# UNIT 1

## MCQ

1. What is the main challenge/s of NLP?

a) Handling Ambiguity of Sentences
b) Handling Tokenization
c) Handling POS-Tagging
d) All of the mentioned

2. Choose form the following areas where NLP can be useful.

a) Automatic Text Summarization
b) Automatic Question-Answering Systems
c) Information Retrieval
d) All of the mentioned

3. Which of the following includes major tasks of NLP?

a) Automatic Summarization
b) Discourse Analysis
c) Machine Translation
d) All of the mentioned

4. Which library in Python is commonly used for NLP tasks?

a) NumPy
b) TensorFlow
c) NLTK
d) Matplotlib

5. Which of the following is an example of natural language generation?

a) Converting speech to text
b) Translating a document from English to French
c) Writing a news article
d) Analyzing social media posts

## Short Questions/Answers

### 1. List and explain the challenges of NLP

#### a) Language differences

The human language and understanding is rich and intricated and there many languages spoken by humans. Human language is diverse and thousand of human languages spoken around the world with having its own grammar,

vocabular and cultural nuances. Human cannot understand all the languages and the productivity of human language is high. There is ambiguity in natural language since same words and phrases can have different meanings and different context. This is the major challenges in understating of natural language.

**b) Misspellings and Grammatical Errors**

Overcoming Misspelling and Grammatical Error are the basic challenges in NLP, as there are different forms of linguistics noise that can impact accuracy of understanding and analysis.

**c) Words with Multiple Meanings**

Words with multiple meaning plays a lexical challenge in ***Nature Language Processing*** because of the ambiguity of the word. These words with multiple meaning are known as polysemous or homonymous have different meaning based on the context in which they are used.

d) **Training Data**

The training data given to an NLP system determines its capabilities. If you feed the system inaccurate or skewed data, it will learn the incorrect things or learn inefficiently.

e) **Uncertainty and False Positives**

When an NLP detects a term that should be intelligible and/or addressable but can't be adequately replied to, it's called a false positive. The idea is to create an NLP system that can identify its own limits and clear up uncertainty using questions or hints.

## 2. List various applications of NLP and discuss any 2 applications in detail.

### Applications of Natural Language Processing

- **Sentiment Analysis.**
- **Text Classification.**
- **Chatbots & Virtual Assistants.**
- **Text Extraction.**
- **Machine Translation.**
- **Text Summarization.**
- **Market Intelligence.**
- **Auto-Correct.**

a) **Text Classification:** Texts are a form of unstructured information that possesses very prosperous records inside them. Text Classifiers categorize and arrange exceptionally a great deal with any form of textual content that we use currently. Since texts are unstructured, analyzing, sorting, and classifying them can be very challenging and time-consuming and occasionally even tedious work for humans, no longer to point out all the mistakes that human beings are susceptible to make in the process. This is where Text Classification comes in to picture to serve its motive of performing the stated duties with greater scalability and accuracy.

b) **Sentiment Analysis:** Feedback is one of the fundamental factors of true communication. Be it a brand-new film or a trendy tech that's currently launched, the response of the supposed target audience is what makes or breaks them. Hence, inspecting people's sentiment in the direction of a product is necessary now greater than ever.

## 3. Explain the origin and evolution of NLP

### History of Natural Language Processing (NLP)

The history of NLP (Natural Language Processing**)** is divided into three segments that are as follows:

### The Dawn of NLP (1950s-1970s)

In the 1950s, the dream of effortless communication across languages fueled the birth of NLP. Machine translation (MT) was the driving force, and **rule-based systems** emerged as the initial approach.

### How Rule-Based Systems Worked:

These systems functioned like complex translation dictionaries on steroids. Linguists meticulously crafted a massive set of rules that captured the grammatical structure (syntax) and vocabulary of specific languages.

### The Statistical Revolution (1980s-1990s)

- **A Shift Towards Statistics:** The 1980s saw a paradigm shift towards statistical NLP approaches. Machine learning algorithms emerged as powerful tools for NLP tasks.
- **The Power of Data:** Large collections of text data (corpora) became crucial for training these statistical models.
- **Learning from Patterns:** Unlike rule-based systems, statistical models learn patterns from data, allowing them to handle variations and complexities of natural language.

**The Deep Learning Era (2000s-Present)**

- **The Deep Learning Revolution:** The 2000s ushered in the era of deep learning, significantly impacting NLP.
- **Artificial Neural Networks (ANNs):** These complex algorithms, inspired by the human brain, became the foundation of deep learning advancements in NLP.
- **Advanced Architectures:** Deep learning architectures like recurrent neural networks and transformers further enhanced NLP capabilities. Briefly mention these architectures without going into technical details.

**The Advent of Rule-Based Systems**

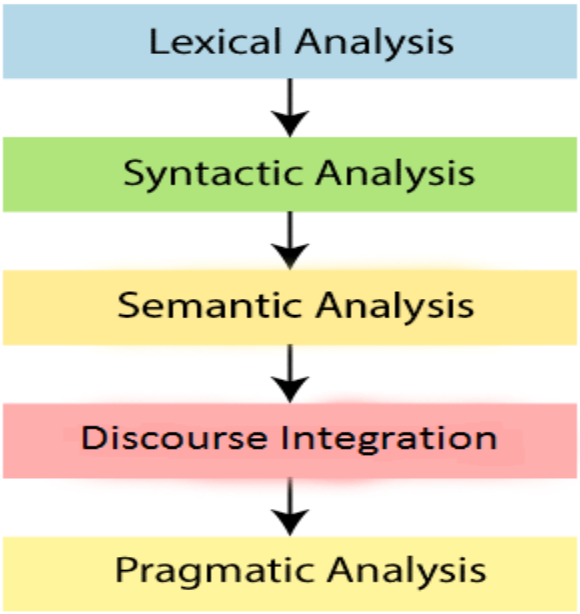
The **1960's and 1970's** witnessed the emergence of rule-primarily based systems inside the **realm of NLP**. Collaborations among linguists and computer scientists precipitated the development of structures that trusted predefined policies to analyze and understand human language.

