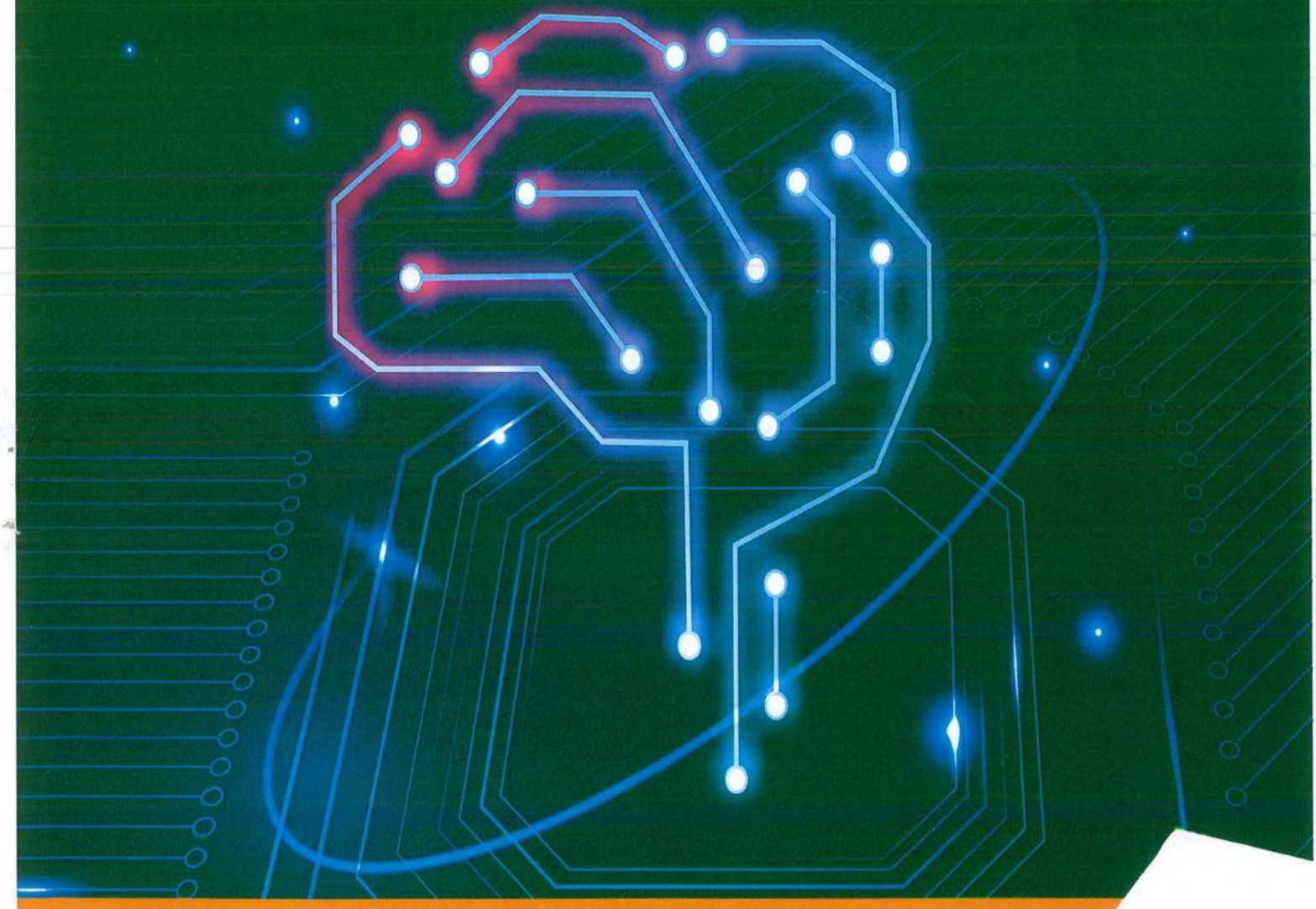
NLP Topper

Natural language Processing (University of Mumbai)

TOPPER'S SOLUTIONS

....In Search of Another Topper



NATURAL LANGUAGE PROCESSING

(SEM 8 | BE - COMPUTER)

As per Revised Syllabus w.e.f 2019-20

Mar 2022 Edition

TOPPER'S SOLUTIONS

....In Search of Another Topper

There are many existing paper solution available in market, but Topper's Solution is the one which students will always prefer if they refer... ;) Topper's Solutions is not just paper solutions, it includes many other important questions which are important from examination point of view. Topper's Solutions are the solution written by the Toppers for the students to be the upcoming Topper of the Semester.

It has been said that "**Action Speaks Louder than Words**" So Topper's Solutions Team works on same principle. Diagrammatic representation of answer is considered to be easy & quicker to understand. So our major focus is on diagrams & representation how answers should be answered in examinations.

Why Topper's Solutions:

- Point wise answers which are easy to understand & remember.
- Diagrammatic representation for better understanding.
- Additional important questions from university exams point of view.
- Covers almost every important question.
- In search of another topper.

"Education is Free.... But its Technology used & Efforts utilized which we charge"

It takes lot of efforts for searching out each & every question and transforming it into Short & Simple Language. Entire Community is working out for betterment of students, do help us.

Thanks for Purchasing & Best Luck for Exams

---- In Association with BackkBenchers Community ----

BACKK BENCHERS

"It is always the start that requires the greatest effort."

---- By James Cash Penney.

Syllabus:

Exam	TT-1	TT-2	AVG	Term Work	Oral/Practical	End of Exam	Total
Marks	20	20	20	25	25	80	150

#	Module	Details Contents	No.
1.	Introduction	History of NLP, Generic NLP system, levels of NLP, Knowledge in language processing , Ambiguity in Natural language, stages in NLP, challenges of NLP, Applications of NLP	05
2.	Word Level Analysis	Morphology analysis–survey of English Morphology, Inflectional morphology & Derivational morphology, Lemmatization, Regular expression, finite automata, finite state transducers (FST), Morphological parsing with FST, Lexicon free FST Porter stemmer. N–Grams- N-gram language model, N-gram for spelling correction	16
3.	Syntax analysis	Part-Of-Speech tagging (POS) - Tag set for English (Penn Treebank), Rule based POS tagging, Stochastic POS tagging, Issues –Multiple tags & words, Unknown words. Introduction to CFG, Sequence labeling: Hidden Markov Model (HMM), Maximum Entropy, and Conditional Random Field (CRF).	26
4.	Semantic Analysis	Lexical Semantics, Attachment for fragment of English- sentences, noun phrases, Verb phrases, prepositional phrases, Relations among lexemes & their senses – Homonymy, Polysemy, Synonymy, Hyponymy, WordNet, Robust Word Sense Disambiguation (WSD), Dictionary based approach	37
5.	Pragmatics	Discourse–reference resolution, reference phenomenon, syntactic & semantic constraints on co reference	53
6.	Applications (preferably for Indian regional languages)	Machine translation, Information retrieval, Question answers system, categorization, summarization, sentiment analysis, Named Entity Recognition.	57

Note: We have tried to cover almost every important question(s) listed in syllabus. If you feel any other question is important and it is not cover in this solution then do mail the question on Support@BackkBenchers.com or WhatsApp us on +91-9930038388 / +91-7507531198

Contact No: 7507531198

Email ID: Support@ToppersSolutions.com

Website: www.ToppersSolutions.com

Website: www.BackkBenchers.com

CHAP - 1: INTRODUCTION

Q1. What is NLP & Describe Applications of NLP.

Ans:

NLP:

1. NLP stands for **Natural Language Processing**.
2. It is part of Computer Science, Human language, and Artificial Intelligence.
3. The field of study that focuses on the interactions between human language and computers is called natural language processing.
4. It is the technology that is used by machines to understand, analyse, manipulate, and interpret human's languages.
5. Natural language recognition and natural language generation are types of NLP.
6. It helps machines to process and understand the human language so that they can automatically perform repetitive tasks.
7. It helps developers to organize knowledge for performing tasks such as translation, automatic summarization, Named Entity Recognition (NER), speech recognition, relationship extraction, and topic segmentation.
8. **Example:** Facebook uses NLP to track trending topics and popular hashtags.

ADVANTAGES OF NLP:

1. NLP helps users to ask questions about any subject and get a direct response within seconds.
2. NLP offers exact answers to the question means it does not offer unnecessary and unwanted information.
3. NLP helps computers to communicate with humans in their languages.
4. It is very time efficient.
5. Most of the companies use NLP to improve the efficiency of documentation processes, accuracy of documentation, and identify the information from large databases.

DISADVANTAGES OF NLP:

1. NLP may not show context.
2. NLP is unpredictable.
3. NLP may require more keystrokes.
4. NLP is unable to adapt to the new domain, and it has a limited function that's why NLP is built for a single and specific task only.

APPLICATIONS:

I) **Machine Translation:**

1. Machine translation is used to translate text or speech from one natural language to another natural language.

2. For performing the translation, it is important to have the knowledge of the words and phrases, grammar of two languages that are involved in translation, semantics of the languages and knowledge of the word.
3. **Example:** Google Translator

II) Question Answering:

1. Question Answering focuses on building systems that automatically answer the questions asked by humans in a natural language.
2. Question-Answering (QA) is becoming more and more popular thanks to applications such as Siri, OK Google, chat boxes and virtual assistants.
3. A QA application is a system capable of coherently answering a human request.
4. It may be used as a text-only interface or as a spoken dialog system.

III) Sentiment Analysis:

1. Sentiment Analysis is also known as **opinion mining**.
2. It is used on the web to analyse the attitude, behaviour, and emotional state of the sender.
3. The goal of sentiment analysis is to identify sentiment among several posts or even in the same post where emotion is not always explicitly expressed.
4. Companies use natural language processing applications, such as sentiment analysis, to identify opinions and sentiment online to help them understand what customers think about their products and services (i.e., "I love the new iPhone" and, a few lines later "But sometimes it doesn't work well" where the person is still talking about the iPhone) and overall indicators of their reputation.

IV) Speech Synthesis:

1. Automatic production of speech is known as **speech synthesis**.
2. It means speaking a sentence in natural language.
3. The speech synthesis system reads mails on your telephone or reads storybooks for you.
4. For generating the utterances text processing is required, so, NLP is an important component in speech synthesis system.

V) Speech Recognition:

1. Speech recognition is used for converting spoken words into text.
2. Speech Recognition is a technology that enables the computer to convert voice input data to machine readable format.
3. There are a lot of fields where speech recognition is used like, virtual assistants, adding speech-to-text, translating speech, sending emails etc.
4. It is used in search engines where the user can voice out the name of their search requirements and get the desired result, making our work easier than typing out the entire command.
5. Example: Hey Siri, Ok Google.

VI) Information Retrieval:

1. In information retrieval the relevant documents related to the user's queries are identified.

2. In information retrieval indexing, query modification, word sense disambiguation, and knowledge bases are used for enhancing the performance.
3. For example, Wordnet, and Longman Dictionary of Contemporary English (LDOCE) are some useful lexical resources for information retrieval research.

VII) Information Extraction:

1. Information extraction is one of the most important applications of NLP.
2. It is used for extracting structured information from unstructured or semi-structured machine-readable documents.
3. Information extraction system captures and output factual information contained within a document.
4. Like information retrieval system, information extraction system also response to user's information need.

VIII)Text Summarisation:

1. There is a huge amount of data available on the internet and it is very hard to go through all the data to extract a single piece of information.
2. With the help of NLP, text summarization has been made available to the users.
3. This helps in the simplification of huge amounts of data in articles, news, research papers etc.
4. This application is used in Investigative Discovery to identify patterns in writing reports, Social Media Analytics to track awareness and identify influencers, and Subject-matter expertise to classify content into meaningful topics.

IX) Recruitment:

1. In this competitive world, big and small companies are on the receiving end of thousands of resumes from different candidates.
2. It has become a tough job for the HR team to go through all the resumes and select the best candidate for one single position.
3. NLP has made the job easier by filtering through all the resumes and shortlisting the candidates by different techniques like information extraction and name entity recognition.
4. It goes through different attributes like Location, skills, education etc. and selects candidates who meet the requirements of the company closely.

X) Chatbots:

1. Chatbots are programs that are designed to assist a user 24/7 and respond appropriately and answer any query that the user might have.
2. Implementing the Chatbot is one of the important applications of NLP.
3. It is used by many companies to provide the customer's chat services.
4. **Example:** Facebook Messenger

Q2. Describe Levels of NLP.

Ans:

NLP:

1. NLP stands for **Natural Language Processing**.
2. It is part of Computer Science, Human language, and Artificial Intelligence.
3. The field of study that focuses on the interactions between human language and computers is called natural language processing.
4. It is the technology that is used by machines to understand, analyse, manipulate, and interpret human's languages.
5. Natural language recognition and natural language generation are types of NLP.
6. It helps machines to process and understand the human language so that they can automatically perform repetitive tasks.
7. It helps developers to organize knowledge for performing tasks such as translation, automatic summarization, Named Entity Recognition (NER), speech recognition, relationship extraction, and topic segmentation.
8. **Example:** Facebook uses NLP to track trending topics and popular hashtags.

LEVELS OF NLP:

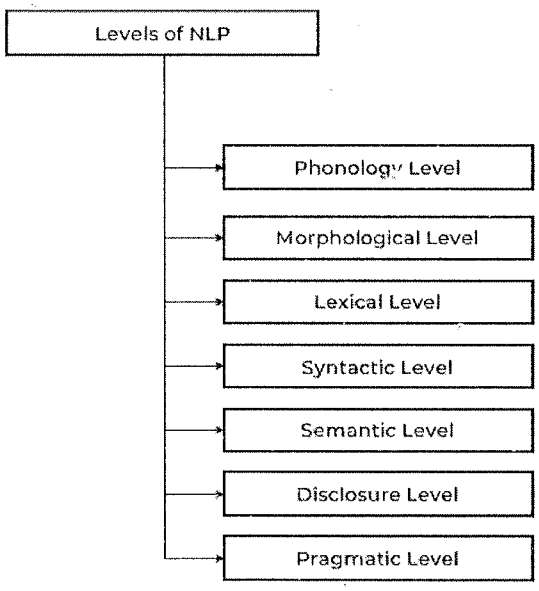


Figure 1.1: Levels of NLP

I) Phonology Level:

1. Phonology level basically deals with the **pronunciation**.
2. Phonology identifies and interprets the sounds that makeup words when the machine has to understand the spoken language.
3. It deals with physical building blocks of language sound system.
4. **Example:** Bank (finance) v/s Bank (River)
5. In hindi, aa-jayenge (will come) or aaj-ayenge (will come today).

II) Morphological Level:

1. Morphological Level deals with the smallest parts of words that convey meaning, suffixes and prefixes.
2. Morphemes means studying how the words are built from smaller meaning.
3. For example, the word 'dog' has single morpheme while the word 'dogs' hav e two morphemes 'dog' and morpheme 's' denotes singular and plural concepts.

III) Lexical Level:

1. Lexical level deals with lexical meaning of a word and part of speech.
2. It uses lexicon that is a collection of individual lexemes.
3. A lexeme is a basic unit of lexical meaning; which is an abstract unit of morphological analysis that represents the set of forms taken by a single morpheme.
4. For example, "Bank" can take the form of a noun or a verb but its part of speech and lexical meaning can only be derived in context with other words used in the sentence or phrase.

IV) Syntactic Level:

1. The part-of-speech tagging output of the lexical analysis can be used at the syntactic level of linguistic processing to group words into the phrase and clause brackets.
2. Syntactic Analysis also referred to as "parsing", allows the extraction of phrases which convey more meaning than just the individual words by themselves, such as in a noun phrase.
3. One example is differentiating between the subject and the object of the sentence, i.e., identifying who is performing the action and who is the person affected by it.
4. For example, "Jethalal thanked Babita Ji" and "Babita Ji thanked Jethalal" are sentences with different meanings from each other because in the first instance, the action of 'thanking' is done by Jethalal and affects Babita Ji, whereas, in the other one, it is done by Babita Ji and affects Jethalal.

V) Semantic Level:

1. The semantic level of linguistic processing deals with the determination of what a sentence really means by relating syntactic features and disambiguating words with multiple definitions to the given context.
2. This level deals with the meaning of words and sentences.
3. There are two approaches of semantic level:
   a. Syntax-Driven Semantic Analysis.
   b. Semantic Grammar.
4. It is a study of the meaning of words that are associated with grammatical structure.
5. For example, Tony Kakkar inputs the data from this statement we can understand that Tony Kakkar is an Agent.

VI) Disclosure Level:

1. The discourse level of linguistic processing deals with the analysis of structure and meaning of text beyond a single sentence, making connections between words and sentences.
2. It deals with the structure of different kinds of text.

