

NLP

Components of NLP

- (I) Natural Language Understanding:
 - Lexical Ambiguity (word level).
 - Syntactical " (sentence level).
 - Referential Ambiguity (referencing issue) using pronouns.

- (II) Natural Language Generation
 - Text planning (Retain Relevant content from KB).
 - Sentence planning (words, meaningful phrases, setting tone)
 - Text Relaxation (Mapping sentence planning units into sentence structure)

Steps in NLP

- ① Syntactic Analysis : we have to analyse the structure of words. The collection of words or phrases in lang is lexicon of lang.
- ② Syntactic Analysis : we use parsing for analysis of word. Although have to arrange words in a particular manner. That shows the relationship b/w words.
- ③ Semantic Analysis : It describes a dictionary meaning which is meaningful. In the task domain, mapping syntactic structure & object
- ④ Discourse Integration
The meaning of any sentence depends upon the meaning of previous sentence. In addition, it changes the meaning to immediately

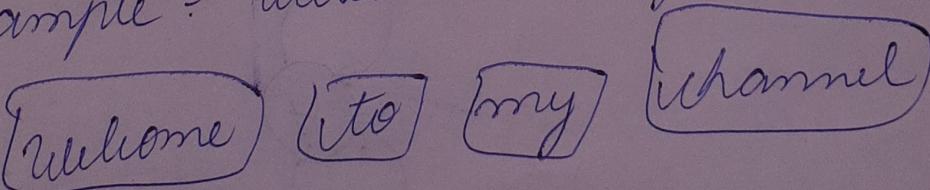
Succeeding sentence

⑤ Pragmatic Analysis

In this step, data is interpreted on what is actually meant. Although we have to derive aspects of lang which require real-world knowledge.

Steps in NLP

① Tokenization:

- Cutting big sentences into small tokens.
- Example : Welcome to my channel.


② Stemming:

- Normalize words into base or root forms

③ Lemmatization

- (3) lemmatization

 - * group together different uninflected forms of words called Lemma.
 - * somehow similar to stemming, as it maps several words into one common root.
 - * Output of lemmatization is a proper word.

ing: gone, going and went → Go

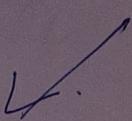
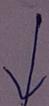
④ Pos tags

- PoS stands for Parts of Speech tags.
 - It indicates how a word functions in meaning as well as grammatically in a sentence.

waits

waited

waiting



Wait

③ Lemmatization

- * group together different uninflected forms of words called Lemma.
- * somehow similar to stemming, as it maps several words into one common root.
- * Output of lemmatization is a proper word.

e.g.: gone, going and went \rightarrow Go

④ POS tags

- POS stands for Part of Speech tags.
- It indicates how a word functions in meaning as well as grammatically in a sentence.

W DT NN VB DT NN.
↓ ↓ ↓ ↓ ↓
The cat killed the sat

eg: 'google' something on Internet.
(NN, VB)