**Long Questions/Answers**

**1. What is Natural Language Understanding? Discuss various levels of analysis under it with example.**

**Phases of NLP**

There are the following five phases of NLP:




**1. Lexical Analysis and Morphological**

The first phase of NLP is the Lexical Analysis. This phase scans the source code as a stream of characters and converts it into meaningful lexemes. It divides the whole text into paragraphs, sentences, and words.

**2. Syntactic Analysis (Parsing)**

Syntactic Analysis is used to check grammar, word arrangements, and shows the relationship among the words.

**Example:** Agra goes to the Poonam

In the real world, Agra goes to the Poonam, does not make any sense, so this sentence is rejected by the Syntactic analyzer.

**3. Semantic Analysis**

Semantic analysis is concerned with the meaning representation. It mainly focuses on the literal meaning of words, phrases, and sentences.

**4. Discourse Integration**

Discourse Integration depends upon the sentences that proceeds it and also invokes the meaning of the sentences that follow it.

**5. Pragmatic Analysis**

Pragmatic is the fifth and last phase of NLP. It helps you to discover the intended effect by applying a set of rules that characterize cooperative dialogues.

**For Example:** "Open the door" is interpreted as a request instead of an order.

NLP is difficult because Ambiguity and Uncertainty exist in the language.

**2. What do you mean by ambiguity in Natural language? Explain with a suitable example. Discuss various ways to resolve ambiguity in NLP**

There are the following three ambiguity -

- **Lexical Ambiguity**

Lexical Ambiguity exists in the presence of two or more possible meanings of the sentence within a single word.

**Example:**

Manya is looking for a **match**.

In the above example, the word match refers to that either Manya is looking for a partner or Manya is looking for a match. (Cricket or other match)

- **Syntactic Ambiguity**

Syntactic Ambiguity exists in the presence of two or more possible meanings within the sentence.

**Example:**

I saw the girl with the binocular.

In the above example, did I have the binoculars? Or did the girl have the binoculars?

- **Anaphoric Ambiguity**

A word that gets its meaning from a preceding word or phrase is called an anaphor.

Example: "**Susan** plays the piano. *She* likes music."

In this example, the word *she* is an anaphor and refers back to a preceding expression i.e., *Susan.* The linguistic element or elements to which an anaphor refers is called an antecedent. The relationship between anaphor and antecedent is termed '**anaphora**'. 'Anaphora resolution' or 'anaphor resolution' is the process of finding the correct antecedent of an anaphor

- **Pragmatic ambiguity**

Pragmatics focuses on the real-time usage of language like what the speaker wants to convey and how the listener infers it. Situational context, the individuals' mental states, the preceding dialogue, and other elements play a major role in understanding what the speaker is trying to say and how the listeners perceive it.

Example:

| ***Sentence*** | ***Direct meaning (semantic meaning)*** | ***Other meanings (pragmatic meanings)*** |
|---|---|---|
| Do you know what time is it? | Asking for the current time | Expressing anger to someone who missed the due time or something ` |
| Will you crack open the door? I am getting hot | To break | Open the door just a little |
| The chicken is ready to eat | The chicken is ready to eat its breakfast, for example. | The cooked chicken is ready to be served |

# UNIT 2

## MCQ

**1. What is the purpose of stemming in natural language processing?**

a) To convert words to their base or root form
b) To identify the parts of speech in a sentence
c) To group similar words together based on their meaning
d) To remove stop words from a sentence

**2. What is the purpose of a corpus in natural language processing?**

a) To represent a language model
b) To store and organize large amounts of text data
c) To measure the accuracy of a language model
d) To train a machine learning algorithm

**3. What is the purpose of part-of-speech tagging in natural language processing?**

a) To identify the subject and object of a sentence
b) To determine the overall sentiment of a text
c) To assign a grammatical category to each word in a sentence
d) To translate a text from one language to another

**4. What is the purpose of named entity recognition in natural language processing?**

a) To identify the tone or emotion expressed in a text
b) To determine the grammatical category of each word in a sentence
c) To identify and categorize named entities in a text, such as people, organizations, and locations
d) To generate new text based on input

**5. Which of the following is an example of a pre-processing step in natural language processing?**

a) Creating a language model

b) Identifying named entities in a text

c) Tokenization

d) Text classification

## Short Question/Answers

### 1. What is the difference between stemming and lemmatization?

Stemming and lemmatization are both techniques used in natural language processing (NLP) to reduce words to their base or root form, but they have different approaches and outcomes.

1. **Stemming**:
   - Stemming is a crude heuristic process that chops off the end of words to reduce them to their base or root form.
   - It doesn't consider the context of the word, just applies simple rules to cut off prefixes or suffixes.
   - Stemmed words may not always be actual words or may be partial words.
   - Examples: "running" -> "run", "cats" -> "cat", "happier" -> "happi"
2. **Lemmatization**:
   - Lemmatization, on the other hand, is a more sophisticated process that involves accurately identifying the base or dictionary form of a word (the lemma).
   - It takes into account the context of the word and its part of speech (POS).
   - The output of lemmatization is always a valid word.
   - Examples: "running" -> "run", "cats" -> "cat", "happier" -> "happy"

In essence, stemming is faster and simpler but may result in some inaccuracies and non-words, while lemmatization is more accurate and nuanced but also more computationally expensive. The choice between them depends on the specific requirements and constraints of the NLP task at hand.