3. There are two types of discourse:
   a. Anaphora Resolution.
   b. Discourse/Text Structure Recognition.
4. For example, "I love dominoes pizza because they put extra cheese", she said.
5. Here there are two entities she and dominoes, where she is in context of "I" and they is in context of "dominoes" so discourse will interpret this sentence has 2 entities (I and dominoes) and 2 anaphor (she and they)

VII) Pragmatic Level:

1. Pragmatic means **practical or logical.**
2. The pragmatic level of linguistic processing deals with the use of real-world knowledge and understanding of how this impacts the meaning of what is being communicated.
3. By analysing the contextual dimension of the documents and queries, a more detailed representation is derived.
4. Examples of Pragmatics: I heart you!
5. Semantically, "heart" refers to an organ in our body that pumps blood and keeps us alive.
6. However, pragmatically, "heart" in this sentence means "love"-hearts are commonly used as a symbol for love, and to "heart" someone has come to mean that you love someone.

Q3. Describe Ambiguity in NLP.

Ans:

NLP:

Refer Q1.

AMBIGUITY IN NLP:

1. Natural language has a very rich form and structure.
2. It is very ambiguous.
3. Ambiguity means not having well defined solution.
4. Any sentences in a language with a large enough grammar can have another interpretation.
5. Figure 1.2 shows different types of ambiguity.

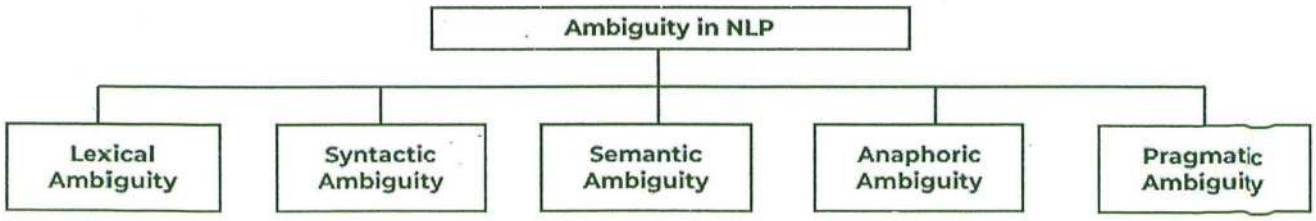


Figure 1.2: Ambiguity in NLP

I) Lexical Ambiguity:

1. Lexical is the ambiguity of a single word.

2. A word can be ambiguous with respect to its syntactic class.
3. **Example:** book, study.
4. The word silver can be used as a noun, an adjective, or a verb
   a. She bagged two silver medals.
   b. She made a silver speech.
   c. His worries had silvered his hair.
5. Lexical ambiguity can be resolved by Lexical category disambiguation i.e., parts-of-speech tagging.

II) Syntactic Ambiguity:

1. This type of ambiguity represents sentences that can be parsed in multiple syntactical forms.
2. Take the following sentence: "I heard his cell phone ring in my office".
3. The propositional phrase "in my office" can be parsed in a way that modifies the noun or on another way that modifies the verb.

III) Semantic Ambiguity:

1. This type of ambiguity is typically related to the interpretation of sentence.
2. This occurs when the meaning of the words themselves can be misinterpreted.
3. Even after the syntax and the meanings of the individual words have been resolved, there are two ways of reading the sentence.
4. Consider the example, Seema loves her mother and Sriya does too.
5. The interpretations can be Sriya loves Seema's mother or Sriya likes her own mother.

IV) Anaphoric Ambiguity

1. This kind of ambiguity arises due to the use of anaphora entities in discourse.
2. For example, the horse ran up the hill. It was very steep. It soon got tired.
3. Here, the anaphoric reference of "it" in two situations cause ambiguity.

V) Pragmatic Ambiguity:

1. Such kind of ambiguity refers to the situation where the context of a phrase gives it multiple interpretations.
2. In simple words, we can say that pragmatic ambiguity arises when the statement is not specific.
3. For example, the sentence "I like you too" can have multiple interpretations like I like you (just like you like me), I like you (just like someone else dose).
4. It is the most difficult ambiguity.

Q4. Explain the stages of NLP.

Ans:

NLP:

Refer Q1.

PHASES/STAGES OF NLP:

1. There are five stages in Natural Language Processing.
2. The figure 1.3 shows the stages of NLP.

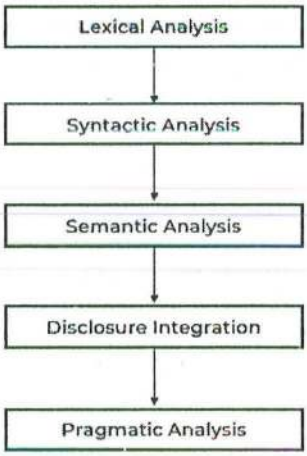


Figure 1.3: Stages of NLP

I) Lexical Analysis:

1. Lexical Analysis is the first stage in NLP.
2. It is also known as morphological analysis.
3. This phase scans the source code as a stream of characters and converts it into meaningful lexemes.
4. It divides the whole text into paragraphs, sentences, and words.
5. The most common lexicon normalization techniques are Stemming:
   a. **Stemming**: Stemming is the process of reducing derived words to their word stem, base, or root form—generally a written word form like-"ing", "ly", "es", "s", etc
   b. **Lemmatization:** Lemmatization is the process of reducing a group of words into their lemma or dictionary form. It takes into account things like POS (Parts of Speech), the meaning of the word in the sentence, the meaning of the word in the nearby sentences, etc. before reducing the word to its lemma.

II) Syntactic Analysis:

1. It is also known as **parsing**.
2. Syntactic Analysis is used to check grammar, word arrangements, and shows the relationship among the words.
3. **Example:** Agra goes to the Rutuja
4. In the real world, Agra goes to the Rutuja, does not make any sense, so this sentence is rejected by the Syntactic analyzer.

5. Dependency Grammar and Part of Speech (POS) tags are the important attributes of text syntactic.

III) Semantic Analysis:

1. Semantic analysis is **concerned with the meaning representation**.
2. It mainly focuses on the literal meaning of words.
3. Consider the sentence: "The apple ate a banana".
4. Although the sentence is syntactically correct, it doesn't make sense because apples can't eat.
5. Semantic analysis looks for meaning in the given sentence.
6. It also deals with combining words into phrases.
7. For example, "red apple" provides information regarding one object; hence we treat it as a single phrase.
8. Similarly, we can group names referring to the same category, person, object or organisation.
9. "Robert Hill" refers to the same person and not two separate names – "Robert" and "Hill".

IV) Disclosure Integration:

1. The meaning of any sentence depends upon the meaning of the sentence just before it.
2. Furthermore, it also bring about the meaning of immediately following sentence.
3. In the text, "Emiway Bantai is a bright student. He spends most of the time in the library." Here, discourse assigns "he" to refer to "Emiway Bantai".

V) Pragmatic Analysis:

1. Pragmatic is the fifth and last phase of NLP.
2. It helps you to discover the intended effect by applying a set of rules that characterize cooperative dialogues.
3. During this, what was said is re-interpreted on what it truly meant.
4. It contains deriving those aspects of language which necessitate real world knowledge.
5. For example: "Open the book" is interpreted as a request instead of an order.

Q5. Write short notes on challenges in NLP.

Ans:

NLP:

Refer Q1.

CHALLENGES IN NLP:

NLP is a powerful tool with huge benefits, but there are still a number of Natural Language Processing limitations and problems:

I) Contextual words and phrases and homonyms:

1. The same words and phrases can have different meanings according the context of a sentence and many words – especially in English – have the exact same pronunciation but totally different meanings.
2. For example:

a. I ran to the store because we ran out of milk.
b. Can I run something past you really quick?
c. The house is looking really run down.

3. These are easy for humans to understand because we read the context of the sentence and we understar d all of the different definitions.
4. And, while NLP language models may have learned all of the definitions, differentiating between them in context can present problems.
5. Homonyms – two or more words that are pronounced the same but have different definitions – can be problematic for question answering and speech-to-text applications because they aren't written in text form.
6. Usage of their and there, for example, is even a common problem for humans.

II) **Synonyms:**

1. Synonyms can lead to issues similar to contextual understanding because we use many different words to express the same idea.
2. Furthermore, some of these words may convey exactly the same meaning, while some may be levels of complexity (small, little, tiny, minute) and different people use synonyms to denote slightly different meanings within their personal vocabulary.
3. So, for building NLP systems, it's important to include all of a word's possible meanings and all possible synonyms.
4. Text analysis models may still occasionally make mistakes, but the more relevant training data they receive, the better they will be able to understand synonyms.

III) **Irony and sarcasm:**

1. Irony and sarcasm present problems for machine learning models because they generally use words and phrases that, strictly by definition, may be positive or negative, but actually connote the opposite.
2. Models can be trained with certain cues that frequently accompany ironic or sarcastic phrases, like "yeah right," "whatever," etc., and word embedding's (where words that have the same meaning have a similar representation), but it's still a tricky process.

IV) **Ambiguity:**

1. Ambiguity in NLP refers to sentences and phrases that potentially have two or more possible interpretations.
2. There are Lexical, Semantic & Syntactic ambiguity.
3. Even for humans the sentence alone is difficult to interpret without the context of surrounding text.
4. POS (part of speech) tagging is one NLP solution that can help solve the problem, somewhat.

V) **Errors in text and speech:**

1. Misspelled or misused words can create problems for text analysis.
2. Autocorrect and grammar correction applications can handle common mistakes, but don't always understand the writer's intention.

3. With spoken language, mispronunciations, different accents, stutters, etc., can be difficult for a machine to understand.
4. However, as language databases grow and smart assistants are trained by their individual users, these issues can be minimized.

VI) Colloquialisms and slang:

1. Informal phrases, expressions, idioms, and culture-specific lingo present a number of problems for NLP.
2. Because as formal language, colloquialisms may have no "dictionary definition" at all, and these expressions may even have different meanings in different geographic areas.
3. Furthermore, cultural slang is constantly morphing and expanding, so new words pop up every day.
4. For example: Bantai

VII) Domain-specific language:

1. Different businesses and industries often use very different language.
2. An NLP processing model needed for healthcare, for example, would be very different than one used to process legal documents.
3. These days, however, there are a number of analysis tools trained for specific fields, but extremely niche industries may need to build or train their own models.

VIII) Low-resource languages:

1. AI machine learning NLP applications have been largely built for the most common, widely used languages.
2. And it's downright amazing at how accurate translation systems have become.
3. However, many languages, especially those spoken by people with less access to technology often go overlooked and under processed.
4. For example, by some estimations, (depending on language vs. dialect) there are over 3,000 languages in Africa, alone.
5. There simply isn't very much data on many of these languages.

IX) Lack of research and development

1. Machine learning requires A LOT of data to function to its outer limits – billions of pieces of training data.
2. The more data NLP models are trained on, the smarter they become.
3. That said, data (and human language!) is only growing by the day, as are new machine learning techniques and custom algorithms.
4. All of the problems above will require more research and new techniques in order to improve on them.

CHAP - 2: WORD LEVEL ANALYSIS

Q1. Describe types of word formation.

Ans:

TYPES OF WORD FORMATION:

1. Word formation is the **process of creating new words**.
2. Words are the fundamental building block of language.
3. Every human language, spoken, signed or written is composed of words.
4. There are three types of word formation. i.e. Inflection, Derivation and Compounding.

I) Inflection:

1. In morphology, inflection is a process of word formation in which a word is modified to express different grammatical categories such as tense, case, voice, aspect, person, number, gender, mood and definiteness.
2. Nouns have simple inflectional morphology.
3. Examples of the inflection of noun in English are given below, here an affix marking plural.
   a. Cat (-s)
   b. Butterfly (-lies)
   c. Mouse (mice)
   d. Box (-es)
4. A possessive affix is a suffix or prefix attached to a noun to indicate its possessor.
5. Verbs have slightly more complex inflectional, but still relatively, simple inflectional morphology.
6. There are three types of verbs in English.
   a. Main Verbs – Eat, Sleep and Impeach
   b. Modal Verbs – Can will, should
   c. Primary Verbs – Be, Have, Do
7. In Regular Verbs, all the verbs have the same endings marking the same functions.
8. Regular verbs have four morphological form.
9. Just by knowing the stem we can predict the other forms.
10. **Example:**

Morphological Forms of Regular Verbs:

Stem	Talk	Urge	Cry	Tap
-s form	Talks	Urges	Cries	Taps
-ing form	Talking	Urging	Crying	Tapping

Morphological Forms of Irregular Verbs:

Stem	Eat	Think	Put
-s form	Eats	Thinks	Puts
-ing form	Eating	Thinking	Putting

II) Derivation:

1. Morphological derivation is the process of forming a new word from an existing word, often by adding a prefix or suffix, such as un- or -ness.
2. For example, unhappy and happiness derive from the root word happy.
3. It is differentiated from inflection, which is the modification of a word to form different grammatical categories without changing its core meaning: determines, determining, and determined are from the root determine.
4. Derivational morphology often involves the addition of a derivational suffix or other affix.
5. Examples of English derivational patterns and their suffixes:
   a. adjective-to-noun: -ness (slow → slowness)
   b. adjective-to-verb: -en (weak → weaken)
   c. adjective-to-adjective: -ish (red → reddish)
   d. adjective-to-adverb: -ly (personal → personally)
   e. noun-to-adjective: -al (recreation → recreational)
   f. noun-to-verb: -fy (glory → glorify)
   g. verb-to-adjective: -able (drink → drinkable)
   h. verb-to-noun (abstract): -ance (deliver → deliverance)
   i. verb-to-noun (agent): -er (write → writer)

III) Compounding:

1. Compounding words are formed when two or more lexemes combine into a single new word.
2. Compound words may be written as one word or as two words joined with a hyphen.
3. For example:
   a. noun-noun compound: note + book → notebook
   b. adjective-noun compound: blue + berry → blueberry
   c. verb-noun compound: work + room → workroom
   d. noun-verb compound: breast + feed → breastfeed
   e. verb-verb compound: stir + fry → stir-fry
   f. adjective-verb compound: high + light → highlight
   g. verb-preposition compound: break + up → breakup
   h. preposition-verb compound: out + run → outrun
   i. adjective-adjective compound: bitter + sweet → bittersweet
   j. preposition-preposition compound: in + to → into

Q2. Write short notes on Finite Automata.

Ans:

FINITE AUTOMATA:

1. An automaton having a finite number of states is called a Finite Automaton (FA) or Finite State Automata (FSA).

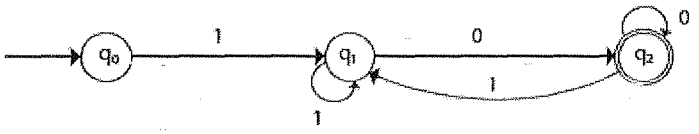
2. Finite automata are **used to recognize patterns.**
3. It takes the string of symbol as input and changes its state accordingly.
4. When the required symbol is found, then the transition happens.
5. When transition takes place, the automata can either move to the succeeding state or stay in the current state.
6. There are two states in FA: **Accept or Reject**
7. When the input string is processed successfully, and the automata reached its final state, then it will accept.
8. Mathematically, an automaton can be represented by a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where –
   a. Q is a finite set of states.
   b. Σ is a finite set of symbols, called the alphabet of the automaton. Inputs
   c. δ is the transition function.
   d. q_0 is the initial state from where any input is processed ($q_0 \in Q$).
   e. F is a set of final state/states of Q ($F \subseteq Q$).

TYPES OF FINITE STATE AUTOMATION (FSA):

I) Deterministic Finite automation (DFA):

1. It may be defined as the type of finite automation wherein, for every input symbol we can determine the state to which the machine will move.
2. It has a finite number of states that is why the machine is called **Deterministic Finite Automaton (DFA).**
3. Deterministic refers to the uniqueness of the computation.
4. In DFA, there is only one path for specific input from the current state to the next state.
5. DFA does not accept the null move, i.e., the DFA cannot change state without any input character.
6. DFA can contain multiple final states.
7. It is used in Lexical Analysis.
8. Mathematically, a DFA can be represented by a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where –
   a. Q is a finite set of states.
   b. Σ is a finite set of symbols, called the alphabet of the automaton.
   c. δ is the transition function where $\delta: Q \times \Sigma \to Q$.
   d. q_0 is the initial state from where any input is processed ($q_0 \in Q$).
   e. F is a set of final state/states of Q ($F \subseteq Q$).
9. Whereas graphically, a DFA can be represented by diagraphs called state diagrams where –
   a. The states are represented by vertices.
   b. The transitions are shown by labelled arcs.
   c. The initial state is represented by an empty incoming arc.
   d. The final state is represented by double circle.
10. **Example of DFA:**
    Design a FA with $\Sigma = \{0, 1\}$ accepts those string which starts with 1 and ends with 0.

The FA will have a start state q_0 from which only the edge with input 1 will go to the next state.



