generates grammatical sentences in target language
→ Uses rules for word ordering, inflection, agreement and morphology

Q-2 What is Machine Translation? What are the applications of Machine Translation

Ans

Page No.
Date

Machine Translation is the process of automatically translating text or speech from one language to another using computers
It uses linguistic rules, statistical models, or neural networks to convert a sentence in the source language into a meaningful sentence in the target language

Applications of Machine Translation
1) Multilingual Communication
2) Web Content Translation
3) International Business & E-commerce
4) Education and Learning
5) Travel & Tourism
6) & Social Media and Messaging Apps.
7) Healthcare & Government Services

Q-3) Describe Rule Based Machine Translation System

Ans RBMT is an approach where translation is performed using manually created linguistic rules
→ Heavily relied on grammar, morphology, syntax.
→ Works well for limited domains, but hard to scale

Types of RBMT
1) Direct Approach: Focuses on direct word-for-word translation and basic grammatical reordering
2) Transfer-Based Approach: Involves multiple steps of analysis, transformation and synthesis for better accuracy
3) Interlingua Approach: Translates the source text into an abstract, language-independent meaning (an interlingua) and then generates target language from that meaning

Q-4 Elaborate on how Parameter learning is conducted in Statistical Machine Translation with an example [S-24, 25]

Ans

In SMT, translation is modeled as a probability problem

For a source sentence f (eg French), the system chooses the English sentence e with highest probability

$$\hat{e} = \arg\max_{e} P(e|f)$$

IBM Model - 1

→ Simplest statistical machine translation model

→ It's job is to learn word-to-word translation probabilities using a parallel corpus

Eg:

French: le chat noir

English: the black cat

It learns $t(e|f)$ = Probability that source word f translates to English word e

Example Probabilities after training e

- $t(\text{the}|\text{le}) = 0.9$
- $t(\text{cat}|\text{chat}) = 0.95$
- $t(\text{black}|\text{noir}) = 0.98$

1) Choose length of French Sentence

$J = 3$

2) Choose Alignment A

French word	English word
Le	the
chat	cat
noir	black

Each alignment is equally likely

$$P(A|E) = \frac{\epsilon}{(I+1)^J}$$

length of french = J
length of english = E
Let ϵ be constant for probability of choosing length J

→ which is a combined probability of choosing length J and $(I+1)^J$ is one of possible alignments

These probabilities can be combined as

$$P(F, A|E) = P(F|E,A) \times P(A|E)$$

$$= \frac{E}{(I+1)} \prod_{j=1}^{J} t(f_j|e_{a_j})$$

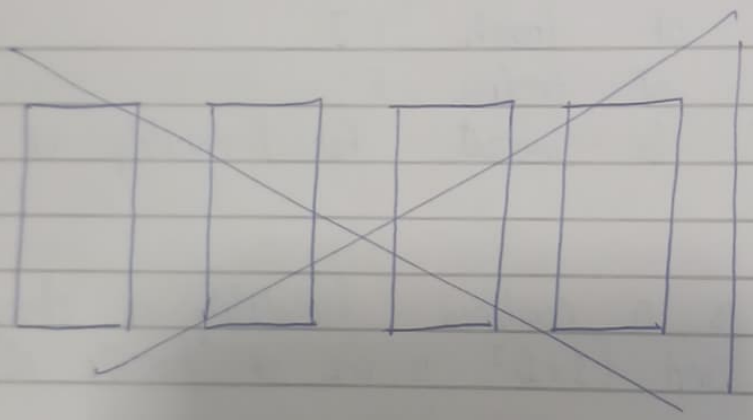
Probability of English sentence

$$P(E|F) = \sum_{A} P(E, A|F) = \frac{\epsilon}{(I+1)} \prod_{j=1}^{J} t(f_j|e_{a_j})$$