### 2. Does the vocabulary of a corpus remain the same before and after text normalization? Why?

The vocabulary of a corpus typically changes after text normalization. Text normalization involves various processes such as converting all letters to lowercase, removing punctuation, expanding contractions, removing special characters, and applying techniques like stemming or lemmatization.

Here's why the vocabulary changes:

1. **Case normalization**: Converting all letters to lowercase eliminates the distinction between words based on their case. For example, "Apple" and "apple" would be considered the same word after case normalization. Therefore, the vocabulary would merge words that differ only in case.
2. **Punctuation removal**: Removing punctuation symbols alters the structure of the text but doesn't affect the vocabulary itself. However, it does impact the tokenization process, which may lead to different token sequences and potentially affect downstream tasks like language modeling or sentiment analysis.
3. **Contractions**: Expanding contractions like "can't" to "cannot" or "I'm" to "I am" changes the tokens themselves, thus altering the vocabulary.
4. **Stemming and lemmatization**: Applying stemming or lemmatization can further reduce the vocabulary by converting different inflections or derivations of a word to its base or dictionary form. This process groups words with similar meanings together, potentially reducing vocabulary size.

Overall, text normalization aims to standardize the text representation to make it easier for downstream tasks, but it often results in changes to the vocabulary due to the transformations applied during normalization.

## 3. What is the need for text normalization in NLP? Explain the various steps involved in text normalization.

Text normalization is essential in natural language processing (NLP) for several reasons:

1. **Standardization**: Text data often comes in various formats with inconsistencies in case usage, punctuation, and formatting. Normalizing text ensures that all text is in a standardized format, making it easier to process and analyze.
2. **Reducing vocabulary size**: Normalization techniques like stemming and lemmatization help reduce the vocabulary size by converting words to their base or root forms. This simplifies the analysis by grouping together words with similar meanings.
3. **Improving accuracy**: By standardizing text, NLP models can focus on the content rather than surface-level variations in spelling, punctuation, or formatting. This can lead to more accurate analysis and better performance on downstream tasks.
4. **Enhancing generalization**: Normalization helps NLP models generalize better across different text sources by reducing the impact of variations in spelling, grammar, and style.

The various steps involved in text normalization typically include:

1. **Lowercasing**: Converting all text to lowercase to ensure consistency and remove case distinctions.
2. **Removing punctuation**: Stripping away punctuation marks such as periods, commas, and quotation marks, which are often irrelevant for many NLP tasks.
3. **Expanding contractions**: Expanding contracted forms like "can't" to "cannot" or "I'm" to "I am" to standardize text and make it more readable.
4. **Removing special characters**: Eliminating non-alphanumeric characters or symbols that don't contribute to the semantic meaning of the text.

5. **Tokenization**: Breaking down the text into individual words or tokens. This step is crucial for further processing and analysis.
6. **Stemming or lemmatization**: Converting words to their base or root forms to reduce vocabulary size and capture the core meaning of words. Stemming chops off prefixes or suffixes, while lemmatization considers the context and morphological analysis to identify the lemma.
7. **Stopword removal**: Removing common words like "the," "and," "is," which appear frequently in text but typically don't carry much semantic meaning.
8. **Normalization specific to domain or task**: Depending on the specific requirements of the NLP task or domain, additional normalization steps may be applied. For example, in medical text processing, normalization might involve converting medical abbreviations to their full forms.

## Long Question/Answers

### 1. What is POS tagging? Discuss various approaches to perform POS tagging.

POS (Part-of-Speech) tagging is the process of assigning a grammatical category (such as noun, verb, adjective, etc.) to each word in a text corpus based on its syntactic function within a sentence. POS tagging is a fundamental task in natural language processing (NLP) and is crucial for various downstream tasks like parsing, machine translation, and information extraction.

There are several approaches to perform POS tagging:

1. **Rule-based POS tagging**:
    - Rule-based tagging relies on handcrafted linguistic rules and patterns to assign POS tags to words based on their context and surrounding words.
    - These rules are often designed by linguists or language experts and may involve dictionaries, regular expressions, and syntactic rules.
    - While rule-based tagging can be precise and accurate in specific domains, it may struggle with capturing the complexities and ambiguities of natural language.
2. **Probabilistic POS tagging**:
    - Probabilistic tagging utilizes statistical models to predict the POS tags of words based on probabilities learned from annotated training data.
    - Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs) are popular probabilistic models used for POS tagging.
    - These models estimate the probability of a sequence of POS tags given the observed words, and they often incorporate features such as word context, word morphology, and previous POS tags.
    - Probabilistic tagging tends to generalize well across different domains and languages but requires large annotated corpora for training.
3. **Deep Learning-based POS tagging**:
    - Deep learning approaches leverage neural network architectures to automatically learn feature representations for POS tagging.
    - Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Transformer-based models are commonly used for POS tagging tasks.

- These models can capture complex patterns and dependencies in language data, leading to state-of-the-art performance in POS tagging.
- Deep learning-based approaches often require large amounts of labeled data for training and are computationally expensive compared to traditional methods.

4. **Hybrid approaches**:
   - Hybrid approaches combine multiple techniques, such as rule-based, probabilistic, and deep learning methods, to achieve better performance and robustness in POS tagging.
   - For example, a hybrid model may use rule-based heuristics to handle specific cases or rare words, while relying on probabilistic or deep learning models for general tagging.
   - Hybrid approaches aim to leverage the strengths of different methods while mitigating their weaknesses.

