Sample Questions

Computer Engineering

Subject Name: Natural Language Processing **Semester: VIII**

Multiple Choice Questions

1.	"He went to the bank". identify the challenge of NLP
Option A:	Discourse resolution
Option B:	**Noun resolution**
Option C:	Verb resolution
Option D:	Pronoun resolution
2.	" Bat is flying in the sky" Identify the dependency checking to perform sense disambiguation of 'Bat'
Option A:	**Bat -> sky, fly**
Option B:	Bat-> sky
Option C:	Sky-> fly
Option D:	Bat-> fly
3.	N-grams are defined as the combination of N keywords together. How many bigram can be generated from given sentence: "Data segmentation is a great source to learn text summarization"
Option A:	7
Option B:	8
Option C:	**9**
Option D:	10
4.	"Given a input sentence "" The crane is loaded"" How will you determine the correct sense of the word 'crane'"
Option A:	Word will be searched in lexicon and first sense of crane will be identified
Option B:	Identify the POS of crane and load, apply rule and determine correct meaning
Option C:	**Determine the clue word load and find the dependency between crane and load. Match with all the definitions of crane in the lexicon. Best match is the answer.**
Option D:	As clue words such as fly, sky are not part of input, so correct sense of crane is machinery sense
5.	HMM model formula P(q2|x2,q1)=p(x2|q2)*P(q2|q1) This formula does not contain
Option A:	State transition Probability
Option B:	Emission probability
Option C:	CDF
Option D:	**Initial state**
6.	In Porter stemmer algorithm,*v* indicates
Option A:	**Stem contains a vowel**
Option B:	Stem contains any character
Option C:	Stem contains VC combinations

7.	Who invented Wordnet
Option A:	Tomas Mikolov
Option B:	Atlas University
Option C:	PENN treebank
Option D:	**Princeton University**
8.	"The Tajmahal is one of the seventh wonder of the world". Identify the application of NLP in the word 'TajMahal'
Option A:	**Named entity recognition**
Option B:	QA system
Option C:	Text categorization
Option D:	Sentiment analysis
9.	The contraction of the morpheme “is”, as in, “That’s the way to do it.” is an example of:
Option A:	**Clitic**
Option B:	Inflection
Option C:	Derivation
Option D:	Suffix
10.	Lesk algorithm
Option A:	converts words to vectors
Option B:	finds comparison between two words
Option C:	**measures overlap between sense definitions for all words in context**
Option D:	check for similarity between words in context
11.	What is morphology?
Option A:	The study of linguistic sounds
Option B:	**It is a study of the way words are built up from smaller meaning-bearing units called morphemes.**
Option C:	The study of the structural relationships between words.
Option D:	The study of linguistic units larger than a single utterance.
12.	Select correct example of inflectional morpheme?
Option A:	Read --> Reader
Option B:	Teach --> Teacher
Option C:	**Tall --> Taller**
Option D:	Play --> Player
13.	Parts of speech can be divided into two broad super categories _____
Option A:	Parent class and derived class
Option B:	**Closed class and open class**
Option C:	Sentence class and character class
Option D:	Sub class and child class
14.	Bigram model also called as _____
Option A:	**First-order Morkov model**
Option B:	Second-order Morkov model
Option C:	Third-order Morkov model
Option D:	(N-1)th-order Morkov model

15.	"Custemer Review system" is example of one of the following?
Option A:	Machine Translation
Option B:	**Sentiment Analysis**
Option C:	Question-Answering system
Option D:	Text-Summerization
16.	"I saw someone on the hill with a telescope." is the example of which type of ambiguity?
Option A:	Lexical Ambiguity
Option B:	Semantic Ambiguity
Option C:	**Syntactic Ambiguity**
Option D:	Pragmatic Ambiguity
17.	Sentiment analysis is also called as ____
Option A:	Summarization
Option B:	Question-Answering
Option C:	**Opinion Mining**
Option D:	Named-Entity Recognition.
18.	What is the task of Robust Word Sense Disambiguation (WSD) for word in given sentence?
Option A:	Define a concept or word meaning
Option B:	Measure overlap between sense definitions for all words in context
Option C:	Define word without senses
Option D:	**selecting the correct sense for a word in a given sentence**
19.	"Please maintain silence" is the example of _____
Option A:	Wh-subject Question
Option B:	Yes-No Question
Option C:	**Imperative sentence**
Option D:	Declarative sentence
20.	Select correct constraint on coreference for given example “John and Mary have Hyundai cars. They love them".
Option A:	Number agreement
Option B:	Gender agreement
Option C:	**Person and Case agreement**
Option D:	Syntactic constraint.
21.	Natural language processing is a sub-domain of,
Option A:	Networking
Option B:	**Artificial Intelligence**
Option C:	Algorithms
Option D:	Databases
22.	Which of this is not an application of NLP?
Option A:	Speech Understanding
Option B:	Chatbot
Option C:	**Scanned Image Classification**
Option D:	News Clustering
23.	This kind of ambiguity occurs when a sentence is parsed in different ways.
Option A:	Lexical Ambiguity
Option B:	**Syntactic Ambiguity**
Option C:	Semantic Ambiguity

Option D:	Pragmatic Ambiguity
24.	“Appoint→Appointee” is an example of ---------- morphology.
Option A:	**Derivational**
Option B:	Inflectional
Option C:	Compounding
Option D:	Cliticization
25.	The stemming algorithm is used to,
Option A:	Form complex words from base form
Option B:	Generats the parse tree of a sentence
Option C:	Check meaning of a word in dictionary
Option D:	**Reduce inflected form of a word to a single base form**
26.	P(dog | the big) is an example of ------ model
Option A:	Unigram
Option B:	Bigram
Option C:	**Trigram**
Option D:	Quadrigram
27.	Which of this is not true about Morphology?
Option A:	Provides systematic rules for forming new words in a language
Option B:	**Provide rules for forming sentences in a language**
Option C:	Can be used to verify if a word is legitimate in a language
Option D:	Group words into classes
28.	CFG captures ----------
Option A:	**Constituency and ordering**
Option B:	word meaning
Option C:	relation between words
Option D:	sentence meaning
29.	Which of the following is a Rule based POS tagger?
Option A:	HMM Tagger
Option B:	Ngram Tagger
Option C:	**ENGTWOL Tagger**
Option D:	Brill Tagger
30.	Syntax analysis concerns with:
Option A:	the way words are built up from smaller meaning bearing units
Option B:	what words mean and how these meanings combine in sentences to form sentence meanings
Option C:	how the immediately preceding sentences affect the interpretation of the next sentence
Option D:	**how words are put together to form correct sentences and what structural role each word has**
31.	Which of the following is not a sequence labeling technique?
Option A:	Maximum Entropy
Option B:	**Context Free Grammar**
Option C:	Conditional Random Fields
Option D:	Hidden Markov Model

32.	Which of the following is an example of "hyponym-hypernym" semantic relationship?
Option A:	**Car-Vehicle**
Option B:	Car-Wheel
Option C:	Wheel-Car
Option D:	Car-Ford
33.	The root form of a word in Wordnet dictionary is called
Option A:	Stem
Option B:	Sense
Option C:	Gloss
Option D:	**Lemma**
34.	Roughly, Semantic analysis is----------
Option A:	**Language Understanding**
Option B:	Language Generation
Option C:	Language Preprocessing
Option D:	Language Translation
35.	"All boys love cricket ". How is this sentence represented in First Order Logic form?
Option A:	∃x boys(x)→love(x,cricket)
Option B:	**∀x boys(x)→love(x,cricket)**
Option C:	∃x,y love(x) ∧ cricket(y)
Option D:	∀x boys(x) ∧ love(x,cricket)
36.	Pragmatic refers to
Option A:	Literal meaning
Option B:	**Intended meaning**
Option C:	Structural meaning
Option D:	Wordnet dictionary meaning
37.	"John bought an Acura Integra today, but the engine seemed noisy." Which of the following is an Inferrable referent?
Option A:	John
Option B:	Acura
Option C:	**Engine**
Option D:	Noisy
38.	Shivaji→ शिवाजी Is an example of:
Option A:	Translation
Option B:	Transfer
Option C:	**Transliteration**
Option D:	Generation
39.	In which of the summarization technique, summary contains the sentences from the given document only?
Option A:	**Extractive Summarization**
Option B:	Abstractive summarization
Option C:	Mixed Summarization
Option D:	Copied summarization

40.	Which of this is not a reference resolution algorithm?
Option A:	Hobb's Algorithm
Option B:	Lappin and Leass's Algorithm
Option C:	Centering Algorithm
Option D:	**Lesk's Algorithm**

Q.1 Explain how word sense disambiguation will be useful for resolving ambiguity

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.

Q.2 Explain the text preprocessing steps of Natural language processing with an example

1. Tokenization

Tokenization is the process of exchanging sensitive data for nonsensitive data called "tokens" that can be used in a database or internal system without bringing it into scope.

Although the tokens are unrelated values, they retain certain elements of the original data—commonly length or format—so they can be used for uninterrupted business operations. The original sensitive data is then safely stored outside of the organization's internal systems.