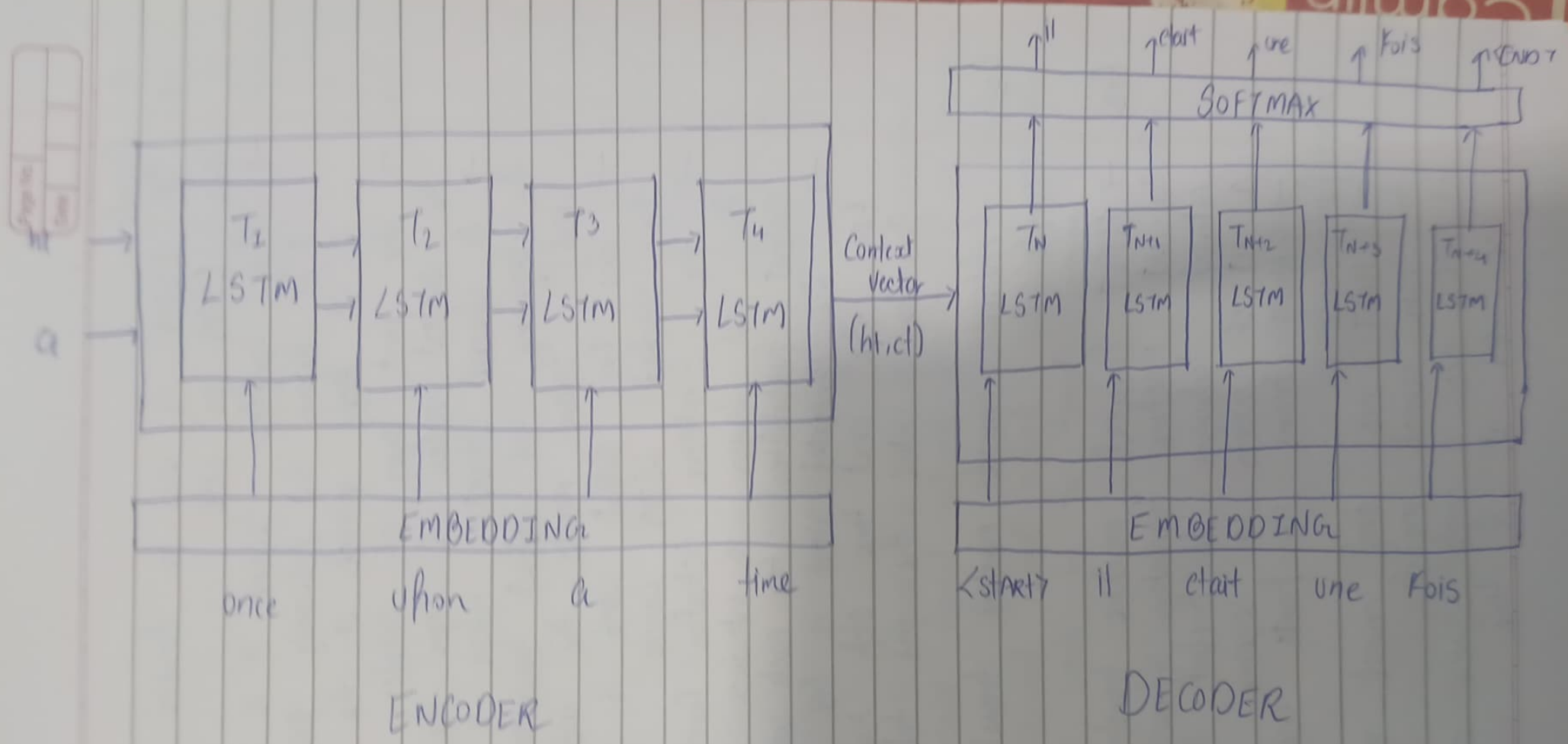
This equation represents generative probability model of IBM model 1.