Overall, the choice of POS tagging approach depends on factors such as the availability of labeled data, computational resources, and the desired balance between accuracy and efficiency for a particular NLP task.

## 2. How Semantic analysis is different from Pragmatic analysis?

Semantic analysis and pragmatic analysis are both important components of natural language understanding (NLU), but they focus on different aspects of language comprehension and interpretation:

1. **Semantic Analysis**:
   - Semantic analysis is concerned with the literal meaning of words, phrases, and sentences in a language.
   - It deals with the denotative or dictionary meaning of linguistic elements, ignoring context-dependent interpretations.
   - Semantic analysis aims to understand the explicit meaning of text by analyzing the relationships between words and their referents.
   - Examples of semantic analysis tasks include word sense disambiguation, named entity recognition, semantic role labeling, and semantic similarity measurement.
   - Semantic analysis is primarily concerned with extracting factual information and understanding the surface-level meaning of language.
2. **Pragmatic Analysis**:
   - Pragmatic analysis goes beyond the literal meaning of language and considers the context, speaker intentions, and social aspects of communication.
   - It deals with the implied meaning, inferred meaning, and intentions behind the words, taking into account the speaker's background knowledge, beliefs, and intentions.
   - Pragmatic analysis involves understanding implicatures, presuppositions, speech acts, conversational implicatures, and contextual dependencies.
   - Pragmatic analysis helps interpret language in real-world contexts, where meaning can be influenced by factors such as tone, gesture, cultural norms, and shared knowledge.

- Unlike semantic analysis, which focuses on the explicit meaning of language, pragmatic analysis deals with the implicit meaning and the broader communicative goals of the speaker and listener.

In summary, semantic analysis is concerned with the literal meaning of language, while pragmatic analysis involves understanding the context-dependent, implied meaning, and intentions behind the words. While semantic analysis deals with the denotative meaning of linguistic elements, pragmatic analysis considers the social, cultural, and contextual factors that shape language use and interpretation.

# UNIT 3

## MCQ

1. **What is the purpose of word embeddings in natural language processing?**

a) To represent words as numerical vectors

b) To identify the tone or emotion expressed in a text

c) To identify and categorize named entities in a text

d) To generate new text based on input

2. **What is the purpose of the Bag-of-Words (BoW) model in NLP?**

a) To represent words as vectors
b) To calculate word frequencies
c) To identify syntactic dependencies
d) To perform sentiment analysis

3. **Which of the following is an example of a word embedding technique?**

a) One-Hot Encoding
b) Bag-of-Words
c) Latent Semantic Analysis (LSA)
d) Word2Vec

4. **Which method is used to calculate the similarity between two documents based on their content?**

a) Cosine similarity
b) Euclidean distance
c) Jaccard similarity
d) Pearson correlation coefficient

5. **Which of the following is a popular pre-trained language model developed by OpenAI?**

a) BERT
b) Word2Vec
c) GloVe
d) ElMo

## Short Questions/Answers

1. **Explain TFIDF with its applications.**

Term Frequency - Inverse Document Frequency (TF-IDF) is a widely used statistical method in natural language processing and information retrieval. It measures how important a term is within a document relative to a collection of documents (i.e., relative to a corpus).

**Term Frequency:** TF of a term or word is the number of times the term appears in a document compared to the total number of words in the document.

```
tf(t,d) = count of t in d / number of words in d
```

**Inverse Document Frequency**: IDF of a term reflects the proportion of documents in the corpus that contain the term. Words unique to a small percentage of documents (e.g., technical jargon terms) receive higher importance values than words common across all documents (e.g., a, the, and).

$IDF=log$(number of the documents in the corpus / number of documents in the corpus contain the term)

The TF-IDF of a term is calculated by multiplying TF and IDF scores.

$$\text{TF-IDF}=TF*IDF$$

Translated into plain English, importance of a term is high when it occurs a lot in a given document and rarely in others. In short, commonality within a document measured by TF is balanced by rarity between documents measured by IDF. The resulting TF-IDF score reflects the importance of a term for a document in the corpus.

**Applications of TF-IDF**:

1. **Information Retrieval**: TF-IDF is widely used in search engines to rank documents based on their relevance to a query. Documents with higher TF-IDF scores for the query terms are considered more relevant and are ranked higher in search results.
2. **Document Classification**: TF-IDF can be used for text classification tasks such as sentiment analysis, spam detection, and topic classification. It helps identify important features (words) that distinguish between different classes of documents.
3. **Keyword Extraction**: TF-IDF can be used to extract keywords or key phrases from documents. Terms with high TF-IDF scores are likely to be important keywords that capture the main themes or topics of the document.
4. **Content Recommendation**: TF-IDF can be used in content recommendation systems to recommend articles, products, or other items based on their similarity to a user's

interests. Items with similar TF-IDF profiles to those of the user's preferences are recommended.

Overall, TF-IDF is a powerful technique for representing and analyzing text data, enabling various applications in information retrieval, text mining, and machine learning.

## 2. Write a short note on the following:

### a) One Hot encoding

One-hot encoding is a technique used to represent categorical data numerically, particularly in machine learning and natural language processing tasks. It converts categorical variables into binary vectors where each category is represented by a vector of zeros except for one element which is marked with a value of one, indicating the presence of that category. This encoding preserves the categorical nature of the data while making it suitable for mathematical and computational operations.

Here's how one-hot encoding works:

1. **Identify the Categories**: First, identify all unique categories in the categorical variable.
2. **Assign Indices**: Assign a unique index or identifier to each category. These indices will be used to represent the categories in the binary vectors.
3. **Create Binary Vectors**: For each data point, create a binary vector where the length of the vector is equal to the number of unique categories. Set the element corresponding to the index of the category to 1, and all other elements to 0.