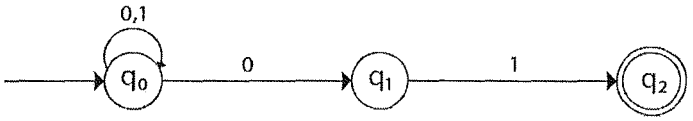
In state q_1, if we read 1, we will be in state q_1, but if we read 0 at state q_1, we will reach to state q_2 which is the final state. In state q_2, if we read either 0 or 1, we will go to q_2 state or q_1 state respectively. Note that if the input ends with 0, it will be in the final state.

II) Non-deterministic Finite Automation (NFA):

1. It may be defined as the type of finite automation where for every input symbol we cannot determine the state to which the machine will move i.e. the machine can move to any combination of the states.
2. It has a finite number of states that is why the machine is called **Non-Deterministic Finite Automation (NFA)**.
3. Every NFA is not DFA, but each NFA can be translated into DFA.
4. Mathematically, NFA can be represented by a 5-tuple (Q, Σ, δ, q0, F), where –
   a. Q is a finite set of states.
   b. Σ is a finite set of symbols, called the alphabet of the automaton.
   c. δ: is the transition function where δ: Q × Σ → 2 Q.
   d. q_0: is the initial state from where any input is processed (q0 ∈ Q).
   e. F: is a set of final state/states of Q (F ⊆ Q).
5. Whereas graphically (same as DFA), a NFA can be represented by diagraphs called state diagrams where –
   a. The states are represented by vertices.
   b. The transitions are shown by labelled arcs.
   c. The initial state is represented by an empty incoming arc.
   d. The final state is represented by double circle.
6. **Example:**

   Design an NFA with ∑ = {0, 1} accepts all string ending with 01.

   Solution: Anything either 0 or 1. Hence, NFA would be:

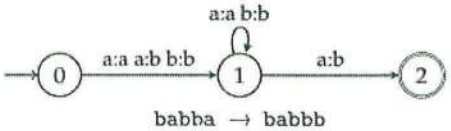


Q3. Write short notes on Finite State Transducers (FST).

Ans:

FINITE STATE TRANSDUCERS:

1. A finite-state transducer (FST) is a finite-state machine with two memory tapes: an input tape and an output tape.
2. It is a finite state machine where transitions are conditioned on a pair of symbols.
3. An FST will read a set of strings on the input tape and generates a set of relations on the output tape.
4. An FST can be thought of as a translator or relater between strings in a set.
5. In morphological parsing, an example would be inputting a string of letters into the FST, the FST would then output a string of morphemes.
6. FSTs are useful in NLP and speech recognition.
7. A Finite State Transducer (FST) is a 5-tuple T = (Q, Σ, Γ, δ, s, γ) where
   a. Q is a finite set of states
   b. Σ is a finite set of input symbols
   c. Γ is a finite set of output symbols
   d. δ: Q × Σ → Q is the transition function
   e. s ∈ Q is the start state.
   f. γ: Q → Γ * is the output function.
8. The FST is a multi-function device, and can be viewed in the following ways:
   a. **Translator:** It reads one string on one tape and outputs another string.
   b. **Recognizer:** It takes a pair of strings as two tapes and accepts/rejects based on their matching.
   c. **Generator:** It outputs a pair of strings on two tapes along with yes/no result based on whether they are matching or not.
   d. **Relater:** It computes the relation between two sets of strings available on two tapes.
9. **Example:**



babba → babbb
aba → bbb
aba → abb

CLOSURE PROPERTIES OF FINITE STATE TRANSDUCERS:

I) Union:

1. The union of two regular relations is also a regular relation.
2. If T_1 and T_2 are two FSTs, there exists a FST $T_1 \cup T_2$ such that $|T_1 \cup T_2| = |T_1| \cup |T_2|$

II) Inversion:

1. The inversion of a FST simply switches the input and output labels.
2. This means that the same FST can be used for both directions of a morphological processor.
3. If T = (Σ1; Σ2, Q, i, F, E) is a FST, there exists a FST T-1 such that | T-1 | (u) = { v ∈ Σ* | u ∈ | T (v) }

III) Composition:

1. If T_1 is a FST from I_1 to O_1 and T_2 is a FST from O_1 to O_2, then composition of T_1 and T_2 (T_1 o T_2) maps from I_1 to O_2.
2. If T_1 is a transducer from I_1 to O_1 and T_2 is a transducer from O_1 to O_2, then T_1 o T_2 maps from I_1 to O2
3. So the transducer function is: $(T_1 \text{ o } T_2)(x) = T_1(T_2(x))$

Q4. Explain N-Gram Model

Ans:

N-GRAM MODEL:

1. N-gram can be defined as the contiguous sequence of 'n' items from a given sample of text or speech.
2. The items can be letters, words, or base pairs according to the application.
3. The N-grams typically are **collected from a text or speech corpus.**
4. Consider the following example: "I love reading books about Machine Learning on BackkBenchers Community"
5. A 1-gram/unigram is a one-word sequence. For the given sentence, the unigrams would be: "I", "love", "reading", "books", "about", "Machine", "Learning", "on", "BackkBenchers", "Community".
6. A 2-gram/bigram is a two-word sequence of words, such as "I love", "love reading" or "BackkBenchers Community".
7. A 3-gram/trigram is a three-word sequence of words like "I love reading", "about Machine Learning" or "on BackkBenchers Community"
8. An N-gram language model predicts the probability of a given N-gram within any sequence of words in the language.
9. A good N-gram model can predict the next word in the sentence i.e. the value of p(w|h) – what is the probability of seeing the word w given a history of previous word h – where the history contains n-1 words.
10. Let's consider the example: "This article is on Sofia", we want to calculate what is the probability of the last word being "Sofia" given the previous words.

$$P(\text{Sofia} \mid \text{This article is on})$$

11. After generalizing the above equation can be calculated as:

$$P(w_5 \mid w_1, w_2, w_3, w_4) \text{ or } P(W)$$
$$= P(w_n \mid w_1, w_2, \ldots. w_n)$$

12. But how do we calculate it? The answer lies in the chain rule of probability:

$$P(A \mid B) = P(A, B) / P(B)$$
$$P(A, B) = P(A \mid B)\, P(B)$$

13. Now generalize the above equation:

$$P(X_1, X_2, \ldots X_n) = P(X_1)\, P(X_2 \mid X_1)\, P(X_3 \mid X_1, X_2) \ldots P(X_n \mid X_1, X_2, \ldots. X_n)$$

$$P(w_1 w_2 w_3 ..... w_n) = \pi_i P(w_i | w_1 w_2, ... w_n)$$

14. Simplifying the above formula using Markov assumptions:

$$P(w_i | w_1, w_2, ... w_{i-1}) \approx P(w_i | w_{i-k}, ... w_{i-1})$$

15. For unigram:

$$P(w_1 w_2, ... W_n) \approx \pi_i P(w_i)$$

16. For Bigram:

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

Q5. Explain N-Gram Model for Spelling Correction

Ans:

N-GRAM MODEL FOR SPELLING CORRECTION:

1. Spelling correction consist of **detecting and correcting errors**.
2. Error detection is the process of finding the misspelled word
3. Error correction is the process of suggesting correct words to a misspelled word.
4. Spelling errors are mainly phonetic, where the misspell word is pronounced in the same way as the correct word.
5. The spelling errors belong to two categories named non word errors and real world errors.
6. When an error results in the world that does not appear in a given lexicon or is not a valid orthographic word form it is known as a non-word error.
7. The real world error result in actual words of the language it occurs because of the typographical mistakes or due to spelling errors.
8. The n-gram can be used for the both non word and real world errors detection because in English alphabet certain bigram or trigram of letters never occur or rarely do so.
9. For example, the trigram 'qst' and bigram 'qd' this information can be used to handle non word error.
10. N-gram technique generally required a large corpus or dictionary as training data so that an n gram table of possible combinations of letter can be compiled.
11. N gram uses chain of custody rule as follows:

$$P(s) = P(w_1 w_2 w_3 ... w_n)$$

$$= P(w_1) P(w_2/w_1) P(w_3/w_1 w_2) w_1 P(w_3/w_1 w_2 w_3) P(w_3/w_1 w_2 w_3 ... w_{n-1})$$

$$= \pi_{i=1}^{n} P(w_i / h_i)$$

Example:

Training set:

1. The Arabian Nights
2. These are the fairy tales of the east
3. The stories of the Arabian Nights are translated in many languages.

Bi-gram Model:

P(the,<s>) = 0.67

P (are/these) = 1.0
P(tales/fairy) = 1.0
P(east/the) = 0.2
P(are/knights) = 1.0
P(many/in) = 1.0
P(languages/many) = 1.0
P(Arabian/the) = 0.4
P(the/are) = 0.5
P(of/tales) = 1.0
P (stories/the) = 0.2
P (translated/are) = 0.5
P(Knights/Arabian) = 1.0
P(fairy/the) = 0.2
P(the/of) = 1.0
P(of/stories) = 1.0
P(in/translated) = 1.0

Test Sentence(s):

The Arabian Nights are the fairy tales of the east

$$P(\text{The}/\langle s\rangle) \times P(\text{Arabian/the}) \times P(\text{Knights/Arabian}) \times P(\text{are/knights}) \times P(\text{the/are}) \times P(\text{fairy/the}) P(\text{tales/fairy}) \times P(\text{of/tales}) \times P(\text{the/of}) \times P(\text{east/the})$$

$$= 0.67 \times 0.5 \times 1.0 \times 1.0 \times 0.5 \times 0.2 \times 1.0 \times 1.0 \times 1.0 \times 0.2$$

$$= 0.0067$$

12. The n-gram model suffers from data sparseness problems.
13. The n-gram that does not occur in the training data is assigned zero probability.
14. Show the large corpus even have many zero entries in its bigram matrix.
15. The smoothing techniques are used to handle the data sparseness problem.
16. Smoothing is generally referring to the task of re-evaluating zero probability or low probability n-grams and assigning them non zero values.

Q6. Explain Lexicon Free FST Porter Stemmer

Ans:

LEXICON FREE FST PORTER STEMMER:

1. The most famous stemmer algorithm is the Porter Algorithm Like morpho-analyzers, stemmers can be seen as cascaded transducers but it has no lexicon.
2. They are used in **Informational Retrieval Applications and Search Engine.**
3. Stemming algorithms are efficient but they may introduce errors because they do not use a lexicon.
4. It is based on a series of simple cascade rules –
   a. ATIONAL → ATE (relational → relate)
   b. ING → ε (motoring → motor)
   c. SSES → SS (grasses → grass)
5. **Some errors of commission are:**
   a. Organization – Organ
   b. Doing – Doe
   c. Generalization – Generic
6. **Some errors of omission are:**
   a. European – Europe
   b. Analysis – Analyzes
   c. Noise – Noisy

PORTER ALGORITHM EXAMPLE:

For words like: falling, attaching, sing, hopping etc.

Step 1:

If the word has more than one syllab and end with 'ing'

I Remove 'ing' and apply the second step

Step 2:

If word finishes by a double consonant (except L S Z)

Transform it into a single letter

Example:

falling → fall

attaching → attach

sing → sing

hopping → hop

Q7. Explain Morphological Parsing with FST.

Ans:

MORPHOLOGICAL PARSING WITH FST:

1. Morphological parsing means breaking down words into components and building a structured representation.
2. The objective of the morphological parsing is to produce output lexicons for a single input lexicon.
3. **Example:**
   a. Cats → cat + N + PL
   b. Caught → catch + V + Past
4. The above example contains the stem of the corresponding word (lexicon) in first column, along with its morphological features like +N means word is noun, +SG means it is singular, +PL means it is plural, +V for verb.
5. There can be more than one lexical level representation for a given word.
6. Two level morphology represents a word as a correspondence between a lexical level, which represents a simple concatenation of morphemes making up a word, and the surface level, which represents the actual spelling of the final word.
7. Morphological parsing is implemented by building mapping rules that map letter sequences like cats on the surface level into morpheme and features sequences like cat +N +PL on the lexical level.
8. Figure 2.1 shows these two levels for the word cats.

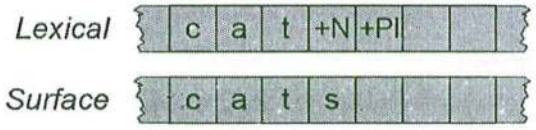


Figure 2.1

9. The automaton that we use for performing the mapping between these two levels is the finite-state transducer or FST.
10. A transducer maps between one set of symbols and another; a finite-state transducer does this via a finite automaton.
11. Thus we usually visualize an FST as a two-tape automaton which recognizes or generates pairs of strings.
12. The FST thus has a more general function than an FSA; where an FSA defines a formal language by defining a set of strings, an FST defines a relation between sets of strings.
13. This relates to another view of an FST; as a machine that reads one string and generates another, Here's a summary of this four-fold way of thinking about transducers:
    a. **FST as recognizer:** A transducer that takes a pair of strings as input and outputs accept if the string-pair is in the string-pair language, and a reject if it is not.
    b. **FST as generator:** A machine that outputs pairs of strings of the language. Thus the output is a yes or no, and a pair of output strings.
    c. **FST as translator:** A machine that reads a string and outputs another string.
    d. **FST as set relater**: A machine that computes relations between sets.

CHAP - 3: SYNTAX ANALYSIS

Q1. Explain Part of Speech (POS) Tagging.

Ans:

PART-OF-SPEECH (POS) TAGGING:

1. Part-of-speech tagging is the **process of assigning a part-of-speech or other lexical class marker to each word in a corpus.**
2. Tags are also usually applied to punctuation markers; thus tagging for natural language is the same process as tokenization for computer languages, although tags for natural languages are much more ambiguous.
3. The input to a tagging algorithm is a string of words and a specified tagset.
4. The output is a single best tag for each word.
5. For example, here are some sample sentences from the Airline Travel Information Systems (ATIS) corpus of dialogues about air-travel reservations.
6. For each we have shown a potential tagged output using the Penn Treebank tagset

VB DT NN .
Book that flight .

VBZ DT NN VB NN ?
Does that flight serve dinner ?

7. Even in these simple examples, automatically assigning a tag to each word is not trivial.
8. **For example, book is ambiguous.**
9. That is, it has more than one possible usage and part of speech.
10. It can be a verb (as in book that flight or to book the suspect) or a noun (as in hand me that book, or a book of matches).
11. Similarly, that can be a determiner (as in Does that flight serve dinner), or a complementizer (as in I thought that your flight was earlier).
12. The problem of POS-tagging is to resolve these ambiguities, choosing the proper tag for the context.
13. Most of the POS tagging falls under Rule Base POS tagging, Stochastic POS tagging and Transformation based tagging.

METHODS:

I) Rule-based POS Tagging:

1. One of the oldest techniques of tagging is rule-based POS tagging.
2. Rule-based taggers use dictionary or lexicon for getting possible tags for tagging each word.
3. If the word has more than one possible tag, then rule-based taggers use hand-written rules to identify the correct tag.
4. Disambiguation can also be performed in rule-based tagging by analysing the linguistic features of a word along with its preceding as well as following words.
5. For example, suppose if the preceding word of a word is article then word must be a noun.

6. As the name suggests, all such kind of information in rule-based POS tagging is coded in the form of rules.
7. These rules may be either
   a. Context-pattern rules
   b. Or, as Regular expression compiled into finite-state automata, intersected with lexically ambiguous sentence representation.

Two-stage architecture of Rule based POS Tagging:

1. In the first stage, it uses a dictionary to assign each word a list of potential parts-of-speech.
2. In the second stage, it uses large lists of hand-written disambiguation rules to sort down the list to a single part-of-speech for each word.

Properties of Rule-Based POS Tagging:

- These taggers are knowledge-driven taggers.
- The rules in Rule-based POS tagging are built manually.
- The information is coded in the form of rules.
- We have some limited number of rules approximately around 1000.
- Smoothing and language modelling is defined explicitly in rule-based taggers.

II) Stochastic POS Tagging:

1. Another technique of tagging is Stochastic POS Tagging.
2. The model that includes frequency or probability (statistics) can be called stochastic.
3. Any number of different approaches to the problem of part-of-speech tagging can be referred to as stochastic tagger.
4. The simplest stochastic tagger applies the following approaches for POS tagging –
   a. **Word Frequency Approach:**
      - In this approach, the stochastic taggers disambiguate the words based on the probability that a word occurs with a particular tag.
      - The tag encountered most frequently with the word in the training set is the one assigned to an ambiguous instance of that word.
      - The main issue with this approach is that it may yield inadmissible sequence of tags.
   b. **Tag Sequence Probabilities:**
      - It is another approach of stochastic tagging, where the tagger calculates the probability of a given sequence of tags occurring.
      - It is also called n-gram approach.
      - It is called so because the best tag for a given word is determined by the probability at which it occurs with the n previous tags.