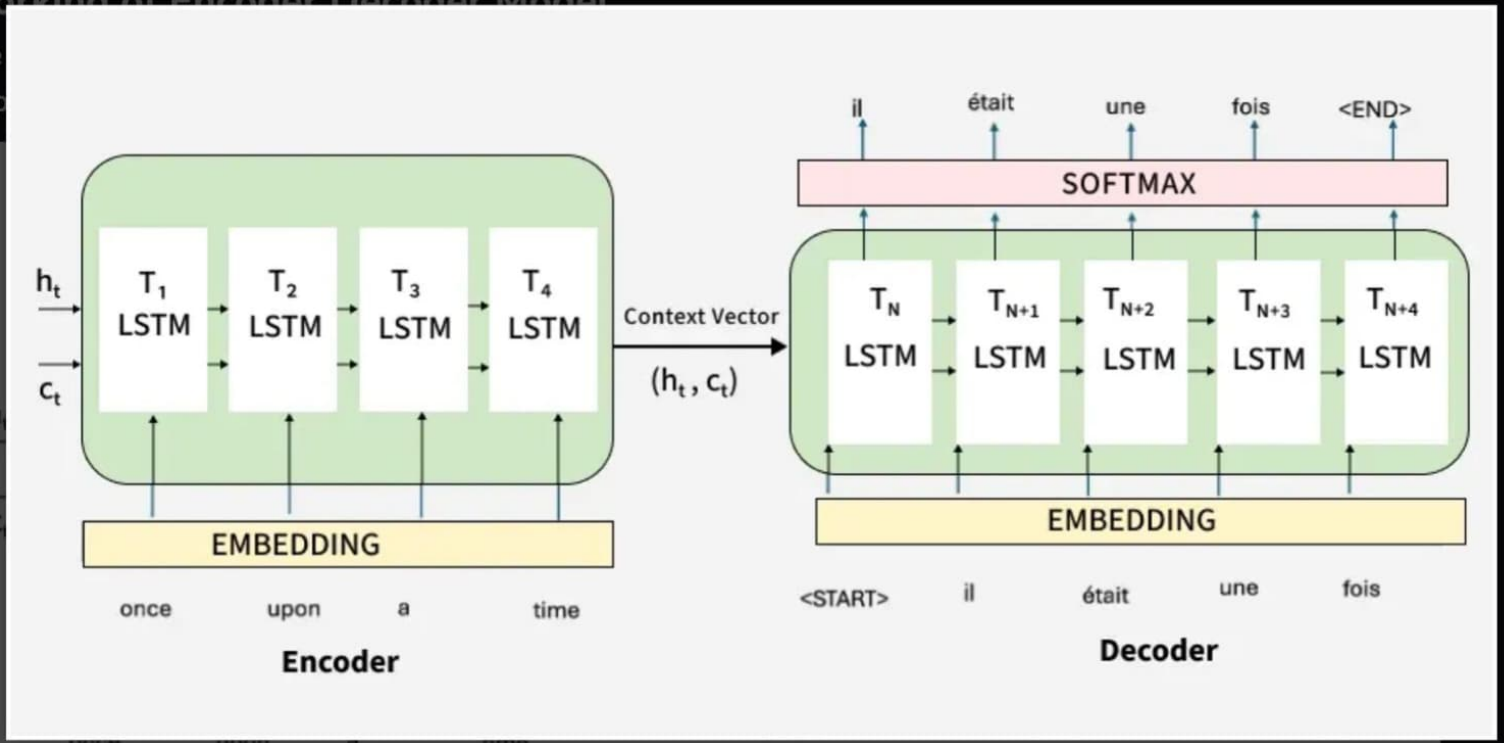
Q.5 Illustrate and provide a comprehensive explanation of the Encoder-Decoder Architecture [S-24, W-23]

Ans The Encoder-Decoder architecture is a neural network framework that converts an input sentence into a fixed-length context vector using an encoder and uses a decoder to transform this vector into an output sequence.



- Encoder
  - → The encoder is the feeding end of system
  - → It understands the input sentence and reduces its dimension
  - → The input sequence is summarized into fixed-size vector known as the context vector (c)
  - → The context vector acts as an input to the decoder, which generates the output sequence until an end token is reached
  - → The models following this structure are called encoder-decoder models
  - → This architecture can handle input and output sequences of variable length


T1
LSTM
T2
LSTM
T3
LSTM
T4
LSTM
Context Vector
(h1, ct)
TN
LSTM
TN+1
LSTM
TN+2
LSTM
TN+3
LSTM
TN+4
LSTM
SOFTMAX
il
était
une
fois
<END>
EMBEDDING
EMBEDDING
Once
upon
a
time
<start>
il
était
une
fois
ENCODER
DECODER


il
était
une
fois
<END>
SOFTMAX
h_t
c_t
T_1 LSTM
T_2 LSTM
T_3 LSTM
T_4 LSTM
Context Vector
(h_t, c_t)
T_N LSTM
T_{N+1} LSTM
T_{N+2} LSTM
T_{N+3} LSTM
T_{N+4} LSTM
EMBEDDING
EMBEDDING
once
upon
a
time
<START>
il
était
une
fois
Encoder
Decoder

Page No.
Date

o Decoder

→ If LSTM is used in decoder, the decoder usually uses LSTM as well

→ More complex because it is in "aware state"
  - It knows the previously generated words
  - It keeps a track of the previous hidden state

→ The first layer of decoder is initialized with the context vector C from the encoder

→ A special <start> token is used to indicate the beginning of output generation, and a special <end> token marks the end.

→ The first output word is generated by running the stacked LSTM Layers.

→ The SoftMax activation function is applied to the last layer to introduce non-linearity and output probabilities for each word.