### b) Bag of words

The Bag of Words (BoW) model is a simple and commonly used technique in natural language processing (NLP) for text representation. It represents text data as a sparse matrix where each row corresponds to a document and each column corresponds to a unique word in the vocabulary. The value in each cell of the matrix represents the frequency of the word in the corresponding document.

### c) N-gram

N-grams are contiguous sequences of n items (or words) from a given text. In the context of natural language processing (NLP), these items are typically words, though they can also be characters or other units of text. N-grams are used to capture the local linguistic context in a piece of text, providing information about the co-occurrence of words and their relationships within a certain window of words.

For example, let's consider the sentence "The cat sat on the mat" and generate bi-grams (2-grams):

- Bi-grams:
    - "The cat", "cat sat", "sat on", "on the", "the mat"

Similarly, for tri-grams (3-grams):

- Tri-grams:
    - "The cat sat", "cat sat on", "sat on the", "on the mat"

## 3. What is WordNet?

WordNet is a lexical database for the English language, developed at Princeton University. It organizes words into sets of synonyms called "synsets," which are interlinked by semantic relations such as hypernyms (is-a), hyponyms (has-a), antonyms, and meronyms (part-of). WordNet provides a rich network of semantic relationships between words and concepts, making it a valuable resource for natural language processing (NLP) tasks such as word sense disambiguation, information retrieval, text mining, and machine translation.

## Long Questions/Answers

### 1. Through a step-by-step process, calculate TFIDF for the given corpus and mention the word(s) having highest value.

Document 1: We are going to Mumbai

Document 2: Mumbai is a famous place.

Document 3: We are going to a famous place.

Document 4: I am famous in Mumbai.

First, let's count the term frequencies (TF) for each term in each document:

Document 1:

- "we": 1
- "are": 1
- "going": 1
- "to": 1
- "mumbai": 1

Document 2:

- "mumbai": 1
- "is": 1
- "a": 1
- "famous": 1
- "place": 1

Document 3:

- "we": 1
- "are": 1
- "going": 1
- "to": 1
- "a": 1

- "famous": 1
- "place": 1

Document 4:

- "i": 1
- "am": 1
- "famous": 1
- "in": 1
- "mumbai": 1

Next, let's calculate the document frequency (DF) for each term:

- "we": 2
- "are": 2
- "going": 2
- "to": 2
- "mumbai": 2
- "is": 1
- "a": 2
- "famous": 3
- "place": 3
- "i": 1
- "am": 1
- "in": 1

Now, we'll calculate the inverse document frequency (IDF) for each term:

IDF(t,D)=log⁡(Total number of documents in the corpus |D|Number of documents containi ng term t)\text{IDF}(t, D) = \log{\left(\frac{\text{Total number of documents in the corpus } |D|}{\text{Number of documents containing term } t}\right)}IDF(t,D)=log(Number of documents containing term tTotal number of documents in the corpus |D|)

Let's assume we have 4 documents in the corpus:

- IDF("we") = log(4/2) = 0.3010
- IDF("are") = log(4/2) = 0.3010
- IDF("going") = log(4/2) = 0.3010
- IDF("to") = log(4/2) = 0.3010
- IDF("mumbai") = log(4/2) = 0.3010
- IDF("is") = log(4/1) = 0.6021
- IDF("a") = log(4/2) = 0.3010
- IDF("famous") = log(4/3) = 0.1249
- IDF("place") = log(4/3) = 0.1249
- IDF("i") = log(4/1) = 0.6021
- IDF("am") = log(4/1) = 0.6021
- IDF("in") = log(4/1) = 0.6021

Now, let's calculate the TF-IDF for each term:

TF-IDF(t,d,D)=TF(t,d)×IDF(t,D)\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \times \text{IDF}(t, D)TF-IDF(t,d,D)=TF(t,d)×IDF(t,D)

For example, for "we" in Document 1:

- TF = 1/5
- IDF = 0.3010
- TF-IDF = (1/5) * 0.3010 = 0.0602

Now, let's calculate TF-IDF for each term in each document and identify the word(s) with the highest TF-IDF value.

## 2. Explain Skipgram and CBOW with diagram.

Skip-gram and Continuous Bag of Words (CBOW) are two popular algorithms used for training word embeddings in natural language processing (NLP). These algorithms are part of the Word2Vec framework developed by Mikolov et al. at Google.

**Continuous Bag of Words (CBOW)**: CBOW predicts the target word based on its context words. It takes a fixed-size window of context words surrounding the target word and predicts the target word using these context words. CBOW is trained by maximizing the probability of predicting the target word given its context words.