Unlike encrypted data, tokenized data is undecipherable and irreversible. This distinction is particularly important: Because there is no mathematical relationship between the token and its original number, tokens cannot be returned to their original form without the presence of additional, separately stored data. As a result, a breach of a tokenized environment will not compromise the original sensitive data.

2. Stop words removal

Stopwords are the **words** in any language which does not add much meaning to a sentence. They can safely be ignored without sacrificing the meaning of the sentence. For some search engines, these are some of the most common, short function words, such as the, is, at, which, and on. In this case, stop words can cause problems when searching for phrases that include them, particularly in names such as “The Who” or “Take That”.

3. Stemming

Stemming is a technique used to extract the base form of the words by removing affixes from them. It is just like cutting down the branches of a tree to its stems. For example, the stem of the words ***eating, eats, eaten*** is ***eat***.

Search engines use stemming for indexing the words. That’s why rather than storing all forms of a word, a search engine can store only the stems. In this way, stemming reduces the size of the index and increases retrieval accuracy.

4. Lemmatization

Lemmatization technique is like stemming. The output we will get after lemmatization is called ‘lemma’, which is a root word rather than root stem, the output of stemming. After lemmatization, we will be getting a valid word that means the same thing.

NLTK provides **WordNetLemmatizer** class which is a thin wrapper around the **wordnet** corpus. This class uses **morphy()** function to the **WordNet CorpusReader** class to find a lemma. Let us understand it with an example –

Q.3 Explain machine translation and its types

MACHINE TRANSLATION:

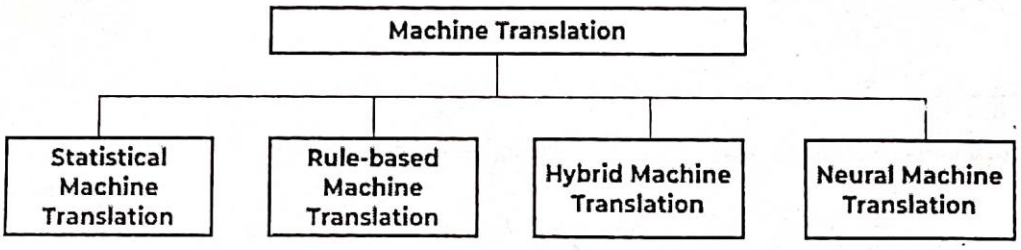
1. Machine Translation is also known as **robotized interpretation or automated translation.**
2. Machine Translation or MT is simply a procedure when a computer software translates text from one language to another without human contribution.
3. At its fundamental level, machine translation performs 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.

Q.4 What is language model? Explain N gram model

A language model in NLP is a probabilistic statistical model that determines the probability of a given sequence of words occurring in a sentence based on the previous words. It helps to predict which word is more likely to appear next in the sentence.

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:

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

$$P(w_1 w_2 w_3 \ldots w_n) = \pi_i P(w_i \mid w_1 w_2, \ldots w_n)$$

14. Simplifying the above formula using Markov assumptions:

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

15. For unigram:

$$P(w_1 w_2, \ldots w_n) \approx \pi_i P(w_i)$$

16. For Bigram:

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

Q.5 What is parsing? Explain Top-down and Bottom-up approach of parsing with suitable example.

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:**

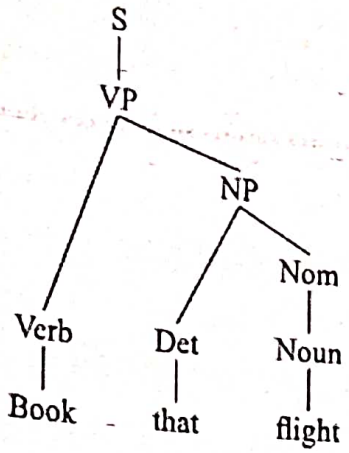


Figure 3.2

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:**

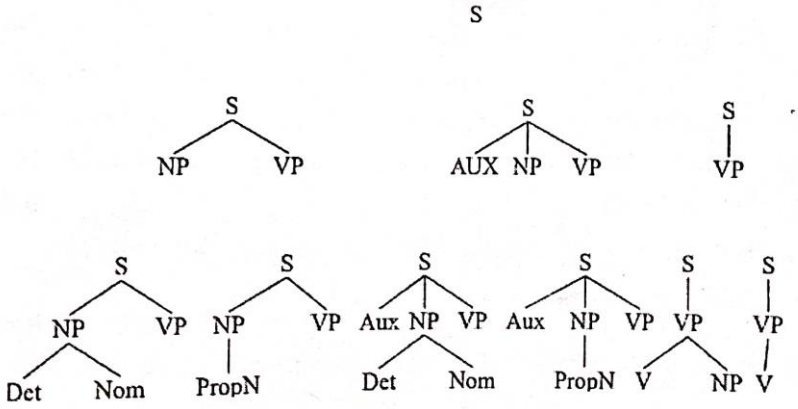


Figure 3.3: Top-down parsing example

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.

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:

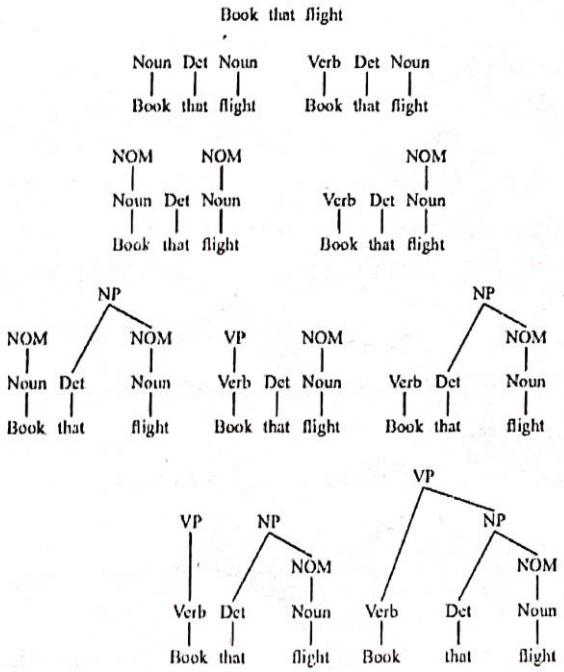


Figure 3.4: Bottom-up parsing example

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.

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.

Q.6 Discuss various approaches to perform Part-Of-Speech (POS) tagging

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.

Q.7 Explain derivational and inflectional morphology in detail with suitable example

1. Inflection

- When the word stem is fused with the grammatical morpheme then it generally results in a word of the similar class as the original stem.
- If we consider the English language then the things that get inflected are nouns, verbs, and sometimes adjectives. So, the affixes are quite small in number.
- Nouns have simple inflectional morphology. Examples of the Inflection of noun in English are given below, here an affix is marking plural.
  - mat (-s), cat(-s)
  - ibis(-es)
  - thrush(-es)
  - waltz(-es), finch(-es), box(-es)
  - butterfly(-lies)
  - ox (oxen), mouse (mice) [irregular nouns]
- A possessive affix is a suffix or prefix attached to a noun to indicate its possessor. the following are some affixes marking possessive
  - Regular singular noun- llama's

- The Table 2.3.1 shows the morphological forms for regular verbs.

Table 2.3.1

Stem	Talk	Urge	Cry	tap
-s form	Talks	urges	cries	Taps
-ing form	Talking	urging	Crying	Tapping
Past form or -ed participle	Talked	urged	Cried	tapped

- The morphological form for the irregular verbs is shown in the Table 2.3.2

Table : 2.3.2

Stem	Eat	Think	put
-s form	Eats	Thinks	puts
-ing form	eating	Thinking	putting
Past form	Ate	Thought	put
-ed participle	Eaten	Thought	put

2. Derivation

- The combination of a word stem with a grammatical morpheme usually resulting in a word of a different class, often with a meaning hard to predict exactly.
- Nominalizations are nothing but the nouns which aregenerated from adjectives in English. The Table 2.3.3 shows the formation of new nouns, often from verbs or adjectives.