→ The generated word is fed back into the decoder for the next time step, repeating the sequence generation until the end token

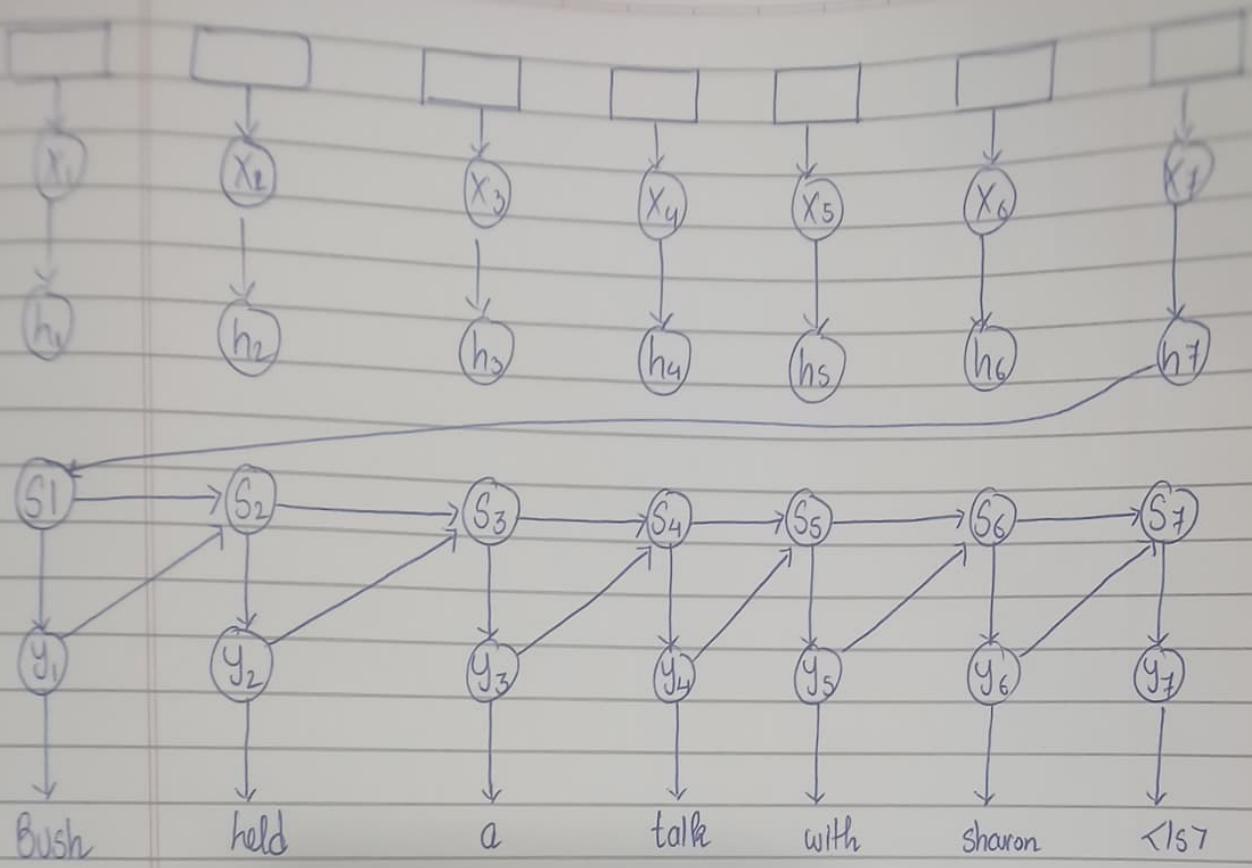
Q-6 Explain neural machine translation with suitable example

Ans → NMT (neural machine translation) is a state-of-the-art machine translation approach that uses neural network techniques to predict the likelihood of a set of words in sequence

→ NMT trains its parts end-to-end to maximize performance

→ NMT uses deep neural networks and A.I to train neural models

→ NMT uses a single sequence model that produces one word at a time



→ As shown in Figure, the NMT uses a bidirectional recurrent model neural network, also called an encoder, to process a source sentence into vectors For a second recurrent neural network, called the decoder, to predict words in the target language

Q-7 Explain the dimensions used to evaluate Machine Translation

Ans
1) Accuracy
2) Fluency
3) Fidelity
4) Comprehensibility
5) Terminology Consistency
6) Speed
7) Localization
8) Human v/s Automatic Evaluation

xplain the dimensions used to evaluate Machine Translation.

When evaluating **Machine Translation (MT)**, we assess the system's performance across multiple **dimensions** that capture different aspects of translation quality. Here's a clear breakdown:

---

1. Accuracy / Adequacy

- **Definition:** Measures how much of the **meaning of the source text** is preserved in the translation.
- **Focus:** Semantic correctness, completeness, and factual consistency.
- **Example:** If the source says *"The cat is on the mat"* and the translation says *"The cat sits near the mat"*, adequacy is partially compromised because the exact relation is lost.

---

2. Fluency / Naturalness

- **Definition:** Evaluates how **grammatically correct and natural-sounding** the translation is in the target language.
- **Focus:** Language quality, word order, and idiomatic expressions.
- **Example:** A literal translation might be accurate but awkward: *"I have hunger"* instead of *"I am hungry"*.