Properties of Stochastic POST Tagging

- This POS tagging is based on the probability of tag occurring.
- It requires training corpus

- There would be no probability for the words that do not exist in the corpus.
- It uses different testing corpus (other than training corpus).
- It is the simplest POS tagging because it chooses most frequent tags associated with a word in training corpus.

III) Transformation-based Tagging:

1. Transformation based tagging is also called Brill tagging.
2. It is an instance of the transformation-based learning (TBL), which is a rule-based algorithm for automatic tagging of POS to the given text.
3. TBL, allows us to have linguistic knowledge in a readable form, transforms one state to another state by using transformation rules.
4. It draws the inspiration from rule-based and stochastic.
5. If we see similarity between rule-based and transformation tagger, then like rule-based, it is also based on the rules that specify what tags need to be assigned to what words.
6. On the other hand, if we see similarity between stochastic and transformation tagger then like stochastic, it is machine learning technique in which rules are automatically induced from data.

Advantages of Transformation-based Learning (TBL)

- We learn small set of simple rules and these rules are enough for tagging.
- Development as well as debugging is very easy in TBL because the learned rules are easy to understand.

Disadvantages of Transformation-based Learning (TBL)

- Transformation-based learning (TBL) does not provide tag probabilities.
- In TBL, the training time is very long especially on large corpora.

Q2. Explain CFG.

Ans:

CFG:

1. CFG stands for **Context Free Grammars**.
2. CFG's are also called **phrase-structure grammars**.
3. CFG is equivalent to **Backus-Naur Form (BNF)**.
4. CFG's are powerful enough to describe most of the structure in natural languages.
5. CFG's are restricted enough so that efficient parsers can be built.
6. CFG is a notation for describing languages and a superset of Regular grammar.
7. Context free grammar is a formal grammar which is used to generate all possible strings in a given formal language.
8. Context free grammar G can be defined by four tuples as: G= (V, T, P, S)
9. Where,
   G describes the grammar
   T describes a finite set of terminal symbols.

V describes a finite set of non-terminal symbols

P describes a set of production rules

S is the start symbol.

10. A Context-free grammar consists of a set of rules or productions, each expressing the ways the symbols of the language can be grouped together, and a lexicon of words.
11. Here are some rules for our noun phrases
    a. NP → Det Nominal
    b. NP → ProperNoun
    c. Nominal → Noun | Nominal Noun
12. Together, these describe two kinds of NPs.
    a. One that consists of a determiner followed by a nominal
    b. And another that says that proper names are NPs.
    c. The third rule illustrates two things: An explicit disjunction and A recursive definition.
13. The symbols that are used in a CFG are divided into two classes.
14. The symbols that correspond to words in the language ('The', 'BackkBenchers') are called terminal symbols.
15. The symbols that express clusters or generalizations of these are called as nonterminal symbols.
16. In each context free rule, the item to the right of the arrow (→) is an ordered list of one or more terminals and nonterminal.
17. While to the left of the arrow is a single nonterminal symbol expressing some cluster or generalization.
18. A CFG is usually thought of in two ways: as a device for generating sentences, or as a device for assigning a structure to a given sentence.

Q3. Write short notes on tagsets for English

Ans:

TAGSETS FOR ENGLISH:

1. There are a small number of popular tagsets for English, many of which evolved from the 87-tag tagset used for the Brown corpus.
2. Three of the most commonly used are the small 45-tag Penn Treebank tagset, the medium-sized 61 tag C5 tagset, and the larger 146-tag C7 tagset;
3. The Penn Treebank tagset has been applied to the Brown corpus and a number of other corpora.
4. The Penn Treebank tagset was culled from the original 87-tag tagset for the Brown corpus.
5. This reduced set leaves out information that can be recovered from the identity of the lexical item.
6. For example, the original Brown tagset and other large tagsets like C5 include a separate tag for each of the different forms of the verbs do, be, and have.
7. These were omitted from the Penn set.
8. Certain syntactic distinctions were not marked in the Penn Treebank tagset because Treebank sentences were parsed, not merely tagged, and so some syntactic information is represented in the phrase structure.

9. For example, prepositions and subordinating conjunctions were combined into the single tag IN, since the tree-structure of the sentence disambiguated them.

Tag	Description	Example	Tag	Description	Example
CC	Coordin. Conjunction	*and, but, or*	SYM	Symbol	+,%, &
CD	Cardinal number	*one, two three*	TO	"to"	*to*
DT	Determiner	*a, the*	UH	Interjection	*ah, oops*
EX	Existential 'there'	*there*	VB	Verb, base form	*eat*
FW	Foreign word	*mea culpa*	VBD	Verb, past tense	*ate*
IN	Preposition/sub-conj	*of, in, by*	VBG	Verb, gerund	*eating*
JJ	Adjective	*yellow*	VBN	Verb, past participle	*eaten*
JJR	Adj., comparative	*bigger*	VBP	Verb, non-3sg pres	*eat*
JJS	Adj., superlative	*wildest*	VBZ	Verb, 3sg pres	*eats*
LS	List item marker	*1, 2, One*	WDT	Wh-determiner	*which, that*
MD	Modal	*can, should*	WP	Wh-pronoun	*what, who*
NN	Noun, sing. or mass	*llama*	WP$	Possessive wh-	*whose*
NNS	Noun, plural	*llamas*	WRB	Wh-adverb	*how, where*
NNP	Proper noun, singular	*IBM*	$	Dollar sign	$
NNPS	Proper noun, plural	*Carolinas*	#	Pound sign	#
PDT	Predeterminer	*all, both*	"	Left quote	(' or ")
POS	Possessive ending	*'s*	"	Right quote	(' or ")
PP	Personal pronoun	*I, you, he*	(	Left parenthesis	([, (, {, <)
PP$	Possessive pronoun	*your, one's*	)	Right parenthesis	( ],), }, >)
RB	Adverb	*quickly, never*	,	Comma	,
RBR	Adverb, comparative	*faster*	.	Sentence-final punc	(. ! ?)
RBS	Adverb, superlative	*fastest*	:	Mid-sentence punc	(: ; ... – -)
RP	Particle	*up, off*			

Penn Treebank Part-of-Speech Tags (Including Punctuation)

Q4. Explain Parsing.

Ans:

Note: We have explained the below answer in detail for clear understanding. While writing in exam, Cut short it as per your understanding

PARSING:

1. Parsing in NLP is the process of determining the syntactic structure of a text by analysing its constituent words based on an underlying grammar (of the language).
2. In syntactic parsing, the parser can be viewed as searching through the space of all possible parse trees to find the correct parse tree for the sentence.
3. Consider the example "Book that flight"
4. **Grammar:**

$S \rightarrow NP\ VP$	$Det \rightarrow that \mid this \mid a$
$S \rightarrow Aux\ NP\ VP$	$Noun \rightarrow book \mid flight \mid meal \mid money$
$S \rightarrow VP$	$Verb \rightarrow book \mid include \mid prefer$
$NP \rightarrow Det\ Nominal$	$Aux \rightarrow does$
$Nominal \rightarrow Noun$	
$Nominal \rightarrow Noun\ Nominal$	$Prep \rightarrow from \mid to \mid on$
$NP \rightarrow Proper\text{-}Noun$	$Proper\text{-}Noun \rightarrow Houston \mid TWA$
$VP \rightarrow Verb$	
$VP \rightarrow Verb\ NP$	$Nominal \rightarrow Nominal\ PP$

Figure 3.1

5. **Parse Tree:**

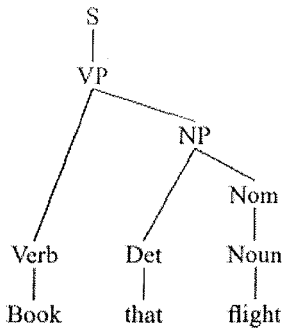


Figure 3.2

RELEVANCE OF PARSING IN NLP:

1. Parser is used to report any syntax error.
2. It helps to recover from commonly occurring error so that the processing of the remainder of program can be continued.
3. Parse tree is created with the help of a parser.
4. Parser is used to create symbol table, which plays an important role in NLP.
5. Parser is also used to produce intermediate representations (IR).

TYPES OF PARSING:

I) Top Down Parsing:

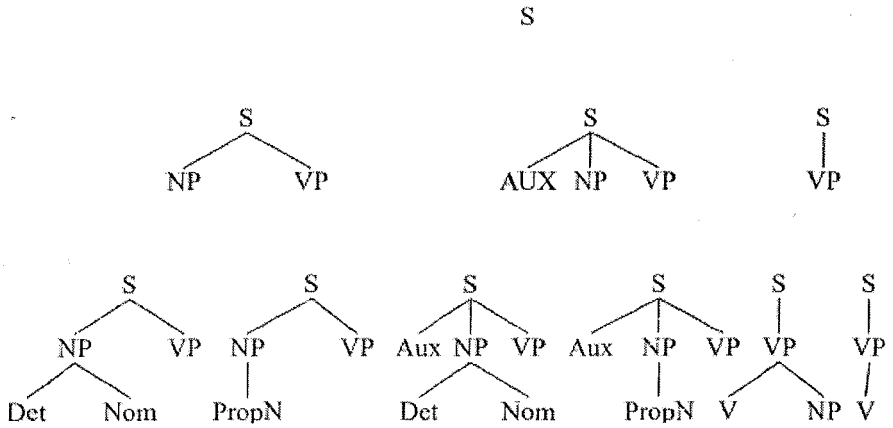


Figure 3.3: Top-down parsing example

1. A top-down parsing is **goal oriented**.
2. A top-down parser searches for a parse tree by trying to build from the root node S down to the leaves.
3. Let's consider the search space that a top-down parser explores, assuming for the moment that it builds all possible trees in parallel.
4. The algorithm starts by assuming the input can be derived by the designated start symbol S.
5. The next step is to find the tops of all trees which can start with S, by looking for all the grammar rules with S on the left-hand side.

6. In the grammar in Figure 3.1, there are three rules that expand S, so the second ply, or level, of the search space in Figure 3.3 has three partial trees.
7. We next expand the constituents in these three new trees, just as we originally expanded S.
8. The first tree tells us to expect an NP followed by a VP, the second expects an Aux followed by an NP and a VP, and the third a √P by itself.
9. To fit the search space on the page, we have shown in the third ply of Figure 3.3 only the trees resulting from the expansion of the left-most leaves of each tree.
10. At each ply of the search space we use the right-hand sides of the rules to provide new sets of expectations for the parser, which are then used to recursively generate the rest of the trees.
11. Trees are grown downward until they eventually reach the part-of-speech categories at the bottom of the tree.
12. At this point, trees whose leaves fail to match all the words in the input can be rejected, leaving behind those trees that represent successful parses.
13. In Figure 3.3, only the 5th parse tree will eventually match the input sentence Book that flight.

Problems with the Top-Down Parser:

- Only judge's grammaticality.
- Stops when it finds a single derivation.
- No semantic knowledge employed.
- No way to rank the derivations.
- Problems with left-recursive rules.
- Problems with ungrammatical sentences.

II) Bottom Up Parsing:

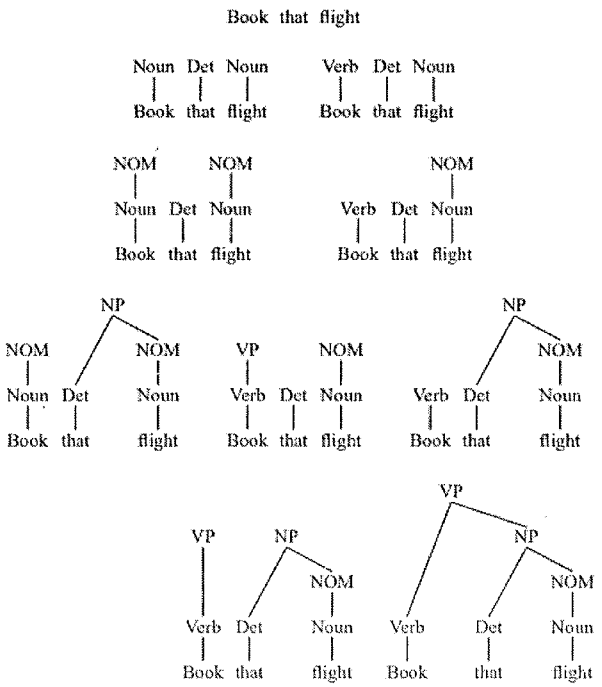


Figure 3.4: Bottom-up parsing example

1. Bottom-up parsing is **data-directed**.
2. Bottom-up parsing is the earliest known parsing algorithm, and is used in the shift-reduce parsers common for computer languages.
3. In bottom-up parsing, the parser starts with the words of the input, and tries to build trees from the words up, again by applying rules from the grammar one at a time.
4. The parse is successful if the parser succeeds in building a tree rooted in the start symbol S that covers all of the input.
5. Figure 3.4 show the bottom-up search space, beginning with the sentence Book that flight.
6. The parser begins by looking up each word (book, that, and flight) in the lexicon and building three partial trees with the part of speech for each word.
7. But the word book is ambiguous; it can be a noun or a verb.
8. Thus the parser must consider two possible sets of trees.
9. The first two plies in Figure 3.4 show this initial bifurcation of the search space.
10. Each of the trees in the second ply are then expanded.
11. In the parse on the left (the one in which book is incorrectly considered a noun), the Nominal ! Noun rule is applied to both of the Nouns (book and flight).
12. This same rule is also applied to the sole Noun (flight) on the right, producing the trees on the third ply.
13. In general, the parser extends one ply to the next by looking for places in the parse-in-progress where the right-hand-side of some rule might fit.
14. This contrasts with the earlier top-down parser, which expanded trees by applying rules when their left-hand side matched an unexpanded nonterminal.
15. Thus in the fourth ply, in the first and third parse, the sequence Det Nominal is recognized as the right-hand side of the NP ! Det Nominal rule.
16. In the fifth ply, the interpretation of book as a noun has been pruned from the search space.
17. This is because this parse cannot be continued: there is no rule in the grammar with the right-hand side Nominal NP.
18. The final ply of the search space (not shown in Figure 3.4) is the correct parse tree (see Figure 3.2).

Problems with bottom-up parsing:

- Unable to deal with empty categories: termination problem, unless rewriting empties as constituents is somehow restricted
- Inefficient when there is great lexical ambiguity.
- Conversely, it is data-directed: it attempts to parse the words that are there.
- Repeated work: anywhere there is common substructure.

Q5. Describe Sequence Labeling.

Ans:

SEQUENCE LABELING:

1. Sequence labeling is a type of **pattern recognition task.**
2. It is a typical NLP task which assigns a class or label to each token in a given input sequence.
3. In this context, a single word will be referred to as a "token".
4. These tags or labels can be used in further downstream models as features of the token, or to enhance search quality by naming spans of tokens.
5. In question answering and search tasks, we can use these spans as entities to specify our search query (e.g.,,. "Play a movie by Tom Hanks") we would like to label words such as: [Play, movie, Tom Hanks].
6. With these parts removed, we can use the verb "play" to specify the wanted action, the word "movie" to specify the intent of the action and Tom Hanks as the single subject for our search.
7. To do this, we need a way of labeling these words to later retrieve them for our query.
8. A common example of a sequence labeling task is part of speech tagging, which seeks to assign a part of speech to each word in an input sentence or document.
9. There are two forms of sequence labeling are:
   a. **Token Labeling:** Each token gets an individual Part of Speech (POS) label and
   b. **Span Labeling:** Labeling segments or groups of words that contain one tag (Named Entity Recognition, Syntactic Chunks).

Q6. Write short notes on Hidden Markov Model.

Ans:

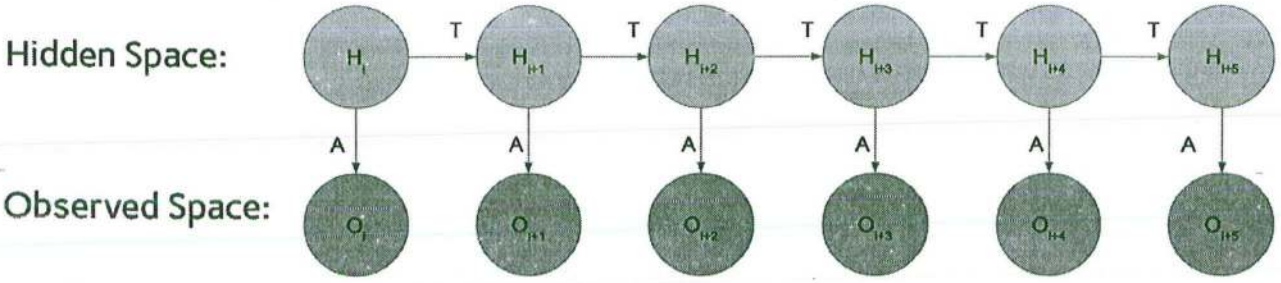
HIDDEN MARKOV MODEL (HMM):

1. Hidden Markov models (HMMs) are **sequence models.**
2. HMMs are "a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobservable (i.e. hidden) states".
3. They are designed to model the joint distribution P(H , O) , where H is the hidden state and O is the observed state.
4. For example, in the context of POS tagging, the objective would be to build an HMM to model P(word | tag) and compute the label probabilities given observations using Bayes' Rule:

$$P(H|O) = \frac{P(O|H)P(H)}{P(O)}$$

5. HMM graphs consist of a Hidden Space and Observed Space, where the hidden space consists of the labels and the observed space is the input.