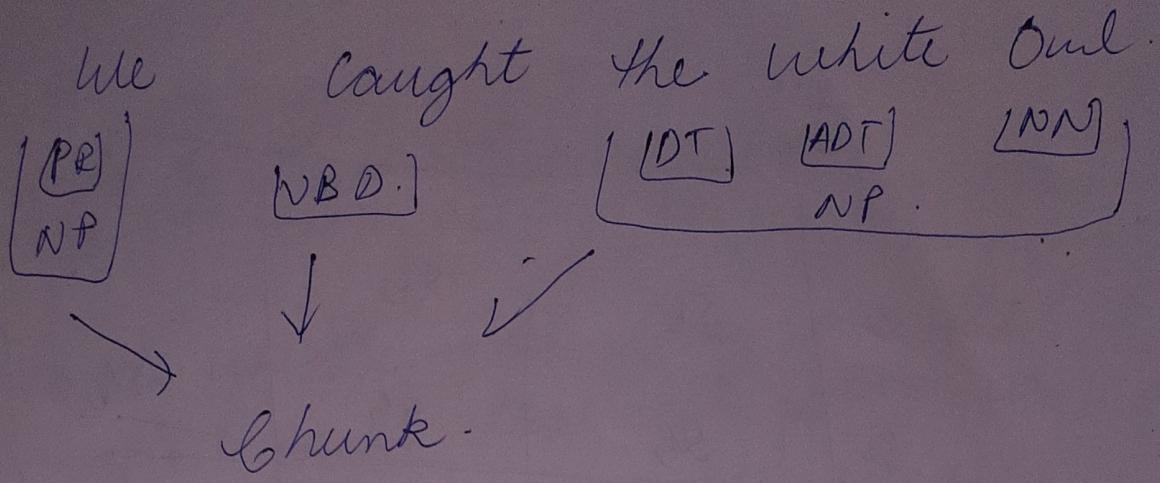
⑤ Name Entity Recognition

- It is a process of detecting name entity such as: person, Organisation, location, Quantities etc.

person
~o~s
eg: Google's CEO Sundar
yuhai introduced the new Pupil 3.
at New York Central Mall.
↓ location ↗ org

⑥ Chunking

Pulling individual pieces of info & grouping them into bigger pieces.



- This helps in getting insight & meaningful info from the text.

Applications of NLP

- Chat Box.
- Speech Recognition
- Machine translation
- Spell checking
- Keyword searching
- Advertisement matching

Words & their Components

- * Words in the most lang are the smallest unit that can form a complete utterance by themselves.
- * Three important items which are integral parts of words are:
 - (i) Phonemes: units of sound in spoken lang.
 - (ii) Graphemes: smallest unit of written lang.
 - (iii) Morphemes: minimal part.
- * In words & components we have:
Tokens, Lexemes, Morphemes, Syntax
- Tokens :-
 - * ~~Gutting big sentences into smaller words~~ ~~& encoding each word~~
 - * ~~This~~

Tokens

- * Tokenization is essentially splitting a phrase, sentence, paragraph or an entire text document into smaller units & encoding them.
- * Each of these smaller units are called tokens
- * eg Natural Language Processing

Natural Language Processing
0.01 0.02 0.03

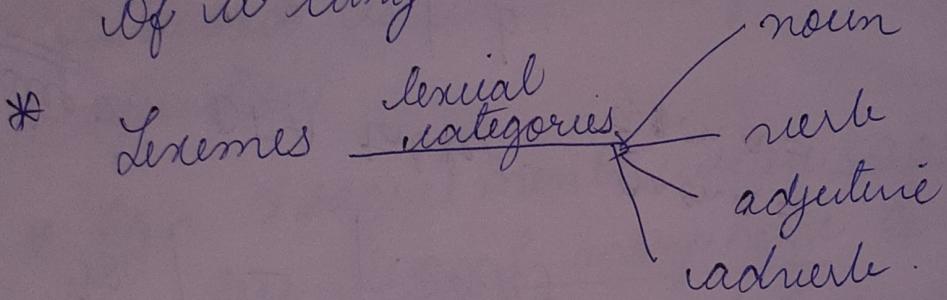
3 tokens

- * eg: I won't read the newspaper
- ① I will not read the Newspaper.
- * This type of analysis is called tokenization & normalization.

Lexemes

- * There are a set of alternative forms that can be expressed for a word. Such sets are called as lexemes or lexical items.

- * They Lexemes constitute to lexicon of a lang.



- * The inflection form of lexeme by which it is identified is called lemma.

- * The root word is called lemma in lemmatization process.

eg In case of receive as exception we derive a set of words from verb to receive.

Morphemes

- * The structural components that associate the properties of words forms are called morphs.
- * The morphs that by themselves represent some aspect of the meaning of a word is called morphemes of some function.
- * Eg: disagreements where lager is a free lexical morpheme & other elements ~~are~~ are bounded grammatical morphemes.
- * Morphs when they interact with each other undergo additional phonological & orthographical changes.
- * These alternative forms are called allomorphs
- * Ex: the past tense morphs, plural morphs

Typology

- * Morphological typology divides lang in groups.
 - * Isolating lang include no words that would comprise more than 1 morpheme.
 - * Synthetic lang can combine more morphemes in one word
 - * Comitative lang linking morphemes one after the other
 - * Non-linear lang allowing structural components to merge non-sequentially
-