---

3. Fidelity

- **Definition:** Measures **faithfulness** to the original text without adding or omitting information.
- **Focus:** Ensures the translation neither exaggerates nor understates the meaning.
- **Example:** Translating *"He won the race"* as *"He participated in the race"* loses fidelity.

---

4. Comprehensibility / Readability

- **Definition:** Assesses whether the **target audience can understand** the translated text easily.
- **Focus:** Smoothness, coherence, and logical flow.

---

4. Comprehensibility / Readability

- **Definition:** Assesses whether the **target audience can understand** the translated text easily.
- **Focus:** Smoothness, coherence, and logical flow.

---

5. Terminology Consistency

- **Definition:** Checks whether **technical or domain-specific terms** are translated consistently throughout the text.
- **Focus:** Especially important in medical, legal, or technical MT.
- **Example:** Translating *"cardiac arrest"* consistently across all sentences.

---

6. Cultural Appropriateness / Localization

- **Definition:** Evaluates if the translation **fits the cultural and contextual norms** of the target language audience.
- **Focus:** Idioms, units of measurement, date formats, or culturally specific references.
- **Example:** Translating *"Fourth of July celebration"* appropriately for a non-US audience.

---

7. Speed / Efficiency (optional)

- **Definition:** Measures how quickly and efficiently the MT system produces translations.
- **Focus:** Important for real-time translation systems.

---

8. Human vs Automatic Evaluation

- **Human evaluation:** Experts judge **adequacy, fluency, fidelity**, and cultural relevance.
- **Automatic metrics:** BLEU, METEOR, TER, and others quantify overlap with reference translations.

✅ **Summary:**
Machine Translation quality is multi-dimensional, involving **meaning preservation (accuracy, fidelity), language quality (fluency, readability), consistency, cultural fit**, and sometimes **efficiency**. A good MT system balances all these dimensions rather than excelling in just one.

Q-8 Write a detailed note on SMT

Ans → In Statistical MT the result is focussed and not the process like the classical MT approaches explained earlier
→ It focuses on phrase-based translations for better fluency
→ While SMT was a significant advance, it has largely been superseded by NMT, due to NMT's ability to handle more complex linguistic nuances through deep learning

0 Working of SMT

1) Data Analysis: SMT systems require very large collections of parallel texts (the same content in two languages) to learn from

2) Statistical Modeling: Statistical models are generated from these bilingual corpora to determine the likelihood of specific words and phrases being translated between languages

3) Bayes theorem: The core of SMT

→ Let Foreign language sentence $F = f_1, f_2, \ldots f_m$ which will be converted to English
→ French & Spanish will be considered as foreign languages but target language is English every time
→ Acc. to probabilistic modes the English sentence $E = e_1, e_2 \ldots e_i$ is best one if it has highest probability $P(E|F)$

If we consider a noisy channel model then this can be rewritten using Bayes rule as

$$\hat{E} = \text{argmax}_E \; P(E|F)$$

$$= \text{argmax}_E \; \frac{P(F|E)\,P(E)}{P(F)}$$

$$= \text{argmax}_E \; P(F|E)\,P(E)$$

4) Statistical Modeling

4) Translation Model: The probability of source text being a translation of target text

5) Language Model: The probability of target language sentence being fluent and natural

6) Decoding: A decoder then uses these models to find the largest sentence that maximizes the overall probability, effectively choosing the most likely translation.

Types of SMT

1) Phrase based SMT: the most common type of SMT, which translates whole sequences of words (phrases) rather than individual words, leading to more natural translations

2) Data Driven: SMT is a data driven approach, meaning it builds its understanding of translation directly from existing human translations in the corpus

Page No.
Date

Q-9 Explain Problems of Machine Translation (MT) OR what is need of MT? Discuss it's Problems

Ans

- Need of Machine Transaltion
  - → Breaking Language Barriers
  - → Information accessibility
  - → Cross lingual Communication
  - → Foundation for Multilingual NLP applications
  - → Automation & Speed
  - → Support for low Resource languages
  - → AI advancement

- Problems of Machine Translation

1) Ambiguity
2) Word Order Differences
3) Idioms and Expressions
4) Context Understanding
5) Cultural Differences
6) Domain Dependency

Q-10 How Direct Machine Translations approach works in NLP [7m]
OR Briefly discuss Direct Machine Translation approach

Ans → In direct translation word by word translation of source language text is done in order of appearance of words
→ Every word is mapped on target word directly
→ To accomplish this there is a need of large bilingual dictionary

Need of Machine Translation

1. 🌐 Breaking Language Barriers

- The world has **7,000+ languages**, but most digital content exists in just a few (English, Chinese, etc.).
- MT helps **translate text and speech automatically**, making information accessible to everyone — regardless of language
- Example: Translating government or healthcare documents for citizens in multiple languages.