6. These spaces are connected via transition matrices {T, A} to represent the probability of transitioning from one state to another following their connections.
7. Each connection represents a distribution over possible options; given our tags, this results in a large search space of the probability of all words given the tag.



8. The main idea behind HMMs is that of making observations and traveling along connections based on a probability distribution.
9. In the context of sequence tagging, there exists a changing observed state (the tag) which changes as our hidden state (tokens in the source text) also changes.

Q7. Write short notes on Conditional Random Fields (CRF).

Ans:

CONDITIONAL RANDOM FIELDS (CRF):

1. Maximum Entropy Markov Models (MEMMs) also have a well-known issue known as **label bias**.
2. The label bias problem was introduced due to MEMMs applying local normalization.
3. This often leads to the model getting stuck in local minima during decoding.
4. The local minima trap occurs because the overall model favors nodes with the least amount of transitions.
5. To solve this, Conditional Random Fields (CRFs) normalize globally and introduce an undirected graphical structure.
6. The conditional random field (CRF) is a conditional probabilistic model for sequence labeling; just as structured perceptron is built on the perceptron classifier, conditional random fields are built on the logistic regression classifier.
7. Conditional random fields (CRFs) are a probabilistic framework for labeling and segmenting sequential data, based on the conditional approach described in hidden markov model.
8. A CRF is a form of undirected graphical model that defines a single log-linear distribution over label sequences given a particular observation sequence.
9. The primary advantage of CRFs over hidden Markov models is their conditional nature, resulting in the relaxation of the independence assumptions required by HMMs in order to ensure tractable inference.
10. CRFs outperform both MEMMs and HMMs on a number of real-world sequence labeling tasks.
11. Figure 3.5 shows the graphical structure of CRF

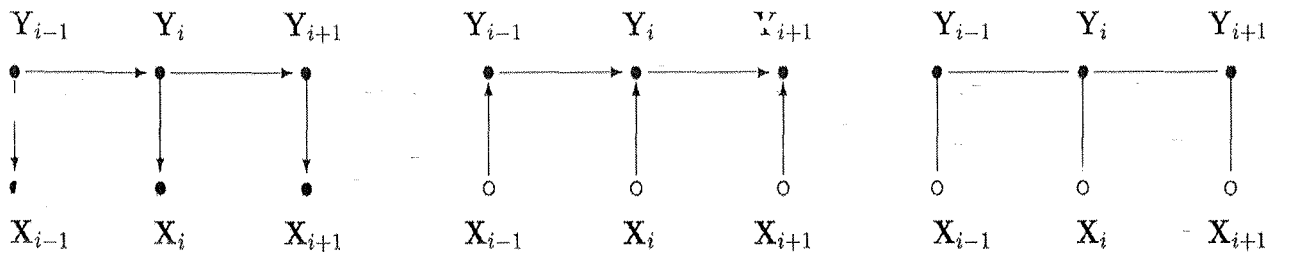


Figure . Graphical structures of simple HMMs (left), MEMMs (center), and the chain-structured case of CRFs (right) for sequences. An open circle indicates that the variable is not generated by the model.

Figure 3.5

CHAP - 4: SEMANTIC ANALYSIS

Q1. Write short notes on lexical semantics.

Ans:

LEXICAL SEMANTICS:

1. Lexical Semantics is the **study of word meaning**.
2. Lexical semantics plays a crucial role in semantic analysis, allowing computers to know relationships between words, phrasal verbs, etc.
3. Semantic analysis is the process of extracting meaning from text.
4. It permits computers to know and interpret sentences, paragraphs, or whole documents.
5. In Lexical Semantics words, sub-words, etc. are called lexical items.
6. In simple terms, lexical semantics is the relationship between lexical items, meaning of sentences and syntax of sentence.
7. The study of lexical semantics looks at:
   a. The classification and decomposition of lexical items.
   b. The differences and similarities in lexical semantic structure cross-linguistically.
   c. The relationship of lexical meaning to sentence meaning and syntax.

ELEMENTS OF LEXICAL SEMANTIC ANALYSIS:

Followings are some important elements of lexical semantic analysis:

1. **Hyponymy and Hypernymy:**
   - Hyponymy and hypernymy refers to a relationship between a general term and the more specific terms that fall under the category of the general term.
   - For example, the colors red, green, blue and yellow are hyponyms. They fall under the general term of color, which is the hypernym.
2. **Synonymy:**
   - Synonymy refers to words that are pronounced and spelled differently but contain the same meaning.
   - **Example: Happy, joyful, glad**
3. **Antonymy:**
   - Antonymy refers to words that are related by having the opposite meanings to each other.
   - There are three types of antonyms: graded antonyms, complementary antonyms, and relational antonyms.
   - **Example:**
     - dead, alive
     - long, short
4. **Homonymy:**
   - Homonymy refers to the relationship between words that are spelled or pronounced the same way but hold different meanings.
   - **Example:**

- bank (of river)
- bank (financial institution)

5. **Polysemy:**
   - Polysemy refers to a word having two or more related meanings.
   - **Example:**
     - bright (shining)
     - bright (intelligent)
6. **Meronomy:**
   - It is a logical arrangement of text and words that represent a part of or member something.
   - **Example: A segment of an apple.**

Q2. Explain the concept of attachments for a fragment of english.

Ans:

Note: For better understanding we have explained the below answer in detail. Kindly cut short it as per your understanding while attempting in exam.

SEMANTIC ATTACHMENT:

1. Semantic Attachment is the **process of making semantics of a sentence by attaching pieces of semantics to the syntax tree**.
2. It helps in creating semantic representation of a sentence.
3. There are three ways that can help to get from the syntax tree to the semantic representation. They are:
   a. Semantic Specialists.
   b. Lambda Calculus.
   c. Feature Unification.

ATTACHMENTS FOR A FRAGMENT OF ENGLISH:

I) Sentences:

1. Considering the following examples.
   a. Flight 487 serves lunch.
   b. Serve lunch.
   c. Does Flight 207 serve lunch?
   d. Which flights serve lunch?
2. The meaning representations of these examples all contain propositions concerning the serving of lunch on flights.
3. However, they differ with respect to the role that these propositions are intended to serve in the settings in which they are uttered.
4. More specifically, the first example is intended to convey factual information to a hearer, the second is a request for an action, and the last two are requests for information.

5. To capture these differences, we will introduce a set of operators that can be applied to FOPC sentences.
6. Specifically, the operators DCL, IMP, YNQ, and WHQ will be applied to the FOPC representations of declaratives, imperatives, yes-no questions, and wh-questions, respectively.
7. Flight 487 serves lunch:

$$S \rightarrow NP\ VP \qquad \{DCL(VP.sem(NP.sem))\}$$

8. Serve lunch:

$$S \rightarrow VP \qquad \{IMP(VP.sem(DummyYou))\}$$

Applying this rule to Example, results in the following representation

$$IMP(\exists e Serving(e) \wedge Server(e, DummyYou) \wedge Served(e, Lunch)$$

9. Does Flight 207 serve lunch?:

$$S \rightarrow Aux\ NP\ VP \qquad \{YNQ(VP.sem(NP.sem))\}$$

The use of this rule with for example produces the following representation

$$YNQ(\exists e Serving(e) \wedge Server(e, Flt207) \wedge Served(e, Lunch))$$

10. Which flights serve lunch?:

$$S \rightarrow WhWord\ NP\ VP \qquad \{WHQ(NP.sem.var, VP.sem(NP.sem))\}$$

The following representation is the result of applying this rule to Example

$$WHQ(x, \exists e, x\ Isa(e, Serving) \wedge Server(e, x) \wedge Served(e, Lunch) \wedge Isa(x, Flight))$$

II) <u>Compound Nominals:</u>

1. Compound nominals, also known as noun-noun sequences, consist of simple sequences of nouns, as in the following examples.
   a. Flight schedule
   b. Summer flight schedule
2. The syntactic structure of this construction can be captured by the regular expression Noun, or by the following context-free grammar rules.

$$Nominal \rightarrow Noun$$

$$Nominal \rightarrow Noun\ Nominal$$
$$\{\lambda x\ Nominal.sem(x) \wedge NN(Noun.sem, x)\}$$

3. The relation NN is used to specify that a relation holds between the modifying elements of a compound nominal and the head Noun.
4. In the examples given above, this leads to the following meaning representations

$$\lambda x Isa(x, Schedule) \wedge NN(x, Flight)$$

$$\lambda x Isa(x, Schedule) \wedge NN(x, Flight) \wedge NN(x, Summer)$$

III) Adjective Phrases:

1. English adjectives can be split into two major categories: pre-nominal and predicate.
2. These categories are exemplified by the following BERP examples.
   a. I don't mind a cheap restaurant.
   b. This restaurant is cheap.
3. For the pre-nominal case, an obvious and often incorrect proposal for the semantic attachment is illustrated in the following rules.

$$Nominal \rightarrow Adj\ Nominal$$
$$\{\lambda x\ Nominal.sem(x) \wedge Isa(x, Adj.sem)\}$$
$$Adj \rightarrow cheap \qquad \{Cheap\}$$
$$\lambda x\ Isa(x, Restaurant) \wedge Isa(x, Cheap)$$

4. This is an example of what is known as intersective semantics.
5. The best approach is to simply note the status of a specific kind of modification relation and assume that some further procedure with access to additional relevant knowledge can replace this vague relation with an appropriate representation

$$Nominal \rightarrow Adj\ Nominal$$
$$\{\lambda x\ Nominal.sem(x) \wedge AM(x, Adj.sem)\}$$

6. Applying this rule to a cheap restaurant results in the following formula

$$\exists x\ Isa(x, Restaurant) \wedge AM(x, Cheap)$$

IV) Infinitive Verb Phrases:

1. A fair number of English verbs take some form of verb phrase as one of their arguments.
2. This complicates the normal verb phrase semantic schema since these argument verb phrases interact with the other arguments of the head verb in ways that are not completely obvious
3. Consider the following example: "I told Harry to go to Maharani".
4. The meaning representation for this example should be something like the following

$$\exists e, f, x\ Isa(e, Telling) \wedge Isa(f, Going)$$
$$\wedge Teller(e, Speaker) \wedge Tellee(e, Harry) \wedge ToldThing(e, f)$$
$$\wedge Goer(f, Harry) \wedge Destination(f, x)$$

5. There are two interesting things to note about this meaning representation: the first is that it consists of two events, and the second is that one of the participants, Harry, plays a role in both of the two events.
6. The difficulty in creating this complex representation falls to the verb phrase dominating the verb tell which will something like the following as its semantic attachment.

$$\lambda x, y\ \lambda z\ \exists e\ Isa(e, Telling)$$
$$\wedge Teller(e, z) \wedge Tellee(e, x) \wedge ToldThing(e, y)$$

7. Semantically, we can interpret this subcategorization frame for Tell as providing three semantic roles: a person doing the telling, a recipient of the telling, and the proposition being conveyed.

8. **Problem:** Harry is not available when the Going event is created within the infinitive verb phrase.
9. **Solution:**

$$VP \rightarrow Verb\ NP\ VPto \qquad \{Verb.sem(NP.sem,\ VPto.sem)\}$$

$$VPto \rightarrow to\ VP \qquad \{VP.sem\}$$

$$Verb \rightarrow tell$$
$$\{\lambda x,y$$
$$\lambda z$$
$$\exists e,y.variable\ Isa(e,Telling)$$
$$\wedge Teller(e,z) \wedge Tellee(e,x)$$
$$\wedge ToldThing(e,y,variable) \wedge y(x)$$

10. In this approach, the λ-variable x plays the role of the Tellee of the telling and the argument to the semantics of the infinitive, which is now contained as a λ-expression in the variable y.
11. The expression y(x) represents a λ-reduction that inserts Harry into the Going event as the Goer.
12. The notation y: variable, is analogous to the notation used for complex-term variables, and gives us access to the event variable representing the Going event within the infinitive's meaning representation.

V) Noun Phrases:

1. A noun phrase is a group of two or more words accompanied by a noun that includes modifiers. **Example:** the, a, of them, with him
2. A noun phrase plays the role of a noun.
3. In a noun phrase, the modifiers can come before or after the noun.
4. Genitive noun phrases make use of complex determiners that consist of noun phrases with possessive markers, as in Atlanta's airport and Maharani's menu.
5. A little introspection, however, reveals that the relation between a city and its airport has little in common with a restaurant and its menu.
6. Therefore, as with compound nominals, it turns out to be best to simply state an abstract semantic relation between the various constituents.

$$NP \rightarrow ComplexDet\ Nominal$$
$$\{< \exists x Nominal.sem(x) \wedge GN(x, ComplexDet.sem) >\}$$

$$ComplexDet \rightarrow NP\ 's \qquad \{NP.sem\}$$

7. Applying these rules to Atlanta's airport results in the following complex term

$$< \exists x Isa(x, Airport) \wedge GN(x, Atlanta) >$$

VI) Prepositional Phrases:

1. At a fairly abstract level, prepositional phrases serve two distinct functions: they assert binary relations between their heads and the constituents to which they are attached, and they signal arguments to constituents that have an argument structure.

2. These two functions argue for two distinct types of prepositional phrases that differ based on their semantic attachments.
3. We will consider three places in the grammar where prepositional phrases serve these roles: modifiers of noun phrases, modifiers of verb phrases, and arguments to verb phrases.
4. **Nominal Modifier Prepositional Phrases:**

   Example: A restaurant on Pearl

$$\exists x\, Isa(x, Restaurant) \wedge On(x, Pearl)$$

$$NP \rightarrow Det\ Nominal$$

$$Nominal \rightarrow Nominal\ PP$$

$$PP \rightarrow P\ NP$$

$$P \rightarrow on \qquad \{\lambda y \lambda x\ On(x,y)\}$$

$$PP \rightarrow P\ NP \qquad \{P.sem(NP.sem)\}$$

5. **Verb Phrase Modifier Prepositional Phrases:**

   Example: ate dinner in a hurry

$$VP \rightarrow VP\ PP$$

$$\lambda x \exists e\ Isa(e, Eating) \wedge Eater(e,x) \wedge Eaten(e, Dinner)$$

$$\lambda x\ In(x, < \exists h\ Hurry(h) >)$$

$$VP \rightarrow VP\ PP \qquad \{\lambda y VP.sem(y) \wedge PP.sem(VP.sem.variable)\}$$

$$\lambda y \exists e\ Isa(e, Eating) \wedge Eater(e,y) \wedge Eaten(e, Dinner) \wedge In(e, < \exists h Hurry(h) >)$$

6. **Verb Argument Prepositional Phrases:**

   Example: I need to go from Boston to Dallas.

   In examples like this, the arguments to go are expressed as a prepositional phrase. However, the meaning representations of these phrases should consist solely of the unaltered representation of their head nouns. To handle this, argument prepositional phrases are treated in the same way that nonbranching grammatical rules are; the semantic attachment of the noun phrase is copied unchanged to the semantics of the larger phrase.

$$PP \rightarrow P\ NP \qquad \{NP.sem\}$$

Q3. Explain Homonymy and Polysemy.

Ans:

HOMONYMY:

1. Homonymy refers **to two unrelated words that look or sound the same.**
2. Two or more words become homonyms if they either sound the same (homophones), have the same spelling (homographs), or if they both homophones and homographs, but do not have related meanings.
3. Given below are some examples of homonyms:
   a. **Stalk:**
      - The main stem of a herbaceous plant
      - Pursue or approach stealthily
   b. **Bank:**
      - Financial Institution
      - Riverside

POLYSEMY:

1. Polysemy refers to words or phrases with different, but related meanings.
2. A word becomes polysemous if it can be used to express different meanings.
3. The difference between these meanings can be obvious or subtle.
4. It is sometimes difficult to determine whether a word is polysemous or not because the relations between words can be vague and unclear.
5. But, examining the origins of the words can help to decide whether a word is polysemic or homonymous.
6. The following sentences contain some examples of polysemy.
   a. He drank a glass of milk.
   b. He forgot to milk the cow.
   c. The enraged actor sued the newspaper.
   d. He read the newspaper.
   e. His cottage is near a small wood.
   f. The statue was made out of a block of wood.
   g. He fixed his hair.
   h. They fixed a date for the wedding

DIFFERENCE BETWEEN POLYSEMY AND HOMONYMY:

Polysemy	Homonymy
Polysemy is the coexistence of many possible meanings for a word or phase.	Homonymy refers to the existence of unrelated words that look or sound the same.
Polysemy has different, yet related meanings.	Homonymy has completely different meanings.
Polysemy has related word origins.	Homonymy has different origins.