Issues & Challenges

Irregularity

Ambiguity

Productivity

a) Irregularity:

- * Morphological parsing precludes generalisation & abstraction in the world of words.
- * In English the general past form comes by adding -ed or -t (accepted & built).
- * The irregular verbs in English tend to take different forms in the past or in present participle depending on the origin of word

by
blow
break
bring

Past tense
blew
broke
brought

Past Participle
blown
broken
brought

Ambiguity

- * words forms that look the same but have distinct functions or meanings are called homonyms (example: kind, ring, night, nose).
- * Ambiguity is present in all aspects of morphological processing.
- * Inverting of Sandhi during tokenization can provide multiple solutions to the problem of Indolem lang (na vasatoh vidyate bharah) which means the unreal has no existence)

Productivity

- * An important question to be answered - Is the inventory of words in a language finite or infinite?
- * In one view lang can be seen as simply a collection of utterances actually pronounced or written.

- * This data set can be linguistic corpora - a finite collection of linguistic data.
- * If we consider language as a system, we discover structural devices like revision, iteration that allow to produce an infinite set of concrete linguistic utterances
- * This process is called morphological productivity.
- * Eg Autowiki "googol" is a made-up word denoting a no "one followed by hundred zeros".
- * The name of the company google is virtually a misspelled word.
- * Now both of these words entered the English lexicon.

Q) Parsing Natural Languages

~~(eg1)~~ He wanted to go for dine in movie
He wanted to go for dine in the country

* There is a natural pause between
dine & in in the 2nd sentence

* The gap reflects an underlying
hidden structure

* Parsing provides a structural description
that identifies such a break in the
intonation

~~(eg2)~~ The cat who lies dangerously
had nine lies.

* A text to speech system needs to
know that the first instance of a
word is noun & the second is noun.

* This is an instance of POS tagging problem.

* Another important application where
parsing is important is text summarization

(ir3)

Open borders imply increasingly
cultural fragmentation in
EUROPEAN COUNTRIES.

* In the above sentence, the words
capitalized phrase can be replaced
with other phrases without
changing the meaning of sentence.

* Open borders imply increasingly
cultural fragmentation in
the countries of Europe

* " " " "
" " " "

European countries

* " " " " "
" " " "

Europe

* In NLP, Syntactic parsing is used in
many apps like

- * Statistical Machine Translation
- * Information extraction from text collection
- * Language summarization
- * Error correction in text
- * Knowledge acquisition from language

Q2)) Treebanks - A Data Driven Approach to Syntax.

- * Parser requires info that is not explicit to the input sentence.
- * Parser requires some additional knowledge beyond the input sentence that should be produced as output.
- * We can write down rules of the syntax of a sentence as CFG.
- * Here we have CFG which represents a simple grammar of itemstine rules in English.

(Ans)

$$S \rightarrow NP VP$$

$$NP \rightarrow 'John' \mid 'pockets' \mid DN \mid NP PP$$

$$VP \rightarrow VNP \mid VP PP$$

$$V \rightarrow brought$$

$$D \rightarrow a$$

$$N \rightarrow 'shirt'$$

$$PP \rightarrow P NP$$

$$P \rightarrow 'with'$$

- * The above CFG can produce the syntax analysis of sentence like John brought a shirt with pockets.

* Parsing the previous sentences gives

(S (NP John))	(S (NP John))
(VP (VP (V brought)))	(VP (V brought))
(NP (D a))	(NP (NP (D a)))
(N shirt))	(N shirt))
(PP (P with))	(PP (P with))
(NP (pocket)))	(NP (pocket))))

- * Writing a CFG for syntactic analysis of NLP is problematic.
- * A simple list of rules does not consider interactions between different components in the grammar.
- * Listing all possible syntactic constellations in a language is difficult task.
- * It is difficult to exhaustively list the lexical properties of words.
- * This is a typical knowledge acquisition problem.
- * Other problem is that rules interact with each other in nonterminatorily explosive ways.