Need of Machine Translation

2. 📚 Information Accessibility

- Enables access to knowledge across linguistic boundaries.
- Academic papers, news, and technical documents can be **instantly translated** into multiple languages.
- E.g., Translating scientific research from English to Hindi, Gujarati, Tamil, etc.

Need of Machine Translation

3. 💬 Cross-Lingual Communication

- Vital for **global business**, **education**, **tourism**, and **diplomacy**.
- Chatbots and customer support systems use MT to understand and respond to users in their native languages.
- People can communicate to each other irrespective of in which language they speak. E.g. Person1 speaks in Gujarati, "તમે કેમ છો?"
- Person2 after understanding in his marathi language, and replies, "मी ठीक आहे"
- so the communication is there in cross lingual

Need of Machine Translation

4. 🧠 Foundation for Multilingual NLP Applications

- MT is a **base for other NLP tasks**, such as:
  - **Cross-lingual Information Retrieval** (searching English data using Hindi queries),
  - **Multilingual Question Answering**,
  - **Multilingual Sentiment Analysis**,
  - **Speech Translation** systems (e.g., live captioning).

Need of Machine Translation

5. ⚙️ Automation & Speed

- Human translation is slow and expensive.
- MT systems can translate **millions of words per second**, crucial for real-time applications (e.g., news feeds, social media).

6. 🧾 Support for Low-Resource Languages

- Helps preserve and digitize **regional and minority languages** by creating parallel corpora.
- Aids in **language learning** and **revitalization projects.**

Need of Machine Translation

7. 💡 AI Advancement

- MT drives innovation in NLP —
  - Transformers (BERT, GPT, T5) were originally developed for translation tasks.
  - These models now power most modern NLP systems.

Problems of Machine Translation (Any 6)

1. Ambiguity

Languages contain many forms of ambiguity:

- **Lexical ambiguity:** A word has multiple meanings (e.g., *bank* → river bank / financial bank).
- **Syntactic ambiguity:** Sentence structure can be interpreted in different ways.
- **Semantic ambiguity:** Intended meaning may be unclear.

  MT systems often select the wrong interpretation, causing incorrect translations.

---

2. Word Order Differences

Different languages follow different sentence patterns:

- English: **SVO**
- Hindi: **SOV**
- Arabic: **VSO**

  Handling reordering, especially in long or complex sentences, is difficult and leads to unnatural or incorrect translations.

---

3. Idioms and Expressions

Idioms cannot be translated literally because their meanings are figurative.

Example: *"Kick the bucket" ≠ "बाल्टी को लात मारना"*

MT systems often translate idioms word-for-word, losing the actual meaning.

4. Context Understanding

MT has limited ability to maintain broader context:

- Pronoun confusion (**he/she/it**)
- Coreference issues (e.g., *"John met Sam. He was angry."*)
- Lack of real-world knowledge
  This leads to inaccurate translations across multi-sentence texts.

---

5. Cultural Differences

Some cultural concepts do not have direct equivalents in other languages:

- Honorifics (e.g., Japanese politeness levels)
- Relationship-specific terms in Indian languages
  MT systems struggle to capture cultural tone, politeness, and cultural references.

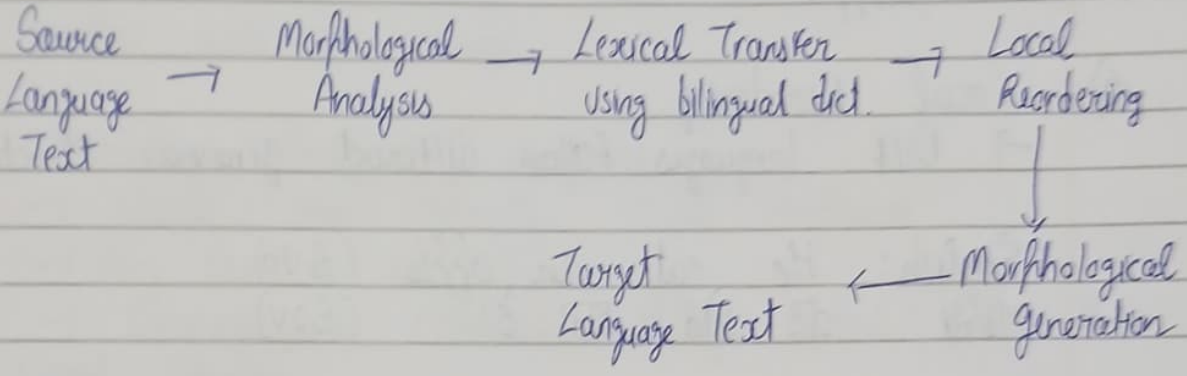
---

6. Domain Dependency

A translation model trained on one domain may perform poorly in another:

- Medical → Legal → Literary texts
  Different domains require specific vocabulary and style, so MT needs domain-specific datasets for accurate results.