Polysemous words are listed under one entry in dictionaries.	Homonymous words are listed separately in dictionaries.
Polysemous words can be understood if you know the meaning of one word.	The meaning of homonymous words cannot be guessed since the words have unrelated meanings.

Q4. Explain Meronymy.

Ans:

MERONYMY:

1. A meronym is a word that represents a constituent part or a member of something.
2. For example, Guava is a meronym of Guava-tree (sometimes written as guava < guava tree).
3. This part-to-whole relationship is named as meronymy.
4. Meronymy is not only a single relation but a bunch of different part-to-whole relationships.
5. It is also expressed in terms of first-order logic.
6. It can also be considered as a partial order.
7. In knowledge representation languages, meronymy is often represented as "part-of".
8. Figure 4.1 shows the example of meronymy

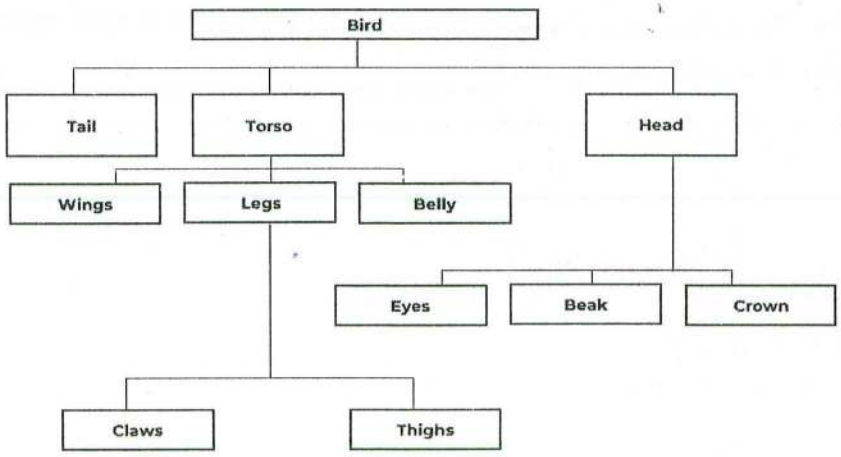


Figure 4.1: Example of Meronymy

Q5. Explain Synonymy & Antonymy.

Ans:

SYNONYMY:

1. Synonymy in semantics refers to **a word with the same or nearly the same meaning as another word.**
2. The term synonymy originates from the greek words sun and onoma which means 'with' and 'name'
3. A synonym is a word or phrase that means exactly the same as another word or phrase, in the same language.

4. In other words, synonyms are words with similar meanings.
5. For instance, words like delicious, yummy, succulent are synonyms of the adjective tasty.
6. Similarly, verbs like commence, initiate, and begin are synonyms of the verb start.
7. However, some synonyms do not have exactly the same meaning - there may be minute differences.
8. Sometimes a word can be synonymous with another in one context or usage, but not in another.
9. **Examples:**
    a. Beautiful – Gorgeous
    b. Purchase – Buy
    c. Use – Employ
    d. Rich – Wealthy
    e. Mistake – Error
    f. Big – Large
    g. Small - Little

ANTONYMY:

1. Antonyms are **words that have opposite or contrasting meanings**.
2. For example, the antonym of hot is cold; similarly, the antonym of day is night.
3. Antonyms are actually the opposite of synonyms.
4. Furthermore, there are three types of antonyms as gradable, complementary, and relational antonyms.
5. Gradable antonyms are pairs of words with opposite meanings that lie on a continuous spectrum.
6. For example, if we take age as a continuous spectrum, young and old are two ends of the spectrum.
7. Complementary antonyms are pairs of words with opposite meanings that do not lie on a continuous spectrum.
8. For example, interior: exterior, true: false, and inhale: exhale.
9. Relational antonyms are pairs of words that refer to a relationship from opposite points of view.
10. For example, doctor: patient, husband: wife, teacher: student, sister: brother.

Q5. Explain Hypernymy and Hyponymy.

Ans:

HYPERNYMY & HYPONYMY:

1. Hypernymy is the sense which is a superclass.
2. **Example:**
    a. Animal is a hypernym of dog
    b. Fruit is a hypernym of mango
    c. Vehicle is a hypernym of car
3. Hyponymy is the sense which is a subclass of another sense.
4. **Example:**
    a. Dog is a hyponym of animal.
    b. Car is a hyponym of vehicle.

c. Mango is a hyponym of fruit.

5. In simpler terms, a hyponym is in a type-of relationship with its hypernym.
6. Hypernyms and hyponyms are asymmetric.
7. Hyponymy can be tested by substituting X and Y in the sentence "X is a kind of Y" and determining if it makes sense.
8. For example, "A screwdriver is a kind of tool" makes sense, but not "A tool is a kind of screwdriver".
9. Hyponymy is a transitive relation, if X is a hyponym of Y, and Y is a hyponym of Z, then X is a hyponym of Z.
10. For example, violet is a hyponym of purple and purple is a hyponym of color; therefore, violet is a hyponym of color.
11. A word can be both a hypernym and a hyponym: for example, purple is a hyponym of color but itself is a hypernym of the broad spectrum of shades of purple between the range of crimson and violet.
12. The hierarchical structure of semantic fields can be mostly seen in hyponymy.
13. They could be observed from top to bottom, where the higher level is more general and the lower level is more specific.
14. Figure 4.2 shows the example of hypernymy and hyponymy.

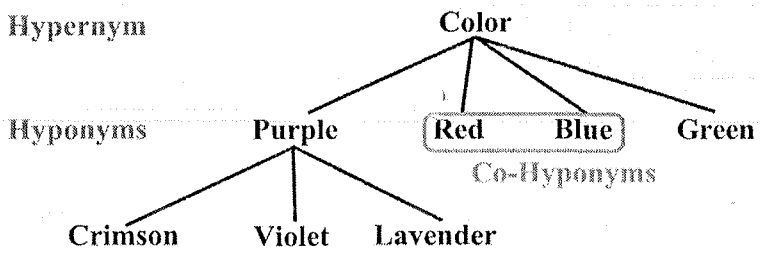


Figure 4.2: Example of Hypernymy and Hyponymy

Q6. Explain WordNet.

Ans:

WORDNET:

1. Wordnet is a **big collection of words from the English language that are related to each other and are grouped in some way**.
2. It is also called as a lexical database.
3. In other words, WordNet is a database of English words that are connected together by their semantic relationships.
4. It is like a superset dictionary with a graph structure.
5. WordNet groups nouns, verbs, adjectives, etc. which are similar and the groups are called synsets or synonyms.
6. In a wordnet a group of synsets may belong to some other synset.
7. For example, the synsets stones and cement belong to the synset "Building Materials" the synset "Stones" also belongs to another synset called "stonework".

8. In the given example, stones and cement are called hyponyms of synset building materials and also the synsets building materials and stonework are called synonyms.
9. Every member of a synset denotes the same concept but not all synset members are interchangeable in context.
10. The membership of words in multiple synsets or concepts mirrors polysemy or multiplicity of meaning.
11. There are three principles the synset construction process must adhere to:
    a. **Minimality:**
        - This principle determines on capturing those minimal set of the words in the synset which especially identifies the concept.
        - For example, (family, house) uniquely identifies a concept example: "she is from the house of the Classical Singers of Hyderabad".
    b. **Coverage:**
        - The main aim of coverage is completion of the synset, that is capturing all those words that represents the concept expressed by the synset.
        - In the synset, the words should be ordered according to their frequency in the collection.
    c. **Replaceability:**
        - Replaceability dictates that the most common words in the synset, that is words towards the beginning of the synset should be able to replace one another.
12. Figure 4.3 shows the example of Wordnet.

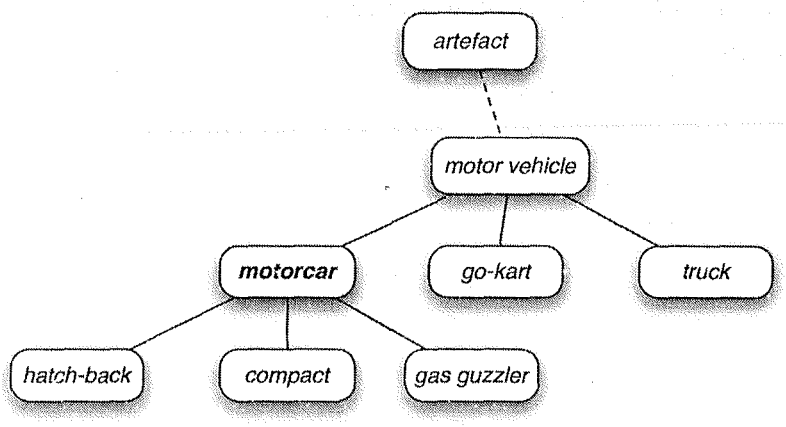


Figure 4.3: Example of Wordnet

13. In the above figure, we can see that motorcar is a motor vehicle, and it also contains subjects like compact and gas guzzler.
14. Trying to capture relationships of each word, and all the senses of each word, is extremely difficult.
15. Agreeing on the senses and boundaries of a word is also not simple.
16. These are just some of the limitations of using wordnet.

Q7. Explain WSD in detail.

Ans:

WSD:

1. WSD stands for **Word Sense Disambiguation.**
2. Words have different meanings based on the context of its usage in the sentence.
3. In human languages, words can be ambiguous too because many words can be interpreted in multiple ways depending upon the context of their occurrence.
4. Word sense disambiguation, in natural language processing (NLP), may be defined as the ability to determine which meaning of word is activated by the use of word in a particular context.
5. Lexical ambiguity, syntactic or semantic, is one of the very first problem that any NLP system faces.
6. Part-of-speech (POS) taggers with high level of accuracy can solve Word's syntactic ambiguity.
7. On the other hand, the problem of resolving semantic ambiguity is called word sense disambiguation.
8. Resolving semantic ambiguity is harder than resolving syntactic ambiguity.
9. For example, consider the two examples of the distinct sense that exist for the word "bass" –
   a. I can hear bass sound.
   b. He likes to eat grilled bass.
10. The occurrence of the word bass clearly denotes the distinct meaning.
11. In first sentence, it means frequency and in second, it means fish.
12. Hence, if it would be disambiguated by WSD then the correct meaning to the above sentences can be assigned as follows –
    a. I can hear bass/frequency sound.
    b. He likes to eat grilled bass/fish.

Approaches and Methods to Word Sense Disambiguation (WSD):

I) Dictionary-based or Knowledge-based Methods:

1. As the name suggests, for disambiguation, these methods primarily rely on dictionaries, treasures and lexical knowledge base.
2. They do not use corpora evidences for disambiguation.
3. The Lesk method is the seminal dictionary-based method introduced by Michael Lesk in 1986.
4. The Lesk definition, on which the Lesk algorithm is based is "measure overlap between sense definitions for all words in context".
5. However, in 2000, Kilgarriff and Rosensweig gave the simplified Lesk definition as "measure overlap between sense definitions of word and current context", which further means identify the correct sense for one word at a time.
6. Here the current context is the set of words in surrounding sentence or paragraph.

II) Supervised Methods:

1. For disambiguation, machine learning methods make use of sense-annotated corpora to train.

2. These methods assume that the context can provide enough evidence on its own to disambiguate the sense.
3. In these methods, the words knowledge and reasoning are deemed unnecessary.
4. The context is represented as a set of "features" of the words.
5. It includes the information about the surrounding words also.
6. Support vector machine and memory-based learning are the most successful supervised learning approaches to WSD.
7. These methods rely on substantial amount of manually sense-tagged corpora, which is very expensive to create.

III) Semi-supervised Methods:

1. Due to the lack of training corpus, most of the word sense disambiguation algorithms use semi-supervised learning methods.
2. It is because semi-supervised methods use both labelled as well as unlabeled data.
3. These methods require very small amount of annotated text and large amount of plain unannotated text.
4. The technique that is used by semi supervised methods is bootstrapping from seed data.

IV) Unsupervised Methods:

1. These methods assume that similar senses occur in similar context.
2. That is why the senses can be induced from text by clustering word occurrences by using some measure of similarity of the context.
3. This task is called word sense induction or discrimination.
4. Unsupervised methods have great potential to overcome the knowledge acquisition bottleneck due to non-dependency on manual efforts.

Difficulties in Word Sense Disambiguation (WSD):

I) Differences between dictionaries:

- The major problem of WSD is to decide the sense of the word because different senses can be very closely related.
- Even different dictionaries and thesauruses can provide different divisions of words into senses.

II) Different algorithms for different applications

- Another problem of WSD is that completely different algorithm might be needed for different applications.
- For example, in machine translation, it takes the form of target word selection; and in information retrieval, a sense inventory is not required.

III) Inter-judge variance

- Another problem of WSD is that WSD systems are generally tested by having their results on a task compared against the task of human beings.
- This is called the problem of interjudge variance.

IV) **Word-sense discreteness**

- Another difficulty in WSD is that words cannot be easily divided into discrete submeanings.

Applications of Word Sense Disambiguation (WSD):

I) **Machine Translation:**

- Machine translation or MT is the most obvious application of WSD.
- In MT, Lexical choice for the words that have distinct translations for different senses, is done by WSD.
- The senses in MT are represented as words in the target language.
- Most of the machine translation systems do not use explicit WSD module.

II) **Information Retrieval (IR):**

- Information retrieval (IR) may be defined as a software program that deals with the organization, storage, retrieval and evaluation of information from document repositories particularly textual information.
- The system basically assists users in finding the information they required but it does not explicitly return the answers of the questions.
- WSD is used to resolve the ambiguities of the queries provided to IR system.
- As like MT, current IR systems do not explicitly use WSD module and they rely on the concept that user would type enough context in the query to only retrieve relevant documents.

III) **Text Mining and Information Extraction (IE):**

- In most of the applications, WSD is necessary to do accurate analysis of text.
- For example, WSD helps intelligent gathering system to do flagging of the correct words.
- For example, medical intelligent system might need flagging of "illegal drugs" rather than "medical drugs"

IV) **Lexicography:**

- WSD and lexicography can work together in loop because modern lexicography is corpus based.
- With lexicography, WSD provides rough empirical sense groupings as well as statistically significant contextual indicators of sense.

Q8. Describe Dictionary Based Algorithm.

Ans:

DICTIONARY BASED ALGORITHM:

1. A simple approach to segment text is to scan each character one at a time from left to right and look up those characters in a dictionary.
2. If the series of characters found in the dictionary, then we have a matched word and segment that sequence as a word.
3. But this will match a shorter length word as Khmer has many of them.
4. There are several ways to better implement this approach.

I) Maximal Matching:

1. One way to avoid matching the shortest word is to find the longest sequence of characters in the dictionary instead.
2. This approach is called the longest matching algorithm or maximal matching.
3. This is a greedy algorithm that matches the longest word.
4. For example, in English, we have these series of characters: "themendinehere"
5. For the first word, we would find: the, them, theme and no longer word would match after that.
6. Now we just choose the longest which is "theme", then start again from 'n'.
7. But now we don't have any word in this series "ndine...".
8. When we can't match a word, we just mark the first character as unknown.
9. So in "ndineh...", we just took 'n' out as an unknown word and matching the next word starting with 'd'.
10. Assume the word "din" or "re" are not in our dictionary, we would get the series of words as "theme n dine here".
11. We only get one unknown word here.
12. But as you can see the longest word can make incorrect segmentation.
13. This result in overextending the first word "theme" into a part of the second word "men" making the next word unknown "n".

II) Bi-Directional Maximal Matching:

1. One way to solve this issue is to also match backward.
2. This approach is called bi-directional maximal matching as proposed by Narin Bi et al.
3. It goes from left to right (forward matching) first then from the end of the sentence go from right to left (backward matching).
4. Then choose the best result.
5. As we have seen earlier, the forward gave us incorrect segmentation.
6. But the backward would give us the correct result. Narin Bi et al. shows an accuracy of 98% for Bi-Directionaly Maximal Matching algorithm.

III) Maximum Matching:

1. Another approach to solving the greedy nature of longest matching is an algorithm called 'maximum matching'.
2. This approach would segment multiple possibilities and chose the one with fewer words in the sentence.
3. It would also prioritize fewer unknown words.
4. Using this approach, we would get the correct segmentation from the example text as shown below in figure 4.4.