(end) \Rightarrow Noun phrase has clunky branching tree

$N \rightarrow NN$ (Rewriting Rule).

$N \rightarrow 'natural' \mid language \mid 'processing' \mid 'book'$

- * For input 'natural language processing' the rewriting rules produce two ambiguous parsers.

$(N(N(N(N(natural))))$
 $(N(language)))$
 $(N(processing)))$

$(N(N(natural)))$
 $(N(N(language)))$
 $(N(processing)))$

- * For CFG's it can be proved that the no. of parsers obtained by using rewriting rule in ities is $\frac{1}{n+1} \binom{2n}{n}$.
 Catalan number ($1, 1, 2, 5, 14, 42, 132, 429, 1430$) of n .

$$cat(n) = \frac{1}{n+1} \binom{2n}{n}$$

- * For the input " natural language processing book" only one of the 2 parsers obtained by using CFG is correct.

$(N(N(N(N(N(natural))))$
 $(N(language))))$,
 $(N(processing))))$,
 $(N(book))))$.

- * This is the second knowledge acquisition problem.
- * We not only need to know the rules but also analysis.
- * The construction of a treebank is a data driven approach to syntax analysis
- * It allows us to address both the knowledge acquisition bottlenecks in one stroke
- * A treebank is a collection of sentences.
- * Where each sentence is provided by complex syntax analysis.
- * The syntax analysis for each sentence has been judged by a human expert.
- * A set of annotation guidelines is written before the annotation process to ensure
- * Treebank contains annotations of syntactic structure of large set of sentences.
- * Supervised ML techniques are used to train the parser from the training data extracted from the banks.
- * The first knowledge acquisition problem is addressed by providing syntactic analysis directly instead of grammar.

- * The second knowledge acquisition problem is solved as for each sentence in a ctrebank the most feasible syntactic analysis is provided.
- * Using supervised ML techniques are used to learn scoring functions for all possible syntax analysis

Q3 Representation of Syntactic Structure

(Q3)
Syntax analysis using dependency graphs.

- * Dependency graphs connects head of a phrase to its dependents.
- * Definition: In dependency graph
 - nodes are words of input sentence.
 - edges are binary relations from head to dependent.
- * It is often assumed that all words except one have a syntactic head.
- * It means graph will have a tree with single independent node as root
- * In labelled dependency parsing the parser assigns a specific type to each dependency relation holding the head word as dependent word

* In dependency trees, 0 index is used to indicate the root symbol

[FSB0]

[Vsak, JE, 8] [ZIP, II.]

but

(naji, VPP3A,2)

have

[ZIP, 6]

[zajem N45]

interest

[chylu, VPP3A,9]

miss.

[student n1,1]

student

[O, RY13]

in

[fakulte N3,7]

faculty

[azj/N10]

teacher

engh

[dayky, NPA4A,1A]

languages.

* This is an example of a dependency graph syntax analysis for a Czech sentence taken from Prague Dependency Treebank.

* Each node in the graph is a word, its parts of speech & not the word in sentence.

* Ex: (fakulte, N3,7) is the 2nd word in the sentence with Part of speech N3 which

St. Mary's

who tells us that it has dative case.

- * The node [ZSB, 0] is the root node in dependency tree.
- * The English equivalent is provided to each node.
- * It is observed that dependency analysis make minimal assumption about syntactic structure for avoiding annotation of hidden structure like empty elements to represent missing arguments of predicate.

The dia: The students are interested in languages, but the faculty is missing structures of English.

Projectivity

- * Projectivity is the constraint on semantic analysis due to effect of linear order of words on dependencies b/w words.
- * The dia shows the English sentence with requirements of missing dependencies.

root This saw a dog yesterday which was fed.

Parsing Algo

4) Shift reduce parser

2 Data structure stores:

* Stack - symbols of grammar.

* Input buffer - Input string

* \$ specifies bottom of the stack & end of input buffer

* Actions of shift reduce parser

- shift
- reduce
- accept
- error

* There are 2 types of conflicts:

- Shift reduce conflict
- reduce reduce conflict

(Ex)

$$E \rightarrow E + T \mid T$$

$$T \rightarrow T * F \mid F$$

$$F \rightarrow (E) \mid id$$

$$id \# id$$

Stack
\$
\$ id.
\$ F
\$ T
\$ E id T #
\$ T # id.