→ Each entry in this dictionary can be considered as a small program for translation of one word



1) Source Language Text
→ This is original sentence that we want to translate
→ Eg: "He eats an apple"

2) Morphological Analysis
→ In this step, the system breaks each word into its root form and identifies grammatical information such as:-
- tense
- number
- gender
- prefixes / suffixes

Eg:
"eats" → root word : eat, tense: present

3) Lexical Transfer
→ Main step
→ Every source word is looked up in bilingual dictionary (eg: English → Hindi)
→ Each dictionary entry gives the direct target equivalent

Eg:
"He" → "वह"
"eat" → "खाता"
"apple" → "सेब"

4) Local Reordering
→ Diff languages follow different grammar structures (SVO & SOV)

English: He eats an apple (SVO)
Hindi: "वह सेब खाता है" (SOV)

5) Morphological Generation
→ Finally, the system applies target-language grammar rules to generate correct form of words
This ensures correct:-
- Verb conjugation
- tense
- gender
- number
- agreement

Eg:
"eat" → "खाता है" (male) or "खाती है" depending on subject

6) Target Language Text
→ This is the final translated output after reordering and morphological generation

Eg:
"वह सेब खाता है"

Q-1 Explain morphological typology

Ans Morphological typology is the classification of languages based on how they form words using morphemes (smallest units of meaning)

1) Isolating (Analytic) Languages
→ Words are typically single morphemes
→ Very few or no affixes
→ Grammar is expressed through word rather than word structure
Examples: Chinese, Vietnamese

2) Agglutinative languages

3) Fusional Languages

4) Polysynthetic languages.

Explain morphological typology.

Here is a **clean, exam-ready explanation of Morphological Typology**:

✅ Morphological Typology – Explained

Morphological Typology is the classification of languages based on how they **form words** using **morphemes** (the smallest units of meaning).
It studies **how words are structured**, how morphemes combine, and how different languages express grammatical information.

In simple words:
Morphological typology = how languages build words and express grammar.

✅ Major Types of Morphological Typology

1. Isolating (Analytic) Languages

- Words are typically **single morphemes**.
- Very **few or no affixes**.
- Grammar is expressed through **word order** rather than word structure.

Examples:

Chinese, Vietnamese

Example:

English "I go" → Chinese: 我 去 (each word = one unit)

2. Agglutinative Languages

- Words are formed by adding **multiple affixes**, each expressing **one grammatical function**.
- Morpheme boundaries are **clear and separable**.
- Easy to segment.

Examples:

Turkish, Japanese, Tamil

Example:

Turkish: "evlerinizden"
ev (house) + **ler** (plural) + **iniz** (your) + **den** (from)

---

3. Fusional (Inflectional) Languages

- Each affix may carry **multiple grammatical meanings** at once.
- Morphemes are not clearly separable.
- Word forms "fuse" several features like tense + person + number.

Examples:

Latin, Spanish, Russian, Hindi (partly)

Example:

Spanish verb "hablo" = *I speak*

- "o" indicates **1st person + singular + present tense** all together.

---

4. Polysynthetic Languages

- Very **long, complex words** that contain **many morphemes**.
- A single word can express what would be a **full sentence** in other languages.
- Combines multiple concepts into one.

Examples:

Inuktitut, Mohawk

Example:

One word may express:
"He-gives-it-to-me" → as one long word.

Q-11 Explain morphological typology

Ans Morphological typology is the classification of languages based on how they form words using morphemes (smallest units of meaning)

1) Isolating (Analytic) Languages
→ Words are typically single morphemes
→ Very few or no affixes
→ Grammar is expressed through word rather than word structure
Examples: Chinese, Vietnamese

2) Agglutinative languages

3) Fusional Languages

4) Polysynthetic languages

Q-12 Define SVO, VSO, SOV

Ans (1) Subject - Verb - Object (SVO)
→ Common in English
Eg: "He (S) eats (V) apples (O)"

(2) VSO (Verb - Subject - Object)
→ Arabic & Irish
Eg: "Eats he apple"

(3) SOV
→ Hindi, Japanese & Turkish
→ "वह सेब खाता है"
"He (S) apples (O) eats (V)"

Q-13 Define Referential Density with example

Ans Referential Density refers to how much real-world information (references to people, objects, events, places, actions etc) is packed into a stretch of language