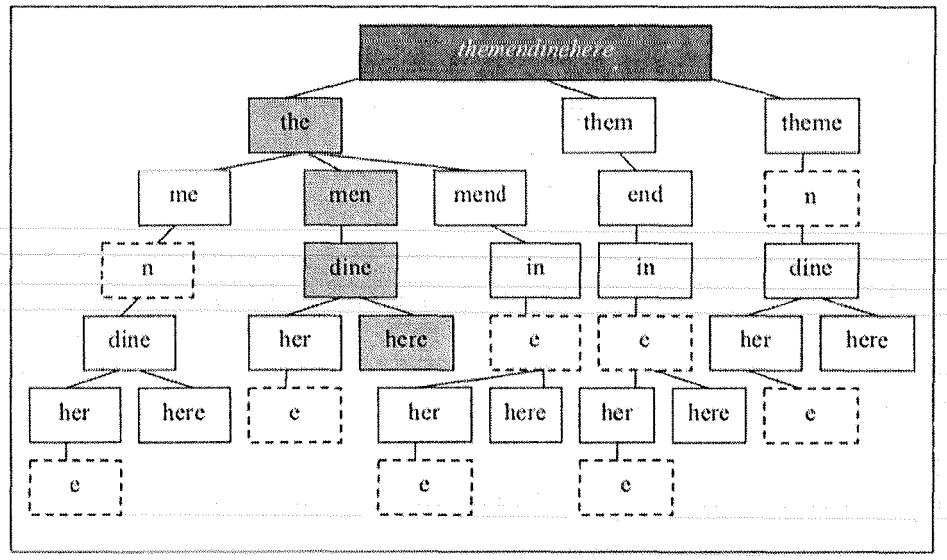


Figure 4.4: Maximum Matching

5. The first word can be "the", "them", and "theme".
6. From each of these nodes, there are multiple choices.
7. Like in "the", the next word can be "me", "men", or "mend".
8. The word "mend" would result in an incorrect word after "in".
9. Only "men" would give use "dine", then "here" as correct segmentation.
10. This approach goes through all the different combinations based on our dictionary.
11. So unknown is still a problem that we have not addressed yet.
12. In addition, it can still result in errors with the known words when the correct segmentable text prefers more words instead of a fewer number of words.

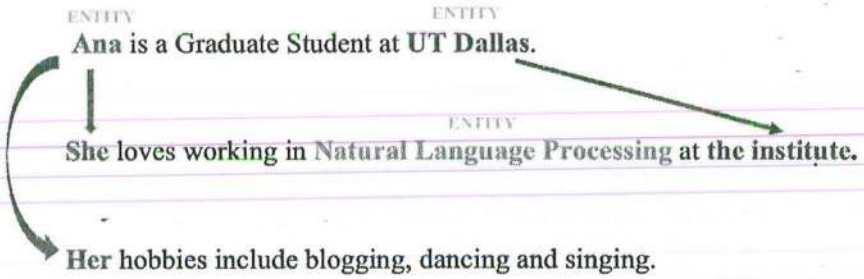
CHAP - 5: PRAGMATICS

Q1. Write short notes on Discourse Reference Resolution & Coreference Resolution.

Ans:

DISCOURSE:

1. Discourse in the context of NLP refers to a sequence of sentences occurring one after the other.
2. There will obviously be entities that are being talked about and possible references to those entities in the discourse.
3. An example of a discourse:



4. Here, "Ana", "Natural Language Processing" and "UT Dallas" are possible entities.
5. "She" and "Her" are references to the entity "Ana" and "the institute" is a reference to the entity "UT Dallas".

REFERENCE:

1. Reference, in NLP, is a linguistic process where one word in a sentence or discourse may refer to another word or entity.
2. The task of resolving such references is known as Reference Resolution.
3. In the above example, "She" and "Her" referring to the entity "Ana" and "the institute" referring to the entity "UT Dallas" are two examples of Reference Resolution.

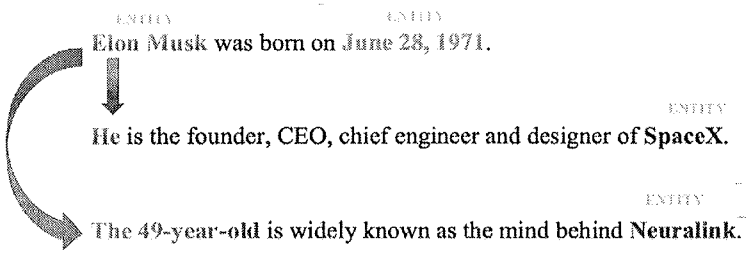
DISCOURSE - REFERENCE RESOLUTION:

1. Discourse in the context of NLP refers to a sequence of sentences occurring one after the other.
2. Reference is a linguistic process where one word in a sentence or discourse refers to another word or entity
3. The task of resolving such references is known as **Discourse - Reference Resolution**.

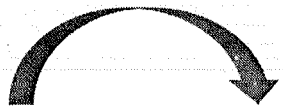
COREFERENCE RESOLUTION:

1. Coreference Resolution in particular, is the process of resolving pronouns to identify which entities are they referring to.
2. It is also a kind of Reference Resolution.
3. The entities resolved may be a person, place, organization, or event.
4. Referent is the object that is being referred to.

5. For example, "Ana" is the referent in the above example.
6. Referring expression are the mentions or linguistic expressions given in the discourse.
7. Two or more referring expressions that refer to the same discourse entity are said to corefer.
8. Now, let us look at another example to understand this better.

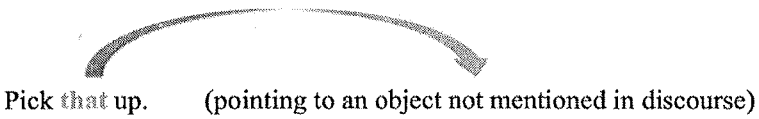


9. Referring Expressions: Elon Musk, He, The 49 year old
10. Referent: Elon Musk
11. Corefering Expressions: {Elon Musk, He}, {Elon Musk, The 49 year old}
12. References are usually of two kinds: Exaphor and Endophor.
13. Endophor refers to an entity that appears in the discourse.
14. While Exaphor refers to an entity that does not appear in the discourse.
15. Example of Endophor:



Here "She" refers to "Ana" which appears as a possible referrent that is mentioned explicitly in the discourse.

16. Example of Exaphor:



Here "that" refers to a object which appears as a possible referrent for a object that it not mentioned explicitly in the discourse

17. There are primarily two kinds of Endophors: Anaphor and Cataphor.
18. Anaphor refers to a situation wherein the referential entity or referent appears before its referencing pronoun in the discourse.
19. Example of Anaphor:

20. While, Cataphor refers to a situation wherein the entity or referent occurs later than its referencing pronoun in the discourse.
21. Example of Cataphor:

When she bought the dress, Ana didn't know it was torn

22. Here "she" occurs before its referential entity or referent "Ana" in the discourse. Thus, this is an example of cataphor.
23. The set of corefering expressions is also called a coreference chain or a cluster.

Q2. Write short notes on Types of Referring Expressions.

Ans:

TYPES OF REFERRING EXPRESSIONS:

The five types of referring expressions are described below

1. **Indefinite Noun Phrases:**
   - Such kind of reference represents the entities that are new to the hearer into the discourse context.
   - For example – in the sentence Ram had gone around one day to bring him some food – some is an indefinite reference.

2. **Definite Noun Phrases:**
   - Opposite to above, such kind of reference represents the entities that are not new or identifiable to the hearer into the discourse context.
   - For example, in the sentence - I used to read The Times of India – The Times of India is a definite reference.

3. **Pronouns:**
   - It is a form of definite reference.
   - For example, Ram laughed as loud as he could. The word he represents pronoun referring expression.

4. **Demonstratives:**
   - These demonstrate and behave differently than simple definite pronouns.
   - For example, this and that are demonstrative pronouns.

5. **Names:**
   - It is the simplest type of referring expression.
   - It can be the name of a person, organization and location also.
   - For example, in the above examples, Ram is the name-refereeing expression.

Q3. Write short notes on Syntactic & Semantic Constraints on co reference.

Ans:

SYNTACTIC & SEMANTIC CONSTRAINTS ON CO REFERENCE:

1. Reference relations may also be constrained by the syntactic relationships between a referential expression and a possible antecedent noun phrase when both occur in the same sentence.
2. For instance, the pronouns in all of the following sentences are subject to the constraints indicated in brackets.
   a. John bought himself a new Acura. [himself=John]
   b. John bought him a new Acura. [him≠John]
   c. John said that Bill bought him a new Acura. [him≠Bill]
   d. John said that Bill bought himself a new Acura. [himself=Bill]
   e. He said that he bought John a new Acura. [He≠John;he≠John]
   f. John wanted a new car. Bill bought him a new Acura. [him=John]
   g. John wanted a new car. He bought him a new Acura. [he=John; him≠John]
3. English pronouns such as himself, herself, and themselves are called reflexives.
4. Oversimplifying the situation considerably, a reflexive corefers with the subject of the most immediate clause that contains it (example: a), whereas a nonreflexive cannot corefer with this subject (example: b).
5. That this rule applies only for the subject of the most immediate clause is shown by examples (c) and (d), in which the opposite reference pattern is manifest between the pronoun and the subject of the higher sentence.
6. On the other hand, a full noun phrase like John cannot corefer with the subject of the most immediate clause nor with a higher-level subject (example: e).
7. Whereas these syntactic constraints apply to a referring expression and a particular potential antecedent noun phrase, these constraints actually prohibit coreference between the two regardless of any other available antecedents that denote the same entity.
8. For instance, normally a nonreflexive pronoun like him can corefer with the subject of the previous sentence as it does in example (f), but it cannot in example (g) because of the existence of the coreferential pronoun he in the second clause.
9. The rules given above oversimplify the situation in a number of ways, and there are many cases that they do not cover.
10. Indeed, upon further inspection the facts actually get quite complicated.
11. In fact, it is unlikely that all of the data can be explained using only syntactic relations.
12. For instance, the reflexive himself and the nonreflexive him in sentences (example: h) and (example: i) respectively can both refer to the subject John, even though they occur in identical syntactic configurations.
    h. John set the pamphlets about Acuras next to himself. [himself=John]
    i. John set the pamphlets about Acuras next to him. [him=John]

CHAP - 6: APPLICATIONS

Q1. Write short notes on machine translation in NLP and explain the different types of machine translations

Ans:

MACHINE TRANSLATION:

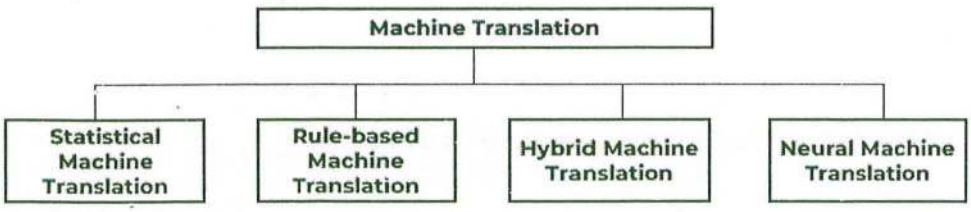
1. Machine Translation is also known as **robotized interpretation or automated translation.**
2. Machine Translation or MT is simply a procedure when a computer software translates text from one language to another without human contribution.
3. At its fundamental level, machine translation performs a straightforward replacement of atomic words in a single characteristic language for words in another.
4. Using corpus methods, more complicated translations can be conducted, taking into account better treatment of contrasts in phonetic typology, express acknowledgement, and translations of idioms, just as the seclusion of oddities.
5. In simple language, we can say that machine translation works by using computer software to translate the text from one source language to another target language.
6. Thus, Machine Translation (MT) is the task of automatically converting one natural language into another, preserving the meaning of the input text, and producing fluent text in the output language.

Challenges of Machine Translation:

1. The large variety of languages, alphabets and grammars.
2. The task to translate a sequence to a sequence is harder for a computer than working with number only.
3. There is no one correct answer (example: translating from a language without gender dependent pronouns, he and she can be the same)

TYPES OF MACHINE TRANSLATIONS:

There are four types of machine translation:



I) Statistical Machine Translation (SMT):

- It works by alluding to statistical models that depend on the investigation of huge volumes of bilingual content.
- It aims to decide the correspondence between a word from the source language and a word from the objective language.

- A genuine example of this is Google Translate.
- Presently, SMT is extraordinary for basic translation, however its most noteworthy disadvantage is that it doesn't factor in context, which implies translation can regularly be wrong. In other words, don't expect great quality translation.
- There are several types of statistical-based machine translation models which are:
  - Hierarchical phrase-based translation.
  - Syntax-based translation.
  - Phrase-based translation.
  - Word-based translation.

II) Rule-based Machine Translation (RBMT):

- RBMT basically translates the basics of grammatical rules.
- It directs a grammatical examination of the source language and the target language to create the translated sentence.
- But, RBMT requires extensive proof reading and its heavy dependence on lexicons means that efficiency is achieved after a long period of time.

III) Hybrid Machine Translation (HMT):

- HMT, as the term demonstrates, is a mix of RBMT and SMT.
- It uses a translation memory, making it unquestionably more successful regarding quality.
- However, even HMT has a lot of downsides, the biggest of which is the requirement for enormous editing, and human translators will be required.
- There are several approaches to HMT like multi-engine, statistical rule generation, multi-pass, and confidence-based.

IV) Neural Machine Translation (NMT):

- NMT is a type of machine translation that relies upon neural network models (based on the human brain) to build statistical models with the end goal of translation.
- The essential advantage of NMT is that it gives a solitary system that can be prepared to unravel the source and target text.
- Subsequently, it doesn't rely upon specific systems that are regular to other machine translation systems, particularly SMT.

Q2. Difference between Rule Based MT vs Statistical MT

Ans:

Table 6.1: Difference between Rule Based MT vs Statistical MT

Rule Based MT	Statistical MT
Consistency between versions	Inconsistency between versions
Knows grammatical rules	Does not know grammar

High performance and robustness	High CPU and disk space requirements
Out-of-domain translation quality	Poor out-of-domain quality
Consistent and predictable quality	Unpredictable translation quality
Lack of fluency	Good fluency
Hard to handle exceptions to rules	Good for catching exceptions to rules
High development and customization costs	Rapid and cost-effective development costs provided the required corpus exists

Q3. Explain information retrieval and its types

Ans:

INFORMATION RETRIEVAL:

1. Information retrieval (IR) may be defined as a software program that deals with the organization, storage, retrieval and evaluation of information from document repositories particularly textual information.
2. Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).
3. Google search is one of the famous example of Information Retrieval.
4. With the help of figure 6.1, we can understand the process of IR

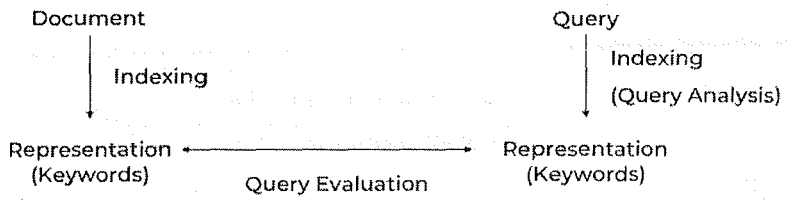


Figure 6.1: Information Retrieval

5. An information retrieval comprises of the following four key elements:
   a. D – Document Representation.
   b. Q – Query Representation.
   c. F – A framework to match and establish a relationship between D and Q.
   d. R (q, di) – A ranking function that determines the similarity between the query and the document to display relevant information.

TYPES OF INFORMATION RETRIEVAL (IR) MODELS:

Information retrieval models predict and explain what a user will find in relevance to the given query. The following are three models that are classified for the Information model (IR) model:

I) **Classical IR Models:**

- It is designed upon basic mathematical concepts and is the most widely-used of IR models.

- Classic Information Retrieval models can be implemented with ease.
- Its examples include Vector-space, Boolean and Probabilistic IR models.
- In this system, the retrieval of information depends on documents containing the defined set of queries. There is no ranking or grading of any kind.
- The different classical IR models take Document Representation, Query representation, and Retrieval/Matching function into account in their modelling.

II) **Non-Classical IR Models:**

- These are completely opposite to the classical IR models.
- These are based on principles other than similarity, probability, Boolean operations.
- Following are the examples of Non-classical IR models: Information logic models, Situation theory models, Interaction models.

III) **Alternative IR Models:**

- It is the enhancement of the classical IR model that makes use of some specific techniques from some other fields.
- Following are the examples of Alternative IR models: Cluster models, Fuzzy models, Latent Semantic Indexing (LSI) models.

Q4. Explain design features of information retrieval systems

Ans:

DESIGN FEATURES OF IR SYSTEMS:

I) **Inverted Index:**

1. The primary data structure of most of the IR systems is in the form of inverted index.
2. We can define an inverted index as a data structure that list, for every word, all documents that contain it and frequency of the occurrences in document.
3. It makes it easy to search for 'hits' of a query word.