Table 2.3.3

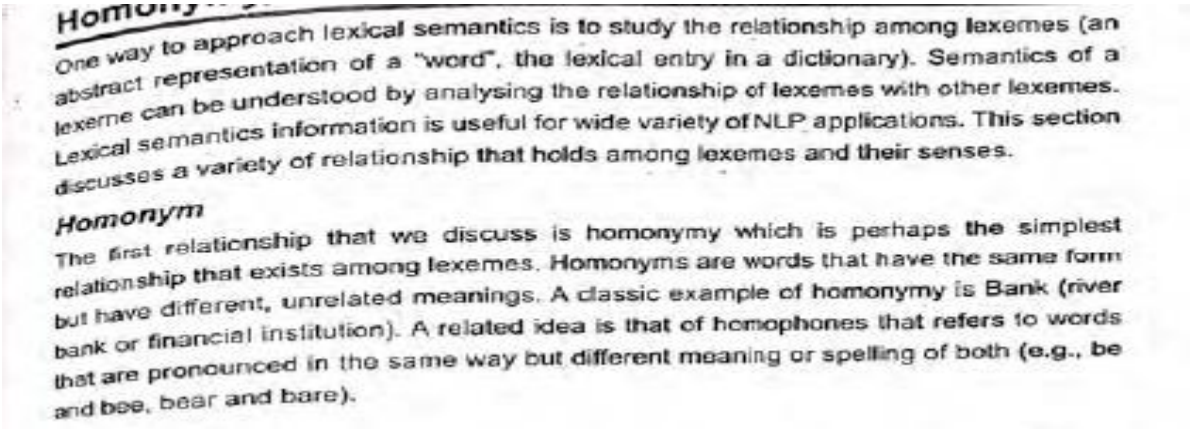
Suffix	Base Verb/Adjective	Derived Noun
-ation	Computerize (V)	Computerization
-ee	Appoint (V)	Appointee
-er	Kill (V)	Killer
-ness	Fuzzy (A)	Fuzziness

- Adjectives can also be derived from nouns or verbs. The Table 2.3.4 shows the objectives derived from the verbs/noun.

Table 2.3.4

Suffix	Base Verb/Adjective	Derived Noun
-al	Computation (N)	Computational
-able	embrace (V)	Embraceable
-less	clue (A)	Clueless
-ness	Fuzzy (A)	Fuzziness

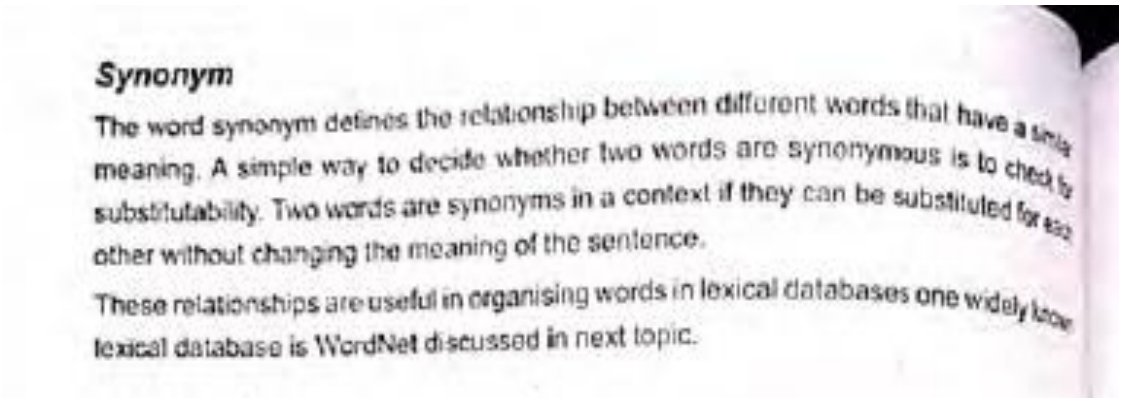
Q.8 Explain following Relations among lexemes & their senses, Homonymy, Synonymy, Hyponymy with example



One way to approach lexical semantics is to study the relationship among lexemes (an abstract representation of a "word", the lexical entry in a dictionary). Semantics of a lexeme can be understood by analysing the relationship of lexemes with other lexemes. Lexical semantics information is useful for wide variety of NLP applications. This section discusses a variety of relationship that holds among lexemes and their senses.

Homonym

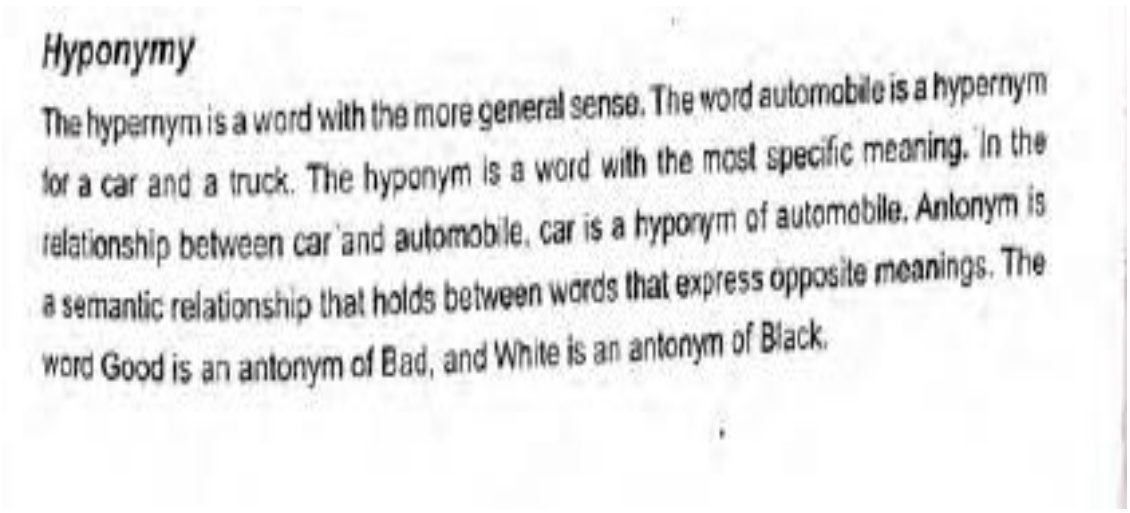
The first relationship that we discuss is homonymy which is perhaps the simplest relationship that exists among lexemes. Homonyms are words that have the same form but have different, unrelated meanings. A classic example of homonymy is Bank (river bank or financial institution). A related idea is that of homophones that refers to words that are pronounced in the same way but different meaning or spelling of both (e.g., be and bee, bear and bare).



Synonym

The word synonym defines the relationship between different words that have a simila meaning. A simple way to decide whether two words are synonymous is to check for substitutability. Two words are synonyms in a context if they can be substituted for each other without changing the meaning of the sentence.

These relationships are useful in organising words in lexical databases one widely know lexical database is WordNet discussed in next topic.



Hyponymy

The hypernym is a word with the more general sense. The word automobile is a hypernym for a car and a truck. The hyponym is a word with the most specific meaning. In the relationship between car and automobile, car is a hyponym of automobile. Antonym is a semantic relationship that holds between words that express opposite meanings. The word Good is an antonym of Bad, and White is an antonym of Black.

Q.9 What are the five types of referring expression? Explain with example

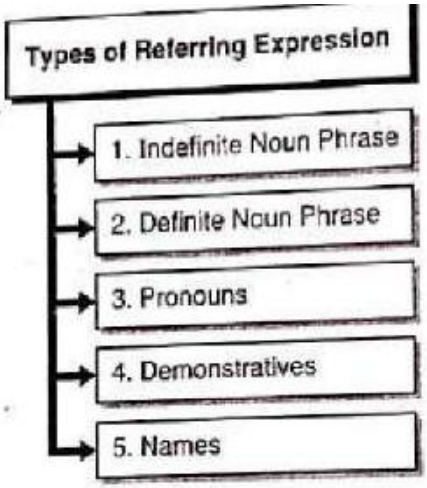


Fig. 5.2.3 : Types of Referring Expressions

1. **Indefinite noun phrases**
   - Typically, an indefinite noun phrase introduces a new entity into the discourse and would not be used as a referring phrase to something else
   - The exception is in the case of cataphora: A Soviet pop star was killed at a concert in Moscow last night. Igor Talkov was shot through the heart as he walked on stage.
2. **Definite noun phrases**
   - Definite reference is used to refer to an entity identifiable by the reader because it is either
     a) already mentioned previously (in discourse), or
     b) contained in the reader's set of beliefs about the world (pragmatics), or
     c) the object itself is unique.
   - For e.g. Mr. Torres and his companion claimed a hard-shelled black vinyl suitcase. The police rushed the suitcase to the Trans-Uranium Institute
3. **Pronouns**
   - It refer to entities that were introduced fairly recently.
   - Nominative (he, she, it, they, etc.)
     - For e.g., The German authorities said a Colombian who had lived for a long time in the Ukraine flew in from Kiev. He had 300 grams of plutonium 239 in his baggage.

Oblique (him, her, them, etc.)

- For e.g. Undercover investigators negotiated with three members of a criminal group and arrested them after receiving the first shipment. Possessive (his, her, their, etc. + hers, theirs, etc.)
- For e.g., He3 had 300 grams of plutonium 239 in his baggage. The suspected smuggler denied that the materials were his. (*chain). Reflexive (himself, themselves, etc.)
- For e.g. There appears to be a growing problem of disaffected loners who cut themselves off from all groups.

4. Demonstratives – this and that

- Demonstrative pronouns can either appear alone or as determiners this ingredient, that spice
- These NP phrases with determiners are ambiguous.
- They can be indefinite : *I saw this beautiful car today.*

  Or they can be definite :*I just bought a copy of Thoreau's Walden. I had bought one five years ago. That one had been very tattered; this one was in much better condition.*

5. Names

Names can occur in many forms, sometimes called name variants.