"The cat sat on the mat"

Content words: cat, mat, sat → 3
Total words: 6
→ Referential density = 3/6 = 0.5 (moderate)

If High density → Sentence contains meaningful events
If low density → Contains few references

Q-14 Write Wordpiece Algorithm

Ans

✅ WordPiece Algorithm – Expanded Bullet Points (6 Marks)

- WordPiece is a **subword tokenization algorithm** used in models like **BERT, ALBERT, DistilBERT**.
- Begins with a **minimal vocabulary** containing all **characters** and the **UNK token**.
- The corpus is scanned and the algorithm **counts frequencies of adjacent token pairs**.
- At each step, it **selects the pair whose merge gives the highest likelihood** under a language model.
- The selected pair is **merged into a new subword unit**, e.g., "t" + "he" → "the".
- This merge process continues until the vocabulary reaches a **predefined size** (like 30k).
- Produces a vocabulary containing **common words** and **subwords for rare words** (e.g., "play", "##ing").
- Greatly reduces **OOV (out-of-vocabulary)** problems and makes representation **efficient and compact**.

✅ Example of WordPiece Tokenization

Suppose the word is: **"playing"**

Step 1: Start with characters

```
css
p l a y i n g
```

Step 2: Merge most frequent pairs

Assume the algorithm has learned these subwords from training:

- "play"
- "##ing"

("##" means continuation of a word)

Step 3: Apply WordPiece to the word

The word **playing** becomes:

```
nginx
play  ##ing
```

Final Tokenization Output

["play", "##ing"]

Here is a **clean, exam-ready table** differentiating **Statistical MT (SMT)** and **Rule-Based MT (RBMT)**.

✅ Difference: Statistical MT vs Rule-Based MT

Feature	Rule-Based MT (RBMT)	Statistical MT (SMT)
Basic Idea	Uses linguistic rules, grammar, dictionaries, morphology	Uses statistical models learned from large parallel corpora
Knowledge Source	Hand-crafted rules + bilingual dictionaries	Data-driven probabilities from sentence pairs
Language Coverage	Good for languages with strong linguistic resources	Works well if large corpora are available
Development Effort	Time-consuming (needs linguists to write rules)	Easier to build if data is available
Handling Ambiguity	Uses linguistic rules → often rigid	Uses probabilities → better at resolving ambiguity
Quality	More grammatical but sometimes unnatural	More fluent but sometimes grammatically incorrect
Domain Adaptation	Hard to adapt (rules must be rewritten)	Easy to retrain on domain-specific data
Output Style	Literal/word-by-word translations common	More context-aware, flexible translations

---

If you want, I can also add **examples** or provide a **4-mark short version**.

Provide a concise explanation of translation divergence.

Here is a **concise, exam-friendly explanation**:

✅ Translation Divergence (Definition)

Translation divergence refers to the differences in how two languages express the *same meaning*.
Even if the idea is identical, the **structure, word order, grammar, or lexical choices** may differ completely between languages.

✅ Simple Example

English → Hindi
"I like tea."
Literal: *मैं चाय पसंद करता हूँ* (SOV)
Structure changes → **word order divergence**.

✅ Short Explanation

Translation divergence happens because languages do not map word-to-word directly.
They differ in **syntax**, **morphology**, **semantics**, **idioms**, and **cultural expressions**, leading to non-parallel translation patterns.

If you want, I can also give **types of divergence** (structural, lexical, thematic, etc.) for more marks.

What are the characteristic required for neural machine translation?

Here is a clear **exam-ready answer** for **Characteristics required for Neural Machine Translation (NMT)** — suitable for 4 to 6 marks.

✅ Characteristics Required for Neural Machine Translation (NMT)

1. Large Parallel Corpora

- NMT needs huge amounts of bilingual sentence pairs to learn accurate mappings between source and target languages.

2. Powerful Computational Resources

- Requires GPUs/TPUs because training deep neural networks is computationally expensive.

3. Continuous Vector Representations

- Words must be represented as **embeddings** (dense vectors) so the model can learn semantic relationships.

4. End-to-End Learning Framework

- NMT systems learn translation in a single unified model (encoder–decoder) instead of separate modules.

5. Ability to Capture Context

- Uses RNNs, LSTMs, GRUs, or Transformers to consider long-range dependencies and sentence-level meaning.

6. Attention Mechanism

- Essential for aligning source and target words; helps the decoder focus on relevant parts of the input.

7. Generalization to Unseen Words

- Uses subword units (BPE/WordPiece) to handle rare or unknown words better than SMT and RBMT.

8. Robust Training Data Quality

- Needs noise-free, domain-consistent, grammatically correct parallel sentences for high-quality translation.