II) **Stop Word Elimination:**

1. Stop words are those high frequency words that are deemed unlikely to be useful for searching.
2. They have less semantic weights.
3. All such kind of words are in a list called stop list.
4. For example, articles "a", "an", "the" and prepositions like "in", "of", "for", "at" etc. are the examples of stop words.
5. The size of the inverted index can be significantly reduced by stop list.
6. As per Zipf's law, a stop list covering a few dozen words reduces the size of inverted index by almost half.
7. On the other hand, sometimes the elimination of stop word may cause elimination of the term that is useful for searching.
8. For example, if we eliminate the alphabet "A" from "Vitamin A" then it would have no significance.

III) Stemming:

1. Stemming, the simplified form of morphological analysis, is the heuristic process of extracting the base form of words by chopping off the ends of words.
2. For example, the words laughing, laughs, laughed would be stemmed to the root word laugh.

Q5. Explain the Boolean Model

Ans:

BOOLEAN MODEL:

1. It is the oldest information retrieval (IR) model.
2. The model is based on set theory and the Boolean algebra, where documents are sets of terms and queries are Boolean expressions on terms.
3. The Boolean model can be defined as –

   D: A set of words, i.e., the indexing terms present in a document. Here, each term is either present (1) or absent (0).

   Q: A Boolean expression, where terms are the index terms and operators are logical products – AND, logical sum – OR and logical difference – NOT

   F: Boolean algebra over sets of terms as well as over sets of documents

   If we talk about the relevance feedback, then in Boolean IR model the Relevance prediction can be defined as follows –

   R: A document is predicted as relevant to the query expression if and only if it satisfies the query expression as –

$$((text \vee information) \wedge rerieval \wedge \tilde{} \ theory)$$

4. We can explain this model by a query term as an unambiguous definition of a set of documents.
5. For example, the query term "economic" defines the set of documents that are indexed with the term "economic".
6. Now, what would be the result after combining terms with Boolean AND Operator?
7. It will define a document set that is smaller than or equal to the document sets of any of the single terms.
8. For example, the query with terms "social" and "economic" will produce the documents set of documents that are indexed with both the terms.
9. In other words, document set with the intersection of both the sets.
10. Now, what would be the result after combining terms with Boolean OR operator?
11. It will define a document set that is bigger than or equal to the document sets of any of the single terms.
12. For example, the query with terms "social" or "economic" will produce the documents set of documents that are indexed with either the term "social" or "economic".
13. In other words, document set with the union of both the sets.

Advantages of the Boolean Mode:

- The simplest model, which is based on sets.

- Easy to understand and implement.
- It only retrieves exact matches
- It gives the user, a sense of control over the system.

Disadvantages of the Boolean Model:

- The model's similarity function is Boolean. Hence, there would be no partial matches. This can be annoying for the users.
- In this model, the Boolean operator usage has much more influence than a critical word.
- The query language is expressive, but it is complicated too.
- No ranking for retrieved documents.

Q5. Explain the difference between Data Retrieval and Information Retrieval

Ans:

Table 6.2: Difference between Data Retrieval and Information Retrieval.

	Data Retrieval (DR)	Information Retrieval (IR)
Definition	Data retrieval means obtaining data from a Database Management System (DBMS) such as ODBMS.	An information retrieval (IR) system is a set of algorithms that facilitate the relevance of displayed documents to searched queries
Data	Data retrieval deals with structured data with well-defined semantics	IR deals with unstructured/semi-structured data
Results	Querying a DR system produces exact/precise results or no results if no exact match is found	Querying an IR system produces multiple results with ranking. Partial match is allowed
Queries	The input queries are of the form of SQL or relational algebra.	The input queries are of the form of keywords or natural language.
Ordering of results	Mostly, the results are unordered.	The results are always ordered by relevance.
Accessibility	DR systems can be accessed by only knowledgeable users or any processes run by automation.	IR systems can be accessed by any non expert human unlike that of DR.
Inference	The inference used in data retrieval is of the simple deductive kind	In information retrieval it is far more common to use inductive inference
Model	It follows deterministic modelling approach.	It follows probabilistic modelling approach.
Classification	In DR, we are most likely to be interested in a monothetic classification, that is, one with classes defined by objects possessing	In IR such a classification is one the whole not very useful, in fact more often a polythetic classification is what is wanted. In such a classification each individual in a class will possess only a

	attributes both necessary and sufficient to belong to a class	proportion of all the attributes possessed by all the members of that class
Error response	Error response is much sensitive.	Error response is much insensitive.

Q6. What is Question Answer System in NLP

Ans:

QUESTION ANSWER SYSTEM:

1. Question Answer System is a **branch of learning of Information Retrieval and NLP.**
2. Question answering focuses on building systems that automatically answer questions posed by humans in a natural language.
3. A computer understanding of natural language consists of the capability of a program system to translate sentences into an internal representation so that this system generates valid answers to questions asked by a user.
4. Valid answers mean answers relevant to the questions posed by the user.
5. To form an answer, it is necessary to execute the syntax and semantic analysis of a question.
6. The process of the system is as follows:
   a. Query Processing.
   b. Document Retrieval.
   c. Passage Retrieval.
   d. Answer Extraction.

TYPES OF QUESTION ANSWERING

I) IR-based Factoid Question Answering:

1. Goal is to answer a user's question by finding short text segments on the Web or some other collection of documents.
2. In the question-processing phase a number of pieces of information from the question are extracted.
3. The answer type specifies the kind of entity the answer consists of (person, location, time, etc.).
4. The query specifies the keywords that should be used for the IR system to use in searching for documents.

II) Knowledge-based question answering:

1. It is the idea of answering a natural language question by mapping it to a query over a structured database.
2. The logical form of the question is thus either in the form of a query or can easily be converted into one.

3. The database can be a full relational database, or simpler structured databases like sets of RDF triples.
4. Systems for mapping from a text string to any logical form are called semantic parsers.
5. Semantic parsers for question answering usually map either to some version of predicate calculus or a query language like SQL or SPARQL.

CHALLENGES IN QUESTION ANSWERING:

I) Lexical Gap:

1. In a natural language, the same meaning can be expressed in different ways.
2. Because a question can usually only be answered if every referred concept is identified, bridging this gap significantly increases the proportion of questions that can be answered by a system.

II) Ambiguity:

1. It is the phenomenon of the same phrase having different meanings; this can be structural and syntactic (like "flying planes") or lexical and semantic (like "bank").
2. The same string accidentally refers to different concepts (as in money bank vs. river bank) and polysemy, where the same string refers to different but related concepts (as in bank as a company vs. bank as a building).

III) Multilingualism:

1. Knowledge on the Web is expressed in various languages.
2. While RDF resources can be described in multiple languages at once using language tags, there is not a single language that is always used in Web documents.
3. Additionally, users have different native languages. A QA system is expected to recognize a language and get the results on the go!

Q7. Write short notes on Categorization

Ans:

CATEGORIZATION:

1. Text classification also known as text tagging or text categorization is the process of categorizing text into organized groups.
2. By using Natural Language Processing (NLP), text classifiers can automatically analyze text and then assign a set of pre-defined tags or categories based on its content.
3. Text Classification is the processing of labeling or organizing text data into groups.
4. It forms a fundamental part of Natural Language Processing.
5. In the digital age that we live in we are surrounded by text on our social media accounts, in commercials, on websites, Ebooks, etc.
6. The majority of this text data is unstructured, so classifying this data can be extremely useful.
7. Sentiment Analysis is an important application of Text Classification.

Approaches:

Text Classification can be achieved through three main approaches

I) **Rule-based approaches:**

1. These approaches make use of handcrafted linguistic rules to classify text.
2. One way to group text is to create a list of words related to a certain column and then judge the text based on the occurrences of these words.
3. For example, words like "fur, "feathers", "claws", and "scales" could help a zoologist identify texts talking about animals online.
4. These approaches require a lot of domain knowledge to be extensive, take a lot of time to compile, and are difficult to scale.

II) **Machine learning approaches:**

1. We can use machine learning to train models on large sets of text data to predict categories of new text.
2. To train models, we need to transform text data into numerical data - this is known as feature extraction.
3. Important feature extraction techniques include bag of words and n-grams.
4. There are several useful machine learning algorithms we can use for text classification.
5. The most popular ones are:
   a. Naive Bayes classifiers
   b. Support vector machines
   c. Deep learning algorithms

III) **Hybrid approaches:**

1. These approaches are a combination of the two algorithms above.
2. The y make use of both rule-based and machine learning techniques to model a classifier that can
3. be fine-tuned in certain scenarios.

Applications:

Text Classification has a wide array of applications. Some popular uses are:

1. Spam detection in emails.
2. Sentiment analysis of online reviews.
3. Topic labeling documents like research papers.
4. Language detection like in Google Translate.
5. Age/gender identification of anonymous users.
6. Tagging online content.
7. Speech recognition used in virtual assistants like Siri and Alexa.

Q8. Write short notes on Summarization & Sentiment Analyses

Ans:

SUMMARIZATION:

1. A summary is a reductive transformation of a source text into a summary text by extraction or generation.
2. The goal of summarization is to produce a shorter version of a source text by preserving the meaning and the key contents of the original document.
3. A well written summary can significantly reduce the amount of work needed to digest large amounts of text.

Types of Text summarization:

I) Extraction based Summarization:

1. The extractive text summarising approach entails extracting essential words from a source material and combining them to create a summary.
2. Without making any modifications to the texts, the extraction is done according to the given measure.
3. This approach works by detecting key chunks of the text, cutting them out, then stitching them back together to create a shortened form.
4. As a result, they rely only on phrase extraction from the original text.

II) Abstractive Summarization:

1. Another way of text summarization is abstractive summarization.
2. We create new sentences from the original content in this step.
3. This is in contrast to our previous extractive technique, in which we only utilized the phrases that were present.
4. It's possible that the phrases formed by abstractive summarization aren't present in the original text.
5. When abstraction is used for text summarization in deep learning issues, it can overcome the extractive method's grammatical errors.
6. Abstraction is more efficient than extraction.
7. The text summarising algorithms necessary for abstraction, on the other hand, are more complex to build, which is why extraction is still widely used.

III) Domain-Specific:

1. In domain-specific summarization, domain knowledge is applied.
2. Specific context, knowledge, and language can be merged using domain-specific summarizers.
3. For example, models can be combined with the terminology used in medical science so that they can better grasp and summarise scientific texts.

IV) Query-based:

1. Query-based summaries are primarily concerned with natural language questions.
2. This is similar to the search results on Google.
3. When we type questions into Google's search field, it occasionally returns websites or articles that answer our questions.
4. It displays a snippet or summary of an article that is relevant to the query we entered.

V) Generic:

1. Generic summarizers, unlike domain-specific or query-based summarizers, are not programmed to make any assumptions.
2. The content from the source document is simply condensed or summarised.

Q8. Write short notes on Sentiment Analyses

Ans:

SENTIMENT ANALYSES:

1. It is also known as **Opinion Mining**.
2. Sentiment analysis is the process of detecting positive or negative sentiment in text.
3. Sentiment analysis is an approach to natural language processing (NLP) that identifies the emotional tone behind a body of text.
4. This is a popular way for organizations to determine and categorize opinions about a product, service, or idea.
5. It involves the use of data mining, machine learning (ML) and artificial intelligence (AI) to mine text for sentiment and subjective information.
6. Since customers express their thoughts and feelings more openly than ever before, sentiment analysis is becoming an essential tool to monitor and understand that sentiment.
7. Automatically analysing customer feedback, such as opinions in survey responses and social media conversations, allows brands to learn what makes customers happy or frustrated, so that they can tailor products and services to meet their customers' needs.
8. For example, using sentiment analysis to automatically analyse 4,000+ reviews about your product could help you discover if customers are happy about your pricing plans and customer service.
9. Sentiment analysis algorithms fall into one of three buckets:
   a. **Rule based:** These systems automatically perform sentiment analysis based on the set of manually crafted rules.
   b. **Automatic:** These systems rely on machine learning techniques to learn from data.
   c. **Hybrid:** These systems combine both rule based and automatic approaches.

Types of Sentiment Analysis:

1. **Fine-grained sentiment analysis** provides a more precise level of polarity by breaking it down into further categories, usually very positive to very negative This can be considered the opinion equivalent of ratings on a 5-star scale.
2. **Emotion detection** identifies specific emotions rather than positivity and negativity. Examples could include happiness, frustration, shock, anger and sadness.
3. **Intent-based analysis** recognizes actions behind a text in addition to opinion. For example, an online comment expressing frustration about changing a battery could prompt customer service to reach out to resolve that specific issue.
4. **Aspect-based analysis** gathers the specific component being positively or negatively mentioned. For example, a customer might leave a review on a product saying the battery life was too short. Then, the system will return that the negative sentiment is not about the product as a whole, but about the battery life.

Q9. Write short notes on Named Entity Recognition

Ans:

NAMED ENTITY RECOGNITION:

1. Named entity recognition (NER) is also called **entity identification or entity extraction.**
2. It is a natural language processing (NLP) technique that automatically identifies named entities in a text and classifies them into predefined categories.
3. Entities can be names of people, organizations, locations, times, quantities, monetary values, percentages, and more.
4. **Example:**

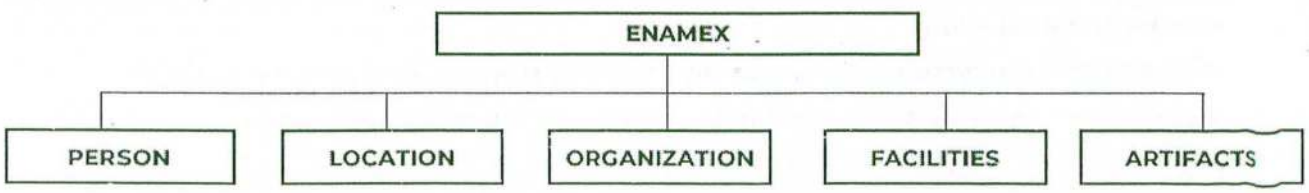
Ousted WeWork founder Adam Neumann lists his Manhattan penthouse for $37.5 million

[organization] [person] [location] [monetary value]

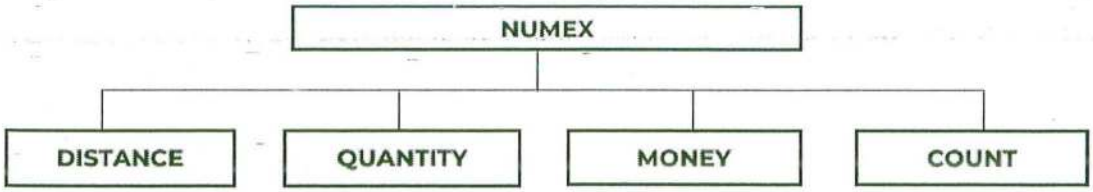
5. With named entity recognition, you can extract key information to understand what a text is about, or merely use it to collect important information to store in a database.

Types of NER:

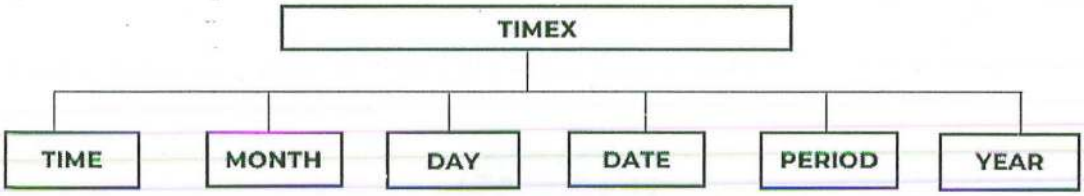
I) **Entity Name Types:**



II) **Numerical Expressions:**



III) **Time Expressions:**



Challenges of NER:

I) **Ambiguity:**

1. Ambiguity between common and proper nouns.
2. Example: common words such as "Roja" meaning Rose flower is a name of a person.

II) **Spell variations:**

One of the major challenges in the web data is that we find different people spell the same entity with differently.

III) **Less Resources:**

1. Most of the Indian languages are less resource languages.
2. Either there are no automated tools available to perform pre-processing tasks required
3. for NER such as Part-of-speech tagging, chunking.
4. Or for languages where such tools are available, they have less performance.

IV) **Lack of easy availability of annotated data:**

1. There are isolated efforts in the development of NER systems for Indian languages.
2. There is no easy availability and access for NE annotated corpus in the community.

V) **Morphologically rich:**

1. Most of the Indian languages are morphologically rich and agglutinative
2. There will be lot of variations in word forms which make machine learning difficult.

VI) **No Capitalization feature:**

1. In English, capitalization is one of the main features, whereas that's not there in Indian languages
2. Machine learning algorithms have to identify different features.