Approach to coreference resolution

- Naively identify all referring phrases for resolution:
  - all Pronouns
  - all definite NPs
  - all Proper Nouns
- Filter things that look referential but, in fact, are not
  - e.g. geographic names, the United State
  - pleonastic "it", e.g. it's 3:45 p.m., it was cold
  - non-referential "it" , "they" , "there"
- For e.g. it was essential, important, is understood, they say, there seems to be a mistake

Identify Referent Candidates

- All noun phrases (both indef. and def.) are considered potential referent candidates.
- A referring phrase can also be a referent for a subsequent referring phrase.

Q.10 What are the stages of NLP? Explain with example.

PHASES/STAGES OF NLP:

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

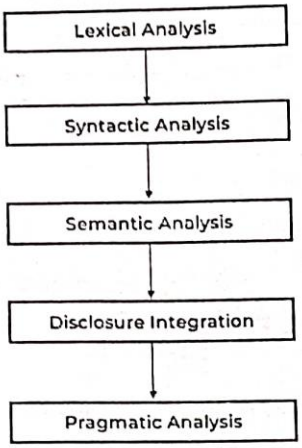


Figure 1.3: Stages of NLP

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.

Q.11 What are basic regular expression patterns? Give brief answer for each with example.

- Regular expressions are also called as regexes. It is used for pattern matching standards for string passing and replacement.
- Regular expressions are powerful way to find and replace string that take a defined format. For example, regular expressions are used to parse email addresses, url's, dates, log files, configuration file, command line, programming script or switches.
- Regular expression is a useful tool to design language compilers as well as they are used in natural language processing for tokenization, describing lexicons, morphological analysis, etc. Many of us have used simple form of regular expression for searching file patterns in MS DOS for example dir*.txt.
- The Unix based editor Ed regular expression for popular in computer science. Perl is the first language that provided integrated support for regular expression. It used slash (/) around each regular expression here we are also following the same notation. however, slashes are not a part of regular expression.
- Regular expressions introduced in 1956 by Kleene. It was originally studied as part of theory of computation. Regular expression is an algebraic formula whose value is a pattern consisting of a set of Strings known as the language of expression.
- The simple type of regular expression contains a single symbol.
- For example the expression : /a/ - The expression denotes a set containing a string 'a'.
- It also pacifies sequence of characters.
- For example : /avengers/ - It denotes Indian notes that contain screen avengers and nothing else.
- **Brackets:** Characters are group by putting them between square brackets. This way any character in the class will match One character in the input.

- **For example,**

  /[abcd]/ will match any of a, b, c, and d.

  / [0123456789]/ specifies any single digit.

- **Range:**Sometimes regular expression led to cumbersome notation.

  For example:

  /[abcdefghijklmnopqrstuvwxyz] - It specifies any lowercase letter.

- In such cases a dash is used to specify a range.

- **For Example:**

  / [3-6]/ specifies any one of the digits 3,4,5, or 6.

  T /[c-f]/ specifies any one of the letter c,d,e, or f.

- **Caret^**: The caret is used at the beginning of the regular expression to specify what a single character cannot be.

- **For example:**

  /[^x] - matches any single character except x

  /[^A-Z]/ --> not an upper-case letter

  /[^Tt]/ --> neither 'T' nor 't'

  /[^\.]/ --> not a period

  /[p^]/ --> either 'p' or '^'

  /x^y/ --> the pattern 'x^y'

- Regular expressions are case sensitive. the pattern /t/ matches *t* but not *T*. It means pattern /Tree/ will not match pattern/tree/. To solve this issue the regular expression for this is written as / [Tt]ree/.

- *** or +** : The use of * or + allows you to add 1 or more of a preceding character.

- **For example:**

  oo*h! → 0 or more of a previous character (e.g. ooh!,ooooh!)

  o+h! → 1 or more of a previous character (e.g. ooh!,oooooh!) .

  paa+ →paa, paaaa, paaaaaa, paaaaaaa

- **?** : The question mark ? letters optionality of the previous expression.

Q.12 What are the attachments for fragment of English? Explain with example.

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

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

Q.13 Differentiate between Derivational and Inflectional morphemes.

Inflectional VS. Derivational Affixes

	Inflectional	Derivational
1.	Never change the part of speech of the base to which they are added.	1. They may change the part of speech of the words to which they are added.
2.	They are related to syntax.	2. They are related to semantics.
3.	Inflectional affixes have a regular meaning.	3. Their meaning is irregular.
4.	They are farther to the root than derivationals.	4. They are nearer to the root.
5.	They close off the word	5. They never close off the word
6.	They can only be added to the stems.	6. The can be added to the base.
7.	Inflection uses a close set of suffixes and their number is quite smaller than derivationals	7. Derivation uses an open set of affixes i.e, their number is much bigger that that of inflectionals.
8.	By adding an inflectional suffix, a new form of the same morpheme will be obtained.	8.By adding aa derivational suffix, a new word (lexeme) will be obtained.
9.	Only one suffix can occur within a word	9. More than one affix can occur within one word

Q.14 Define POS tagging. Explain rule-based POS tagging with example.

1. Part-Of-Speech Tagging

The part of speech tagging is a process of assigning corresponding part of speech like noun, verb, adverb, adjective, verb to each word in a sentence. It is a process of converting a sentence to forms – list of words, list of tuples (where each tuple is having a form (word, tag)). The tag in case of is a part-of-speech tag, and signifies whether the word is a noun, adjective, verb, and so on.

...of descriptors in the given tokens is called Tagging. The descriptor ... given tag in a given context.

Rule-Based Part-of-Speech Tagging

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

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

We can also understand Rule-based POS tagging by its two-stage architecture .In the first stage, it uses a dictionary to assign each word a list of potential parts-of-speech. 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

Rule-based POS taggers possess the following properties –

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.

Advantages of Rule Based Taggers: -

a. Small set of simple rules.
b. Less stored information.

Drawbacks of Rule Based Taggers

a. Generally less accurate as compared to stochastic taggers.

Q15 What are the reference phenomenons? Explain types of referring expression.

The set of referential phenomena that natural languages provide is quite rich indeed. Five types of referring expressions: indefinite noun phrases, definite noun phrases, pronouns, demonstratives, and one-anaphora. Three types of referents that complicate the reference resolution problem: inferrables, discontinuous sets, and generics.

Indefinite Noun Phrases: Indefinite reference introduces entities that are new to the hearer into the discourse context. The most common form of indefinite reference is marked with the determiner a (or an), as in A, but it can also be marked by a quantifier such as B or even the determiner this C.

A. I saw an Acura Inlegra today.

B. Some Acura Integras were being unloaded at the local dealership today.

C. I saw this awesome Acura Integra today.

Such noun phrases evoke a representation for a new entity that satisfies the given description into the discourse model. The indefinite determiner a does not indicate whether the entity is identifiable to the speaker, which in some cases leads to a specific/ non-specific ambiguity. Example A only has the specific reading, since the speaker has a particular Integra in mind, particularly the one she saw. In sentence D, on the other hand, both readings are possible.

D. I am going to the dealership to buy an Acura Integra today.

That is, the speaker may already have the Inlegra picked out (specific), or may just be planning to pick one out that is to her liking (nonspecific). The readings may be disambiguated by a subsequent referring expression in some contexts; if this expression is definite then the reading is specific (I hope they still have it), and if it is indefinite then the reading is nonspecific (I hope they have a car I like). This rule has exceptions, however; for instance definite expressions in certain modal contexts (I will park it in my garage) are compatible with the nonspecific reading.

Definite Noun Phrases Definite reference is used to refer to an entity that is identifiable to the listener, either because it has already been mentioned in the discourse context (and thus is represented in the discourse model), it is contained in the hearer's set of beliefs about the world, or the uniqueness of the object is implied by the description itself. The case in which the referent is identifiable from discourse context is shown in (S1). (S1) I saw an Acura Integra today. The Integra was white and needed to be washed.

Pronouns Another form of definite reference is pronominalization, illustrated in example (S2). (S2) I saw an Acura Integra today. It was white and needed to be washed. The constraints on using pronominal reference are stronger than for full definite noun phrases, requiring that the referent have a high degree of activation or salience in the discourse model. Pronouns usually (but not always) refer to entities that were introduced no further back than one or two sentences back in the ongoing discourse, whereas definite noun phrases can often refer further back.

A . John went to Bob's party, and parked next to a beautiful Acura Integra. b. He went inside and talked to Bob for more than an hour. c. Bob told him that he recently got engaged. d. ?? He also said that he bought it yesterday. d.' He also said that he bought the Acura yesterday. By the time the last sentence is reached, the Integra no longer has the degree of salience required to allow for pronominal reference to it. Pronouns can also participate in cataphora, in which they are mentioned before their referents are, as in example (S2). S2 Before he bought it, John checked over the Integra very carefully. Here, the pronouns he and it both occur before their referents are introduced. Pronouns also appear in quantified contexts in which they are considered to be bound, as in example (S3). BOUND (S3) Every woman bought her Acura at the local dealership.

Under the relevant reading, her does not refer to some woman in context, but instead behaves like a variable bound to the quantified expression every woman. We will not be concerned with the bound interpretation of pronouns in this chapter.