Input buffer
id # id \$.
* id \$.
* id \$.
* id \$.
id \$.
\$

Action
Shift
Reduce by $F \rightarrow id$.
Reduce $T \rightarrow F$.
Shift
Shift
Reduce $F \rightarrow id$.

\$T*F	\$	Reduce T → T*F
\$T	\$.	Reduce T → E
\$E.	<u>\$.</u>	Accepted

Shift-reducing parsing algorithm

- * Start with an empty stack.
- * The buffer must contain the input string.
- * Exit with success if,
 - the top of the stack contains start symbol of the grammar.
 - and if the buffer is empty.
- * Choose b/w the following 2 steps.
 - * If choice is ambiguous, choose one based on an oracle.
 - * Shift a symbol from the buffer onto the stack.
- * If the top k symbols of the stack are $\alpha_1, \dots, \alpha_k$ then replace the top k symbols with the left hand side of non-terminal ' A '.
- * Exit the failure if no action can be taken in previous steps.

Endian

Hyper graphs to chart (pausing OR) CYK Algo.

- * It is a membership algo.
- * It checks whether a string 'abbb' is a valid member of full CGF.
- * CYK is applicable on CNF only.

$$\boxed{\text{CNF: } A \rightarrow BC \text{ (00)} \\ A \rightarrow a}$$

- * It is universal. Applicable on all grammar.
- * Time complexity = $O(n^3)$.
- * Space complexity = $O(n^2)$

for abbb.
1 2 3 4.

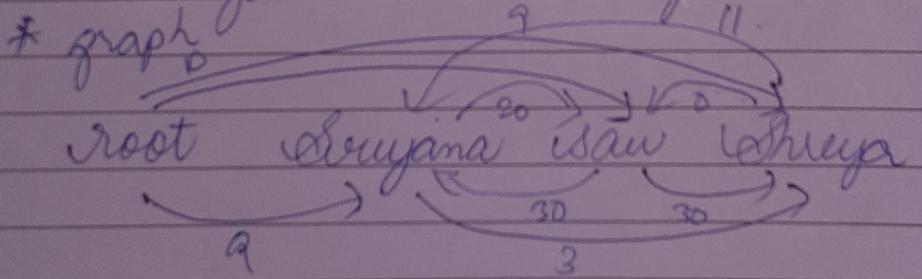
$$\left\{ \begin{array}{l} S \rightarrow AB. \\ A \rightarrow BB/a \\ B \rightarrow AB/b. \end{array} \right.$$

to make 4×4 matrix
Unit half + Rough

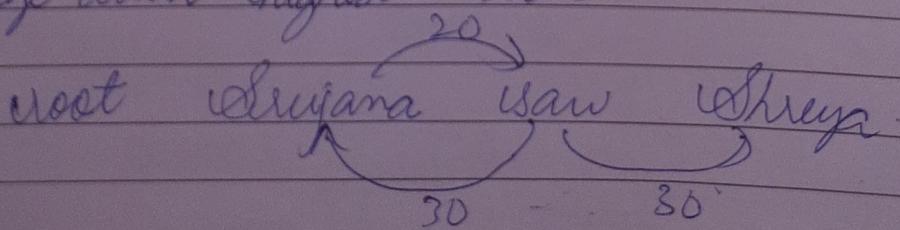
	4	3	2	1	2, 4.	2, 4.	1, 2.	1, 2.	1, 2.
1	S, B.	S, B, A.	S, B.	A.	2, 4.	2, 4.	B, B.	(11) (22)	A, B.
2	S, B.	A.	B.		S, B.	B.	A.		
3	A.	B.			1, 4.	(13)			
4	B.				A, B.	1, 3, 3	2, 3		
							A, B.	(22) (33)	B, B.
							S, B.		A.