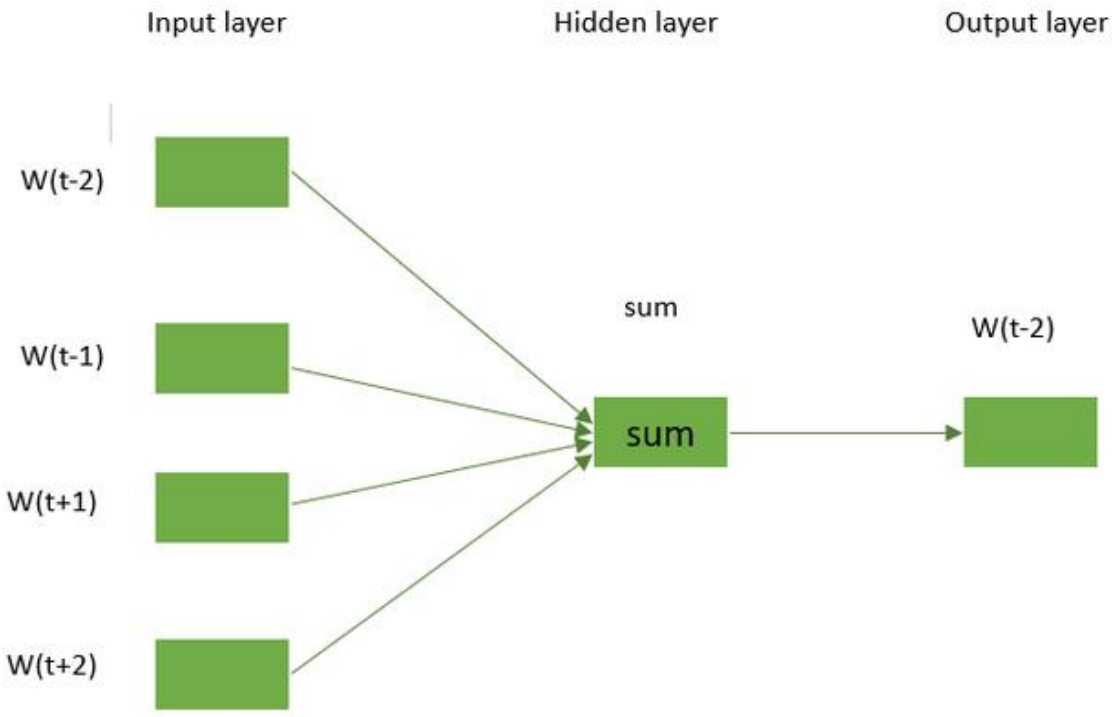



**Diagram Explanation**:

1. **Input Layer**: The input layer consists of one-hot encoded vectors representing context words. Each word in the vocabulary is represented by a unique one-hot encoded vector.

2. **Embedding Layer**: The one-hot encoded vectors are multiplied by an embedding matrix to obtain dense vector representations (word embeddings) for each context word.
3. **Average/Summation**: In CBOW, the word embeddings of the context words are averaged or summed to obtain a single vector representation.
4. **Output Layer**: The averaged or summed vector representation is passed through a softmax layer to predict the target word. The softmax layer outputs the probability distribution over all words in the vocabulary.
5. **Loss Calculation**: The loss is calculated using the predicted probability distribution and the actual target word. The goal is to minimize the loss by adjusting the weights of the embedding matrix to improve the predictions.

**Skip-gram**: Skip-gram, on the other hand, predicts context words given a target word. It takes a target word as input and predicts the context words that are likely to appear around it. Skip-gram is trained by maximizing the probability of predicting context words given the target word.

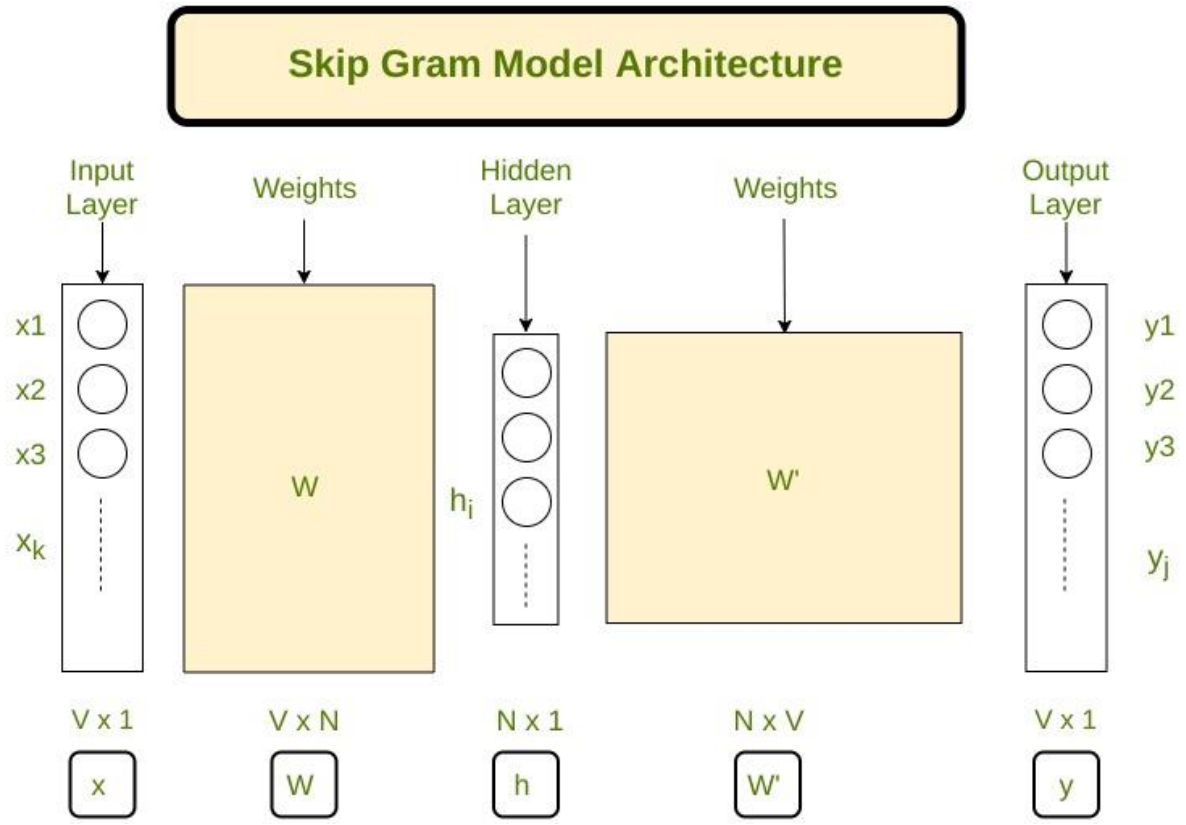



**Diagram Explanation**:

1. **Input Layer**: The input layer consists of one-hot encoded vectors representing the target word. Each word in the vocabulary is represented by a unique one-hot encoded vector.
2. **Embedding Lookup**: The one-hot encoded vector of the target word is multiplied by an embedding matrix to obtain the dense vector representation (word embedding) for the target word.