Demonstratives: Demonstrative pronouns, like this and that, behave somewhat differently than simple definite pronouns like it. They can appear either alone or as determiners, for instance, this Acura, that Acura. The choice between two demonstratives is generally associated with some notion of spatial proximity: this indicating closeness and that signalling distance.

Spatial distance might be measured with respect to the discourse participants' situational context, as in (S4). (S4) [John shows Bob an Acura Integra and a Mazda Miata] Bob: (pointing): I like this better than that. Alternatively, distance can be metaphorically

interpreted in terms of conceptual relations in the discourse model. For instance, consider example (S5). (S5) I bought an Integra yesterday. It's similar to the one I bought five years ago. That one was really nice, but I like this one even better. Here, that one refers to the Acura bought five years ago (greater temporal distance), whereas this one refers to the one bought yesterday (closer temporal distance).

One Anaphora: One-anaphora, exemplified in (S6), blends properties of definite and indefinite reference. (S6) I saw no less than 6 Acura Integras today. Now I want one. This use of one can be roughly paraphrased by one of them, in which them refers to a plural referent (or generic one, as in the case of (S6), see below), and one selects a member from this set (Webber, 1983). Thus, one may evoke a new entity into the discourse model, but it is necessarily dependent on an existing referent for the description of this new entity. This use of one should be distinguished from the formal, non-specific pronoun usage in (S6), and its meaning as the number one in (S7). (S6) One shouldn't pay more than twenty thousand dollars for an Acura. (S7) John has two Acuras, but I only have one.

Inferrables: Now that we have described several types of referring expressions, we now turn our attention to a few interesting types of referents that complicate the reference resolution problem. First, we consider cases in which a referring expression does not refer to an entity that has been explicitly evoked in the text, but instead one that is inferentially related to an evoked entity. Such referents are called inferrables (Haviland and Clark, 1974; Prince, 1981). Consider the expressions a door and the engine in sentence (S8). (S8) I almost bought an Acura Integra today, but a door had a dent and the engine seemed noisy. The indefinite noun phrase a door would normally introduce a new door into the discourse context, but in this case the hearer is to infer something more: that it is not just any door, but one of the doors of the Integra. Similarly, the use of the definite noun phrase the engine normally presumes that an engine has been previously evoked or is otherwise uniquely identifiable. Here, no engine has been explicitly mentioned, but the hearer infers that the referent is the engine of the previously mentioned Integra. Inferrables can also specify the results of processes described by utterances in a discourse. Consider the possible follow-ons (a-c) to sentence (S9) in the following recipe (from Webber and Baldwin (1992)): (S9) Mix the flour, butter, and water. a. Knead the dough until smooth and shiny. b. Spread the paste over the blueberries. c. Stir the batter until all lumps are gone. Any of the expressions the dough (a solid), the batter (a liquid), and the paste (somewhere in between) can be used to refer to the

result of the actions described in the first sentence, but all imply different properties of this result.

Discontinuous Sets: In some cases, references using plural referring expressions like they and them (see page 10) refer to sets of entities that are evoked together, for instance, using another plural expression (their Acuras) or a conjoined noun phrase (John and Mary): (S10) John and Mary love their Acuras. They drive them all the time. However, plural references may also refer to sets of entities that have been evoked by discontinuous phrases in the text: (S11) John has an Acura, and Mary has a Mazda. They drive them all the time. Here, they refers to John and Mary, and likewise them refers to the Acura and the Mazda. Note also that the second sentence in this case will generally receive what is called a pairwise or respectively reading, in which John drives the Acura and Mary drives the Mazda, as opposed to the reading in which they both drive both cars.

Generics: Making the reference problem even more complicated is the existence of generic reference. Consider example (S12). (S12) I saw no less than 6 Acura Integras today. They are the coolest cars. Here, the most natural reading is not the one in which they refers to the particular 6 Integras mentioned in the first sentence, but instead to the class of Integras in general

4. Syntactic and Semantic Constraints on Coreference

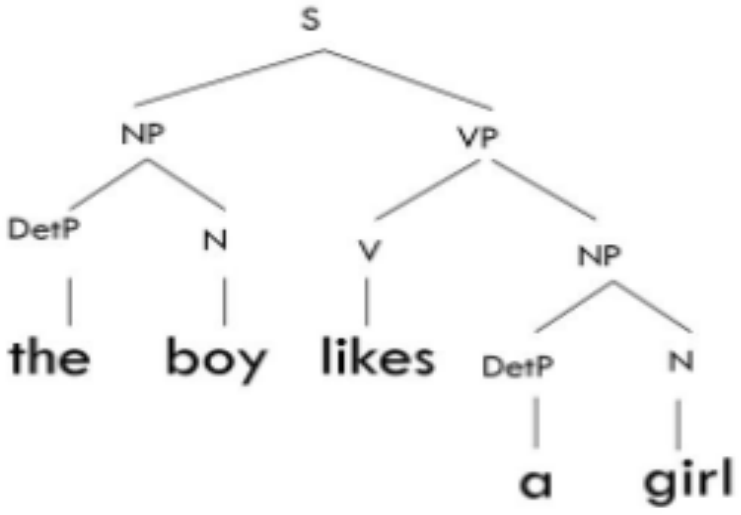
Q16 Differentiate between closed classes and open classes with example.

Open class	Closed class
1)open class are are those that not have relatively fixed membership.	1)closed class are those that have relatively fixed membership.
2)open class also called as content word.	2)closed class also called as function words.
3)open class include noun ,verb, adjective, adverb	3)closed class include pronoun, article, preposition, auxiliary, conjunction
4)This word are infinite	4)This word are finite in number
5) They are mutually inclusive	5) They are mutually exclusive
6) eg Noun-word,girl Adverb- Now , There	6) eg Conjuction- And , or ,But Preposition – at , in , an

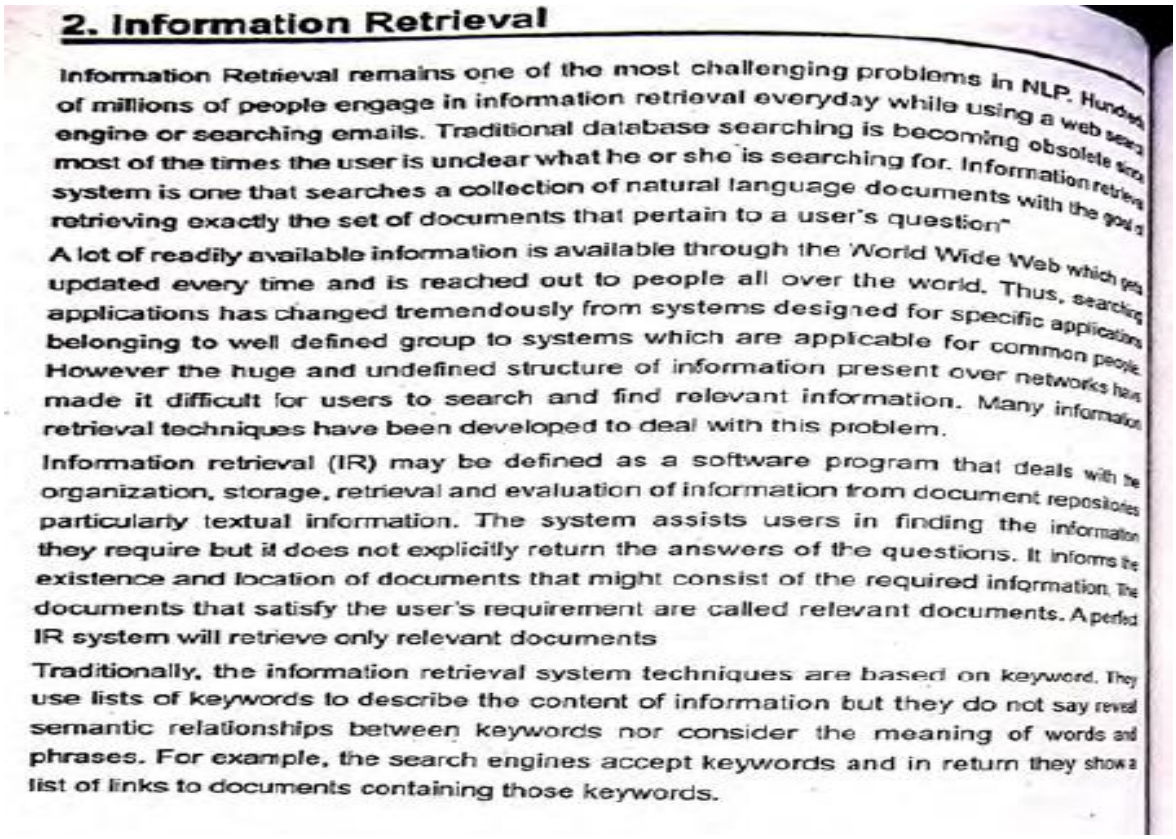
Q17Show derivation of "The boy likes a girl" in parse tree, consider following grammar rule:

S→NP VP
VP →Verb NP
NP→ Det NOM
NOM → Noun
Noun→ boy | girl
Verb → sees | likes
Adj → big | small
Adv → very
Det → a | the

Ans



Q18 What is information retrieval and machine translation in applications? Give brief answer on both.