6). Minimum Spanning Tree

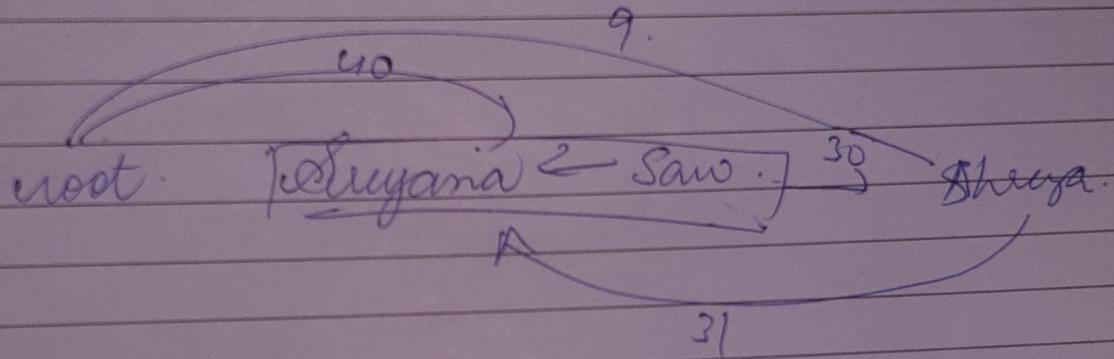
- * The minimum spanning tree corresponds to the optimum branching problem in directed graphs which have roots and does not have cycles.
- * The basic prerequisite is all the dependency links b/w the words must have score.
- * Consider input string as Srujan saw Shreya



- * The scoring function is used to assign weight to edges.
- * The algo starts finding the incoming edge with highest score.



Root \rightarrow Saw \rightarrow Shreyaa $= 10 + 30 = 40$.
 Root \rightarrow Shreyaa \rightarrow Saw $= 9 + 20 = 29$.
 Shreyaa \rightarrow saw - Shreyaa $0 + 30 = 30$.
 Shreyaa \rightarrow Deruka - Saw $11 + 20 = 31$.

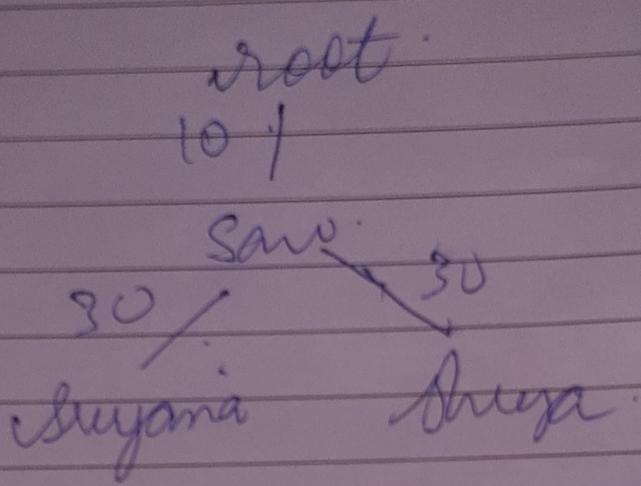


- * MST is applied iteratively.
- * Best incoming edges to each word is fond.
- * In next iteration

Root \rightarrow Shreyaa \rightarrow [Shreyaa \rightarrow Saw] $9 + 31 = 40$.
 Root \rightarrow [Shreyaa - Saw] \rightarrow Shreyaa will have $40 + 30 = 70$.

the best nodes chosen are
 Root [Shreyaa - Saw] \rightarrow Shreyaa

and we get highest scoring 41
dependency parse as



Q2) Models for Ambiguity Resolution in Parsing.

- a) Probabilistic Context Free Grammar.
- b) Generative Models for parsing
- c) Discriminative Models for Parsing

Q8) Semantic parsing

a) Intro

- * Its an imp phase in NLP.
- * Any document is comprised of sets of sentences.
- * These sentences are formed by the words arranged according to the rule based grammar as per constituents in a particular language.
- * To extract meaning out of this syntactically arranged sentences is the object of semantic parsing.
- * Parsing is analysing a text into logical syntactic components.
- * And semantics is study of meaning.
- * The information pieces are identified and related in semantic parsing.
- * The meaningful parts are identified in the text.
- * Then they are transformed into suitable data structure for higher level task accomplishment.

b) Types of Semantic parsing:

- * Semantic parsing is about identifying the meaning.
- * There are 2 approaches

①

- * Consider domain specific applications like travel reservation, gaming simulation etc.
- * Such meaning extraction can be done based on restricted clients of particular domain.
- * In this approach based on query, output is generated from the meaning representation.
- * This approach is known as deep semantic parsing.
- * little or no scope of reusability.