3. **Output Layer**: The word embedding of the target word is passed through a softmax layer to predict the probability distribution over context words. The softmax layer outputs the probability of each context word appearing around the target word.
4. **Loss Calculation**: The loss is calculated using the predicted probability distribution and the actual context words. The goal is to minimize the loss by adjusting the weights of the embedding matrix to improve the predictions.

In summary, CBOW predicts the target word based on its context words, while Skip-gram predicts context words given a target word. Both algorithms are used to learn distributed representations of words (word embeddings) that capture semantic and syntactic information about words in a continuous vector space.

# UNIT 4

## MCQ

1. **Which neural network architecture is commonly used for sequence-to-sequence tasks like machine translation?**

a) Long Short-Term Memory (LSTM)
b) Convolutional Neural Network (CNN)
c) Recurrent Neural Network (RNN)
d) Transformer

2. **Which algorithm is commonly used for sentiment analysis?**

a) Naive Bayes
b) K-nearest neighbors (KNN)
c) Decision trees
d) Support Vector Machines (SVM)

3. **Which of the following is not a sequence labeling task?**

a) Named entity recognition
b) Part-of-speech tagging
c) Sentiment analysis
d) Chunking

4. **Which NLP task aims to determine the correct meaning of a word based on its context?**

a) Sentiment Analysis
b) Named Entity Recognition
c) Word Sense Disambiguation
d) Part-of-Speech Tagging

5. **What does automated question answering in NLP involve?**

a) Generating questions based on a given text
b) Providing answers to questions asked by users
c) Summarizing long documents into shorter answers
d) Translating questions from one language to another

## Short Questions/Answers

1. **Why word sense disambiguation is a challenging problem in NLP?**

Word Sense Disambiguation (WSD) is a challenging problem in natural language processing (NLP) due to several reasons:

1. **Polysemy and Homonymy**: Many words in natural language have multiple meanings (polysemy) or distinct meanings that are spelled the same (homonymy). For example, "bank" can refer to a financial institution or the edge of a river. Identifying the correct

sense of a word in a given context requires distinguishing between these different meanings.

2. **Context Dependency**: The meaning of a word can vary depending on its surrounding context. For example, in the sentence "He caught a fly," "fly" could refer to an insect or the action of flying. WSD algorithms need to consider the context in which a word appears to determine its intended meaning accurately.
3. **Ambiguity Resolution**: Disambiguating between multiple senses of a word often involves resolving ambiguity present in the text. This ambiguity may arise due to syntactic structures, semantic relationships, or pragmatic factors within the language.
4. **Data Sparsity**: Annotating large corpora with sense labels for every word in every context is impractical, leading to data sparsity issues in supervised learning approaches. Limited labeled data can hinder the performance of machine learning models trained for WSD.
5. **Domain and Language Specificity**: Word meanings can vary across different domains and languages, making it challenging to develop generalizable WSD models that perform well across diverse contexts. WSD systems may need to be adapted or fine-tuned for specific domains or languages.
6. **Word Sense Granularity**: Determining the appropriate level of granularity for word senses is non-trivial. Some words may have fine-grained distinctions between senses, while others may have broader semantic categories. Balancing the granularity of sense distinctions is crucial for accurate disambiguation.
7. **Ambiguity Amplification**: Errors made by WSD systems can propagate to downstream NLP tasks, affecting the overall performance of text processing pipelines. Ambiguity amplification occurs when incorrect sense disambiguation leads to erroneous interpretations and decisions in subsequent processing stages.

**2. Write a short note on case study of:**

**a) Sentiment Analysis**

Sentiment analysis, also known as opinion mining, is a natural language processing (NLP) technique used to analyze and interpret the sentiment expressed in textual data. The goal of sentiment analysis is to determine the polarity of opinions or sentiments conveyed in a piece of text, whether it is positive, negative, or neutral. Sentiment analysis has various applications across industries, including social media monitoring, customer feedback analysis, brand reputation management, market research, and product sentiment analysis.

b) **Machine Translation**

Machine translation is the process of automatically translating text from one language to another using computational algorithms and techniques. The goal of machine translation is to produce accurate and fluent translations that preserve the meaning and intent of the original text. Machine translation has become increasingly important in our globalized world, facilitating communication across language barriers and enabling access to information in multiple languages.

**3. What is Information Retrieval? List any 5 examples of it.**

Information Retrieval (IR) is the process of retrieving relevant information from a large collection of data, typically textual documents, based on a user query or information need.

The goal of information retrieval is to provide users with the most relevant and useful documents or resources in response to their queries.

**Key Components of Information Retrieval**:

1. **Indexing**: Organizing and indexing documents to enable efficient retrieval based on query terms.
2. **Query Processing**: Analyzing and processing user queries to retrieve relevant documents from the index.
3. **Ranking**: Ranking retrieved documents based on their relevance to the query, often using algorithms like TF-IDF, BM25, or machine learning models.
4. **User Interaction**: Presenting search results to users and allowing them to refine their queries or interact with the retrieved documents.