2. Information Retrieval

Information Retrieval remains one of the most challenging problems in NLP. Hundreds of millions of people engage in information retrieval everyday while using a web search engine or searching emails. Traditional database searching is becoming obsolete since most of the times the user is unclear what he or she is searching for. Information retrieval system is one that searches a collection of natural language documents with the goal of retrieving exactly the set of documents that pertain to a user's question"

A lot of readily available information is available through the World Wide Web which gets updated every time and is reached out to people all over the world. Thus, searching applications has changed tremendously from systems designed for specific applications belonging to well defined group to systems which are applicable for common people. However the huge and undefined structure of information present over networks have made it difficult for users to search and find relevant information. Many information retrieval techniques have been developed to deal with this problem.

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 assists users in finding the information they require but it does not explicitly return the answers of the questions. It informs the existence and location of documents that might consist of the required information. The documents that satisfy the user's requirement are called relevant documents. A perfect IR system will retrieve only relevant documents

Traditionally, the information retrieval system techniques are based on keyword. They use lists of keywords to describe the content of information but they do not say reveal semantic relationships between keywords nor consider the meaning of words and phrases. For example, the search engines accept keywords and in return they show a list of links to documents containing those keywords.

Machine Translation (MT) is a standard name for computerized systems responsible for the production of translations from one natural language into another with or without human assistance. It is a subfield of computational linguistics that investigates the use of computer software to translate text or speech from one language to another. Development of a fully-fledged bilingual machine translation (MT) system for any two natural languages with limited electronic resources and tools is challenging and demanding task. In order to achieve reasonable translation quality in open source tasks, corpus-based machine translation approaches require large number of parallel corpora that are not always available, especially for less resourced language pairs. On the other hand, the rule based machine translation process is extremely difficult and fails to analyze accurately a large corpus of unrestricted text. Even though there has been an effort towards building English to Indian Language and Indian Language to Indian Language translation system, we do not have an efficient translation system as of today.

Why should we interest in using computers for translation at all? The first most important reason is that there is too much that needs to be translated and human translator cannot cope. A second reason is that technical materials are too boring for human translators, they don't like translating them, so they will look for help from computers. Another reason is that, in corporation area, there is a major requirement that terminology is used consistently; Computers are consistent but human translator tend to seek variety; they do not like to repeat the same translation and it will be not good for technical translation. Computer based translation can increase the volume and speed of translation and corporate area like to have translations immediately, the next day or even the same day.

Q19 Discuss various challenges in processing natural language.

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 understand 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. Al 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.

Q20 What is the role of FSA in Morphological analysis?

4. Finite Automata

The regular expression is more than just a convenient metalanguage for text searching. First, a regular expression is one way of describing a finite-state automaton (FSA). Finite-state automata are the theoretical foundation of a good deal of the computational work we will describe in this book. Any FSA regular expression can be implemented as a finite-state automaton (except regular expressions that use the memory feature; more on this later). Symmetrically, any finite-state automaton can be described with a regular expression. Second, a regular expression is one way of characterizing a particular kind of formal language called a regular language. Both regular expressions and finite-state automata can be used to describe regular languages. The relation among these three theoretical constructions is sketched out in the figure below.

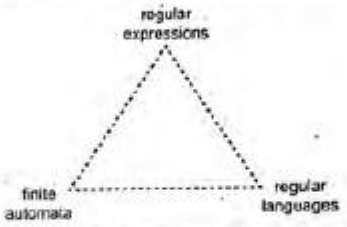


Figure 2: The relationship between finite automata, regular expressions, and regular languages

A formal language is completely determined by the 'words in the dictionary', rather than by any grammatical rules

A (formal) ***language*** L over alphabet Σ is just a set of strings in Σ^*.

Thus any subset $L \subseteq \Sigma^*$ determines a language over Σ

The ***languaage determined by a regular expression*** r over Σ is

$$L(r) \underline{\text{def}} \{u \in \Sigma^* \mid u \text{ matches } r\}.$$

Two regular expressions *r* and *s* (over the same alphabet) are ***equivalent*** iff *L*(r) and *L*(s) are equal sets (i.e. have exactly the same members.)

A finite automaton has a finite set of states with which it accepts or rejects strings.

Finite State Automata (FSA) can be:

Deterministic

On each input there is one and only one state to which the automaton can transition from its current state

Nondeterministic

An automaton can be in several states at once

Deterministic finite state automaton

1. A finite set of states, often denoted Q
2. A finite set of input symbols, often denoted Σ
3. A transition function that takes as arguments a state and an input symbol and returns a state. The transition function is commonly denoted δ If q is a state and a is a symbol, then $\delta(q, a)$ is a state p (and in the graph that represents the automaton there is an arc from q to p labeled a)
4. A start state, one of the states in Q
5. A set of final or accepting states F ($F \subseteq Q$)

A DFA is a tuple $A = (Q, \Sigma, \delta, q_0, F)$

Other notations for DFAs

Transition diagrams

- Each state is a node
- For each state $q \in Q$ and each symbol $a \in \Sigma$, let $\delta(q, a) = p$
- Then the transition diagram has an arc from q to p, labeled a
- There is an arrow to the start state q_0
- Nodes corresponding to final states are marked with doubled circle

Transition tables

- Tabular representation of a function
- The rows correspond to the states and the columns to the input s
- The entry for the row corresponding to state q and the column corresponding to input a is the state $\delta(q, a)$

$A = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_1\})$

where the transition function δ is given by the table

Q21 What is WordNet? How is "sense" defined in WordNet? Explain with example.

WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. The resulting network of meaningfully related words and concepts can be navigated with the browser. WordNet is also freely and publicly available for download. WordNet's structure makes it a useful tool for computational linguistics and natural language processing.

WordNet superficially resembles a thesaurus, in that it groups words together based on their meanings. However, there are some important distinctions. First, WordNet interlinks not just word forms—strings of letters—but specific senses of words. As a result, words that are found in close proximity to one another in the network are semantically disambiguated. Second, WordNet labels the semantic relations among words, whereas the groupings of words in a thesaurus does not follow any explicit pattern other than meaning similarity.

occurrence.

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 words in a particular context. Lexical ambiguity, syntactic or semantic, is one of the very first problems that any NLP system faces. Part-of-speech (POS) taggers with high level of accuracy can solve Word's syntactic ambiguity. On the other hand, the problem of resolving semantic ambiguity is called WSD (word sense disambiguation). Resolving semantic ambiguity is harder than resolving syntactic ambiguity.

For example, consider the two examples of the distinct sense that exist for the word "bank" –

The bank will not be accepting cash on Saturdays.

The river overflowed the bank.

The occurrence of the word **bank** clearly denotes the distinct meaning. In the first sentence, it means commercial (finance) banks, while in the second sentence; it refers to the river bank. Hence, if it would be disambiguated by WSD then the correct meaning to the above sentences can be assigned as follows –

*The **bank/financial institution** will not be accepting cash on Saturdays.*

*The river overflowed the **bank/riverfront.***

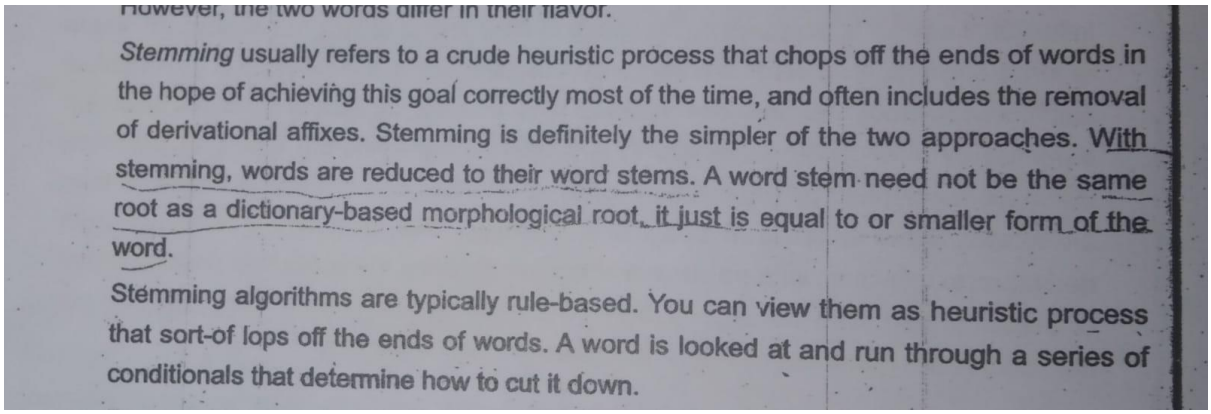
The evaluation of WSD requires the following two inputs:

A Dictionary: The very first input for evaluation of WSD is dictionary, which is used to specify the senses to be disambiguated.