②

- * The second approach is not domain specific.
- * It is more generic.
- * In this approach, the task of meaning representation is divided into small parts.
- * It is known as shallow semantic parsing.
- * These species are responsible for capturing small manageable components that represent meaning.
- * High reusability.

c)

Semantic Interpretation

- * Semantic Interpretation is joining of different components
- * The components define the meaning representation of text.
- * This representation when fed to computer can be further processed for computational manipulations
- * Key search for any applications.
- * Semantic Parsing is the part of Semantic interpretation

~~Components of Semantic interpretation~~

Following are the major components in this process.

① Structural Ambiguity:

- * Any sentence is represented by its syntactic structure.
- * Syntax and semantics are closely related to each other.
- * we consider that the semantic interpretation is based on underlying syntactic structure
- * So syntactic processing is the first step in semantic interpretation

② Word Sense

* Many times in any language a same word is used in different meaning depending upon world knowledge

* Eg:- word bank represents a money bank or it can also means river bank.

* Due to inherent intelligence & language vocabulary by humans, it is not a difficult for them to understand the meaning of word inputted by speaker or author.

* Consider the foll examples

(a) go to bank to withdraw money.

(b) fetch some water from river bank.

* It's easy task for humans to disambiguate the meaning of word bank in alone sentences.

* But for machine this word sense disambiguation is a challenging task & it plays an important role in semantic interpretation.

* But for machine,

③ Entity and Entity Resolution

* Any text consists of various entities.

falling in different categories like person name, locations, quantities etc

- * Identification of these entities in the major task in semantic interpretation system

- * Named Entity System (NER): is a subtask of information extraction for locating & classifying named entities.

- * Ex- [Shreya] person bought 100 shares of [Infosys] person [2022] time.

- * Another important task is of no-reference resolution.

- * No-reference occurs when two or more expressions in text refer to same person.

- * Ex- Shreya said she will sing.

In this sentence, proper noun Shreya & She

- * This task is also under information extraction

- * And it is a major component in semantic interpretation

④ Predicate Argument Structure

- * After finishing all the above mentioned tasks of word sense disambiguation, NER no-reference sol.
- * Next task is to identify what is a

role of entity in particular event

- * This is known as resolving argument structure of predicate in a sentence.
- * It is typically an identification of who did what to whom, how where, when etc.

(3) Meaning Representation :-

- * This is the last step in semantic interpretation.
- * Also called deep representation.
- * It involves building meaning representation.
- * which can be used by algorithms for various applications.
- * Research is still going on in the area of general purpose.
- * Much study has been done in domain specific apps.
- * In : Consider a Gequery domain. The example query can be represented as.
- which a river is longest
- Ans (x, longest (x, river (x)))

(Q10)) System Padagains. (QD)

Approaches for practical simple implementation of Islamic interpretation.

- * Based on diversity of languages, & levels of granularity ...
- * The main constraint is posed by lack of data availability of lang.
- * Due to these constraint, some successful approaches for practical implementation fall in these 3 categories

① System Architecture

a) Knowledge based :-

The systems are designed using predefined set of rules for finding the solution to the particular problem

b) Unsupervised :-

Sol to a particular application is found by minimum him a unsupervised implement

c) Semi Supervised :-

* The technique involves model training by application of diff ML algo.

* Feature functions are created to create function which can used to predict labels to handle censored data

Q) Scope

- a) Domain dependent :- System designed for specific domain such as travel reservations.
- b) Domain independent :- It includes generic techniques.

③ Coverage

- a) Shallow :- In this approach, intermediate representation is generated to be used by machine.
- b) Deep :- Through this approach representation, which is created, directly used by machine.

Q) Word sense 22).