**Examples of Information Retrieval**:

1. **Web Search Engines**: Search engines like Google, Bing, and Yahoo use information retrieval techniques to index and retrieve relevant web pages in response to user queries.
2. **Enterprise Search**: Organizations use information retrieval systems to search and retrieve documents, emails, and other resources stored in their internal databases and repositories.
3. **Digital Libraries**: Online libraries and repositories use information retrieval techniques to enable users to search and access digital collections of books, articles, and multimedia resources.
4. **E-commerce Search**: E-commerce platforms like Amazon and eBay use information retrieval to help users find products based on their search queries, filters, and preferences.
5. **Healthcare Information Systems**: Healthcare professionals use information retrieval systems to search for medical literature, patient records, diagnostic guidelines, and treatment protocols to support clinical decision-making.

**Long Questions/Answers**

1. **What do you mean by word sense disambiguation (WSD)? Discuss its various approaches.**

Word Sense Disambiguation (WSD) is a natural language processing (NLP) task that aims to determine the correct sense or meaning of a word in a given context. Many words in natural language have multiple meanings, known as senses, and identifying the intended sense of a word in a particular context is essential for accurate language understanding and processing.

**Approaches to Word Sense Disambiguation**:

1. **Knowledge-based Approaches**:
    - **Lesk Algorithm**: The Lesk algorithm compares the overlapping words and their definitions in the context of the target word with those in a knowledge base, such as WordNet. It selects the sense with the highest overlap as the correct sense.

    - **Ontology-based Methods**: These methods use structured ontologies or semantic networks to model word senses and their relationships. They rely on semantic similarity measures to determine the sense of a word based on its similarity to other words in the context.
2. **Supervised Learning Approaches**:
    - **Feature-based Models**: Supervised learning models, such as Support Vector Machines (SVM), Naive Bayes, and Decision Trees, are trained on labeled datasets containing examples of word senses in context. They extract features from the context, such as word embeddings, part-of-speech tags, and syntactic patterns, to predict the correct sense of the target word.
    - **Deep Learning Models**: Deep learning architectures, such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformer-based models, learn distributed representations of words and contexts to predict word senses. These models often leverage pretrained language models like BERT or Word2Vec for feature extraction.
3. **Unsupervised and Semi-supervised Approaches**:
    - **Clustering-based Methods**: Unsupervised methods cluster words or contexts based on their semantic similarity to identify distinct word senses. Examples include K-means clustering and spectral clustering.
    - **Graph-based Methods**: Graph-based algorithms, such as PageRank and Label Propagation, model the relationships between words and contexts as a graph and propagate information to disambiguate word senses iteratively.
4. **Hybrid Approaches**:
    - **Combination of Knowledge and Learning**: Hybrid approaches combine knowledge-based techniques with supervised or unsupervised learning methods to leverage both lexical resources and large-scale data.
    - **Ensemble Methods**: Ensemble methods combine multiple WSD algorithms or models to improve performance through voting, stacking, or averaging predictions.

Each approach to WSD has its strengths and weaknesses, and the choice of method depends on factors such as the availability of labeled data, computational resources, domain-specific requirements, and the desired balance between accuracy and efficiency. Experimentation and empirical evaluation are crucial for selecting the most suitable approach for a particular WSD task.

**2. Explain Lesk Algorithm for WSD with suitable example.**

The Lesk Algorithm is a knowledge-based approach for Word Sense Disambiguation (WSD) that compares the meanings of words in a given context with their definitions in a lexical resource, such as WordNet. It selects the sense of the target word that has the highest overlap or similarity with the surrounding context. The algorithm is based on the assumption that the correct sense of a word will have the most shared words with its context definition.

Here's how the Lesk Algorithm works:

1. **Gather Context**: Obtain the target word and its surrounding context. This context may consist of a fixed-size window of words surrounding the target word, typically a few words before and after it.

2. **Retrieve Definitions**: Retrieve the definitions or glosses of the different senses of the target word from a lexical resource like WordNet. Each sense of the word is associated with a set of words that define its meaning.
3. **Compute Overlap**: Calculate the overlap between the context words and the words in each sense's definition. The overlap can be measured using various similarity metrics, such as the Jaccard similarity coefficient or the number of shared words.
4. **Select Sense**: Choose the sense of the target word that has the highest overlap with the context words. This sense is considered the most appropriate interpretation of the word in the given context.

**Example**:

Let's consider the word "bank" in the following sentence:

"The bank was crowded with people waiting to withdraw money."

The word "bank" in this context can refer to different senses, such as a financial institution or the edge of a river.

1. **Gather Context**:
    - Target word: "bank"
    - Surrounding context: "The", "was", "crowded", "with", "people", "waiting", "to", "withdraw", "money"
2. **Retrieve Definitions**:
    - Sense 1 (financial institution): "An institution for receiving, lending, exchanging, and safeguarding money and, in some cases, issuing notes and transacting other financial business."
    - Sense 2 (edge of a river): "A long pile or heap; mass"
3. **Compute Overlap**:
    - Sense 1 overlap: 2 words ("institution", "money")
    - Sense 2 overlap: 0 words
4. **Select Sense**:
    - Sense 1 has a higher overlap with the context, so the Lesk Algorithm would select it as the correct sense of the word "bank" in this context.

In this example, the Lesk Algorithm correctly disambiguates the word "bank" based on the context, identifying the sense related to a financial institution rather than the edge of a river.

3.

4.

5. Compare different algorithms used in sentiment analysis.

6. Discuss in detail text summarization in NLP with types.

7..

8. What is QA system? Explain its types.