Test Corpus: Another input required by WSD is the high-annotated test corpus that has the target or correct-senses. The test corpora can be of two types:

- **Lexical sample** - This kind of corpora is used in the system, where it is required to disambiguate a small sample of words.
- **All-words** - This kind of corpora is used in the system, where it is expected to disambiguate all the words in a piece of running text.

Q22 What do you mean by stemming? Explain Porter's stemming algorithm in detail.



However, the two words differ in their flavor.

Stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes. Stemming is definitely the simpler of the two approaches. With stemming, words are reduced to their word stems. A word stem need not be the same root as a dictionary-based morphological root, it just is equal to or smaller form of the word.

Stemming algorithms are typically rule-based. You can view them as heuristic process that sort-of lops off the ends of words. A word is looked at and run through a series of conditionals that determine how to cut it down.

Stemming Algorithms Examples

Porter stemmer: This stemming algorithm is an older one. It's from the 1980s and its main concern is removing the common endings to words so that they can be resolved to a common form. It's not too complex and development on it is frozen. Typically, it's a nice starting basic stemmer, but it's not really advised to use it for any production/ complex application. Instead, it has its place in research as a nice, basic stemming algorithm that can guarantee reproducibility. It also is a very gentle stemming algorithm when compared to others.

This algorithm is also known as the Porter2 stemming algorithm. It is

Q23 How HMM is used for POS tagging? Explain in detail.

Hidden Markov Model (HMM) POS Tagging

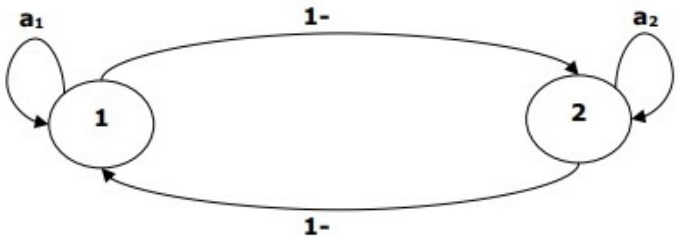
Before digging deep into HMM POS tagging, we must understand the concept of Hidden Markov Model (HMM).

Hidden Markov Model

An HMM model may be defined as the doubly-embedded stochastic model, where the underlying stochastic process is hidden. This hidden stochastic process can only be observed through another set of stochastic processes that produces the sequence of observations.

Example

For example, a sequence of hidden coin tossing experiments is done and we see only the observation sequence consisting of heads and tails. The actual details of the process - how many coins used, the order in which they are selected - are hidden from us. By observing this sequence of heads and tails, we can build several HMMs to explain the sequence. Following is one form of Hidden Markov Model for this problem −



P(H) =P_1	P(H) =P_2
P(T) =1-P_1	P(T) =1-P_2

We assumed that there are two states in the HMM and each of the state corresponds to the selection of different biased coin. Following matrix gives the state transition probabilities −

$$A = \begin{bmatrix} a11 & a12 \\ a21 & a22 \end{bmatrix}$$

Here,

- $\mathbf{a_{ij}}$ = probability of transition from one state to another from i to j.
- $\mathbf{a_{11}} + \mathbf{a_{12}} = 1$ and $a_{21} + a_{22} = 1$
- $\mathbf{P_1}$ = probability of heads of the first coin i.e. the bias of the first coin.
- $\mathbf{P_2}$ = probability of heads of the second coin i.e. the bias of the second coin.

We can also create an HMM model assuming that there are 3 coins or more.

This way, we can characterize HMM by the following elements −

- N, the number of states in the model (in the above example N =2, only two states).
- M, the number of distinct observations that can appear with each state in the above example M = 2, i.e., H or T).
- A, the state transition probability distribution − the matrix A in the above example.
- P, the probability distribution of the observable symbols in each state (in our

- I, the initial state distribution.

Use of HMM for POS Tagging

The POS tagging process is the process of finding the sequence of tags which is most likely to have generated a given word sequence. We can model this POS process by using a Hidden Markov Model (HMM), where **tags** are the **hidden states** that produced the **observable output,** i.e., the **words**.

Mathematically, in POS tagging, we are always interested in finding a tag sequence (C) which maximizes −

P (C|W)

Where,

C = C_1, C_2, C_3... C_T

W = W_1, W_2, W_3, W_T

On the other side of coin, the fact is that we need a lot of statistical data to reasonably estimate such kind of sequences. However, to simplify the problem, we can apply some mathematical transformations along with some assumptions.

The use of HMM to do a POS tagging is a special case of Bayesian interference. Hence, we will start by restating the problem using Bayes' rule, which says that the above-mentioned conditional probability is equal to −

(PROB (C_1,..., CT) * PROB (W_1,..., WT | C_1,..., CT)) / PROB (W_1,..., WT)

We can eliminate the denominator in all these cases because we are interested in finding the sequence C which maximizes the above value. This will not affect our answer. Now, our problem reduces to finding the sequence C that maximizes −

PROB (C_1,..., CT) * PROB (W_1,..., WT | C_1,..., CT) **(1)**

Even after reducing the problem in the above expression, it would require large amount of data. We can make reasonable independence assumptions about the two probabilities in the above expression to overcome the problem.

First Assumption

The probability of a tag depends on the previous one (bigram model) or previous two (trigram model) or previous n tags (n-gram model) which, mathematically, can be explained as follows −

PROB (C_1,..., C_T) = $\Pi_{i=1..T}$ PROB ($C_i|C_{i-n+1}...C_{i-1}$) **(n-gram model)**

PROB (C_1,..., CT) = $\Pi_{i=1..T}$ PROB ($C_i|C_{i-1}$) **(bigram model)**

The beginning of a sentence can be accounted for by assuming an initial probability for each tag.

PROB ($C_1|C_0$) = PROB $_{initial}$ (C_1)

Second Assumption

The second probability in equation (1) above can be approximated by assuming that a word appears in a category independent of the words in the preceding or succeeding categories which can be explained mathematically as follows −

PROB (W_1,..., W_T | C_1,..., C_T) = $\Pi_{i=1..T}$ PROB ($W_i|C_i$)

Now, on the basis of the above two assumptions, our goal reduces to finding a sequence C which maximizes

$\Pi_{i=1...T}$ PROB($C_i|C_{i-1}$) * PROB($W_i|C_i$)

Now the question that arises here is has converting the problem to the above form really helped us. The answer is - yes, it has. If we have a large tagged corpus, then the two probabilities in the above formula can be calculated as −

PROB ($C_{i=VERB}|C_{i-1=NOUN}$) = (# of instances where Verb follows Noun) / (# of instances where Noun appears) (2)

PROB ($W_i|C_i$) = (# of instances where W_i appears in C_i) /(# of instances where C_i appears) (3)

Q24 Explain use of CFG in Natural Language Processing with suitable example.

Ans:

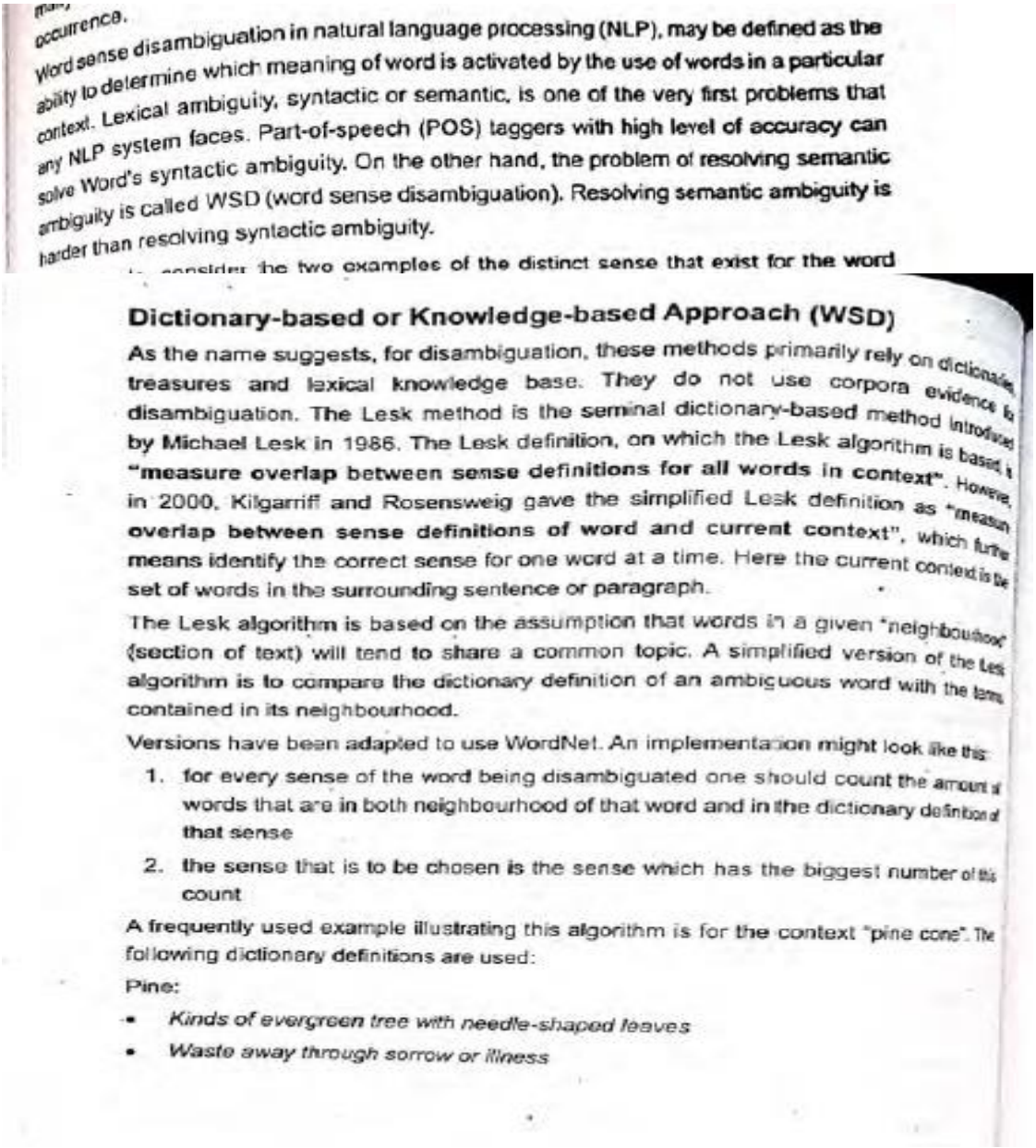
CFG:

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.

Q25 Compare Information Retrieval with Information Extraction system.

	Information Retrieval	Information Extraction
1.	Document Retrieval	Feature Retrieval
2.	Return set of relevant documents	Return facts out of documents
3.	The goal is to find documents that are relevant to the user's information need	The goal is to extract pre-specified features from documents or display information.
4.	Real information is buried inside documents	Extract information from within the documents
5.	The long listing of documents	Aggregate over the entire set
6.	Used in many search engines – Google is the best IR system for the web.	Used in database systems to enter extracted features automatically.
7.	Typically uses a bag of words model of the source text.	Typically based on some form of semantic analysis of the source text.
8.	Mostly use the theory of information, probability, and statistics.	Emerged from research into rule-based systems.

Q26 What is Word Sense Disambiguation? Illustrate with example how Dictionary-based approach identifies correct sense of an ambiguous word.



occurrence.

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 words in a particular context. Lexical ambiguity, syntactic or semantic, is one of the very first problems that any NLP system faces. Part-of-speech (POS) taggers with high level of accuracy can solve Word's syntactic ambiguity. On the other hand, the problem of resolving semantic ambiguity is called WSD (word sense disambiguation). Resolving semantic ambiguity is harder than resolving syntactic ambiguity.

... consider the two examples of the distinct sense that exist for the word

Dictionary-based or Knowledge-based Approach (WSD)

As the name suggests, for disambiguation, these methods primarily rely on dictionaries, treasures and lexical knowledge base. They do not use corpora evidence for disambiguation. The Lesk method is the seminal dictionary-based method introduced by Michael Lesk in 1986. The Lesk definition, on which the Lesk algorithm is based is **"measure overlap between sense definitions for all words in context"**. 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. Here the current context is the set of words in the surrounding sentence or paragraph.

The Lesk algorithm is based on the assumption that words in a given "neighbourhood" (section of text) will tend to share a common topic. A simplified version of the Lesk algorithm is to compare the dictionary definition of an ambiguous word with the terms contained in its neighbourhood.

Versions have been adapted to use WordNet. An implementation might look like this:

1. for every sense of the word being disambiguated one should count the amount of words that are in both neighbourhood of that word and in the dictionary definition of that sense
2. the sense that is to be chosen is the sense which has the biggest number of this count

A frequently used example illustrating this algorithm is for the context "pine cone". The following dictionary definitions are used:

Pine:

- *Kinds of evergreen tree with needle-shaped leaves*
- *Waste away through sorrow or illness*

Q27 Discuss in detail any application considering any Indian regional language of your choice.

3. Question Answering System

Humans are always in a quest to extract information related to some topic or entity. Question answering system helps user to find the precise answer to the question articulated in natural language. Question answering system provides explicit, concise and accurate answer to user questions rather than providing a set of relevant documents or web pages as answers as most of the information retrieval system does.

Any question answering system basically consists of three parts as question processing, answer retrieval and answer generation. In question processing users natural language questions are parsed to formulate questions in machine readable form using different approaches. Then in answer retrieval candidate answers are extracted based on intermediate representation of question. Finally in answer generation phase user understandable precise and accurate answer is generated and provided to user.

Any question answering system can be classified into two main types: close domain QAS and open domain QAS. In close domain QAS scope of user question is limited to a particular domain like medicine, movies, history and others. So if a QAS is created for history for history domain it will provide answers to questions related to history only. An open domain QAS mostly works like search engines like Google and all where it provides explicit answers to question belonging to any domain. So in open domain QAS the scope for question is global.

Question in any question answering system can be of varying types. Question can be factoid question for which answers are simple fact about the entity in question. Some questions can be of descriptive type where one needs to full detail about a person, place or any event. There can be simple yes/no type of question which simple provides answers as yes or no. A question can also be an instruction-based question where answers are provided as an instruction to accomplish any task. Question in a QAS can be of many other forms which provide precise answer in the same format as that of question provided.

Example: Question Answering System for Hindi and Marathi language

The Question Answering System for Hindi and Marathi language is as shown in Figure 6.3. Here input to the system is natural language Hindi or Marathi Query. Input query is first tokenized to generate individual token and then these tokens undergo word grouping where two or three corresponding words are merged together if they are related with each other by using the available word grouped list. POS tagging is performed on word

grouped tokenized query text to extract relevant part of speech associated with the query text. POS tagged query text then passes through chunking process where noun and verb grouped present in the query text are extracted. Based on the extracted chunked groups initially query triples are extracted using Subject, Object and Verb (SOV). Then next process is to generate onto triples by fetching relevant onto words from ontology. Finally ontology is traversed to fetch relevant answer based on generate onto triples, if onto triple matches with any onto set in ontology then corresponding answer is fetched and passed to answer generation process to present the answer as natural way as possible mostly in the form of natural language text. The system for Hindi Question Answering System and Marathi Question Answering System both follow the same flow and have the same basic architecture with exception in Marathi that it requires separate case extraction phase, because case marker are mainly attached to noun words as suffixes, before extraction of SOV from query text as Marathi is much more inflected language compared to Hindi.

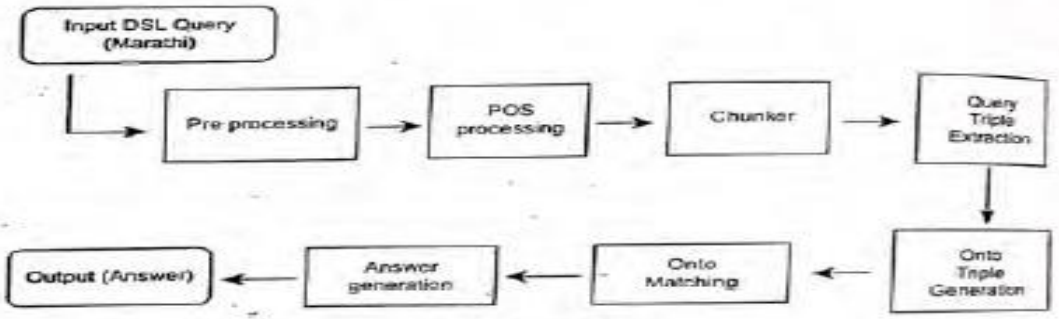


Figure 4: Basic QA system

The overall algorithm for proposed QA system is as follows:

Input: Natural language question in Hindi/Marathi language

Output: Answer in Hindi/Marathi language

Step 1: Tokenize the input question into word tokens.

Step 2: Grouped the correlated words into one merged word.

Step 3: Extract POS tag of each word in the tokenized list.

Step 4: Chunk the POS tagged words into noun and verb groups.

Step 5: Extract query triple from chunked grouped list.

Step 6: Generate onto triple.

Step 7: Traverse ontology to fetch answer.

Step 8: Formulate answer as natural language text.

Sample input and output for Hindi query:

Input Question:शिवाजी की माँ कौन थी?

Answer: शिवाजी की माँ जीजाबाई थी।

Sample input and output for Marathi query:

Input Question: शिवाजिची आई कोण होती?

Answer: शिवाजिची आई जीजाबाई होती.

The question answering system for Marathi natural language using concept of ontology as a formal representation of knowledge base for extracting answers. Ontology is used to express domain specific knowledge about semantic relations and restrictions in the given domains. The ontologies are developed with the help of domain experts and the query is analyzed both syntactically and semantically. The results obtained are accurate enough to satisfy the query raised by the user. The level of accuracy is enhanced since the query is analyzed semantically.

A question answering system is much more effective than traditional search engines as it provides accurate and precise answer to question rather than providing links to relevant documents or set of matching contents. Question answering system has become part of daily life of users, over a period of time many personal assistance software like Siri, Cortana, and Google Now are developed which provide precise and accurate answer to user question.