Solution to NLP Viva Questions

1. **What do you understand by NLP.**

Ans:

- Language is a method of communication with the help of which we can speak, read and write. Natural Language Processing (NLP) is a subfield of Computer Science that deals with Artificial Intelligence (AI), which enables computers to understand and process human language.

- Natural Language Processing is the technology used to aid computers to understand the human's natural language.

(ek machine ka insano ki language samjhna fir uspe answer back karna ya jo action kaha hai usko perform karna is natural Language processing)

2. **Applications of NLP (anyone in detail)**

Ans:

- Machine Translation
- Sentimental Analysis
- Automatic Summarization
- Question answering
- Speech recognition

Machine Translation

- Machine translation (MT), the process of translating one source language or text into another language, is one of the most important applications of NLP.
- There are different types of machine translation systems. Let us see what the different types are.
- Bilingual MT systems produce translations between two particular languages.
- Multilingual MT systems produce translations between any pair of languages. They may be either unidirectional or bi-directional in nature.
- Example : Google translator

Sentiment Analysis

- Another important application of natural language processing (NLP) is sentiment analysis.

- As the name suggests, sentiment analysis is used to identify the sentiments among several posts.
- It is also used to identify the sentiment where the emotions are not expressed explicitly.
- Companies are using sentiment analysis, an application of natural language processing (NLP) to identify the opinion and sentiment of their customers online.
- It will help companies to understand what their customers think about the products and services.
- Companies can judge their overall reputation from customer posts with the help of sentiment analysis.
- In this way, we can say that beyond determining simple polarity, sentiment analysis understands sentiments in context to help us better understand what is behind the expressed opinion.

Automatic Summarization

- In this digital era, the most valuable thing is data, or you can say information.
- However, do we really get useful as well as the required amount of information? The answer is 'NO' because the information is overloaded and our access to knowledge and information far exceeds our capacity to understand it.
- We are in a serious need of automatic text summarization and information because the flood of information over the internet is not going to stop.
- Text summarization may be defined as the technique to create short, accurate summary of longer text documents.
- Automatic text summarization will help us with relevant information in less time. Natural language processing (NLP) plays an important role in developing an automatic text summarization.

Question-answering

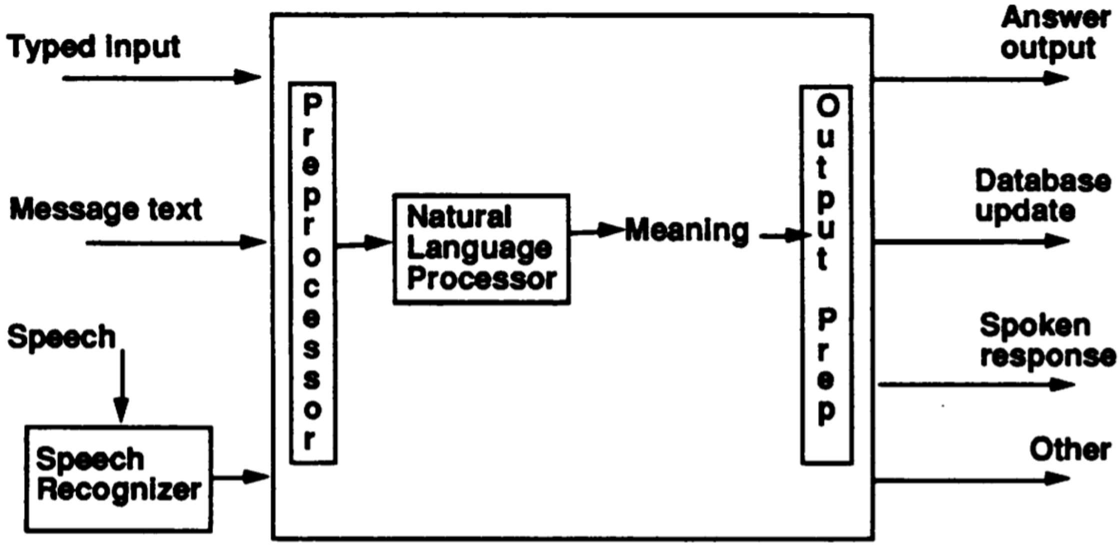
- Another main application of natural language processing (NLP) is question-answering. Search engines put the information of the world at our fingertips, but they are still lacking when it comes to answering the questions posted by human beings in their natural language.
- We have big tech companies like Google are also working in this direction.
- Question-answering is a Computer Science discipline within the fields of AI and NLP.
- It focuses on building systems that automatically answer questions posted by human beings in their natural language.
- A computer system that understands the natural language has the capability of a program system to translate the sentences written by humans into an internal representation so that the valid answers can be generated by the system.

- The exact answers can be generated by doing syntax and semantic analysis of the questions. Lexical gap, ambiguity and multilingualism are some of the challenges for NLP in building a good question answering system.

Speech recognition

- Speech recognition (enables computers to recognize and transform spoken language into text – dictation – and, if programmed, act upon that recognition – e.g. in case of assistants like Google Assistant Cortana or Apple's Siri)
- Speech recognition is simply the ability of a software to recognise speech.
- Anything that a person says, in a language of their choice, must be recognised by the software.
- Speech recognition technology can be used to perform an action based on the instructions defined by the human.
- Humans need to train the speech recognition system by storing speech patterns and vocabulary of their language into the system.
- By doing so, they can essentially train the system to understand them when they speak.
- Speech recognition and Natural Language processing are usually used together in Automatic Speech Recognition engines, Voice Assistants and Speech analytics tools.

3. **Generic NLP system.**



4. **Need for NLP.**

Ans:

- The essence of NLP lies in making computers understand Natural Language
- Computers can understand the structured form of data (Eg Spreadsheets and tables in databases)
- Human Languages,texts and voices form an unstructured category of data.
- Hence,Difficult for computer to understand it.
- There arises the need of NLP

5. **Goals of NLP.**

Ans:

- The ultimate goal of natural language processing is for computers to achieve human-like comprehension of texts/languages**.** When this is achieved, computer systems will be able to understand, draw inferences from, summarize, translate and generate accurate and natural human text and language**.**
- The goal of natural language processing is to specify a language comprehension and production theory to such a level of detail that a person is able to write a computer program which can understand and produce natural language
- The basic goal of NLP is to accomplish human like language processing. The choice of word "processing" is very deliberate and should not be replaced with "understanding". For although the field of NLP was originally referred to as Natural Language Understanding (NLU), that goal has not yet been accomplished. A full NLU system would be able to:

- ➢ Paraphrase an input text.
- ➢ Translate the text into another language.
- ➢ Answer questions about the contents of the text.
- ➢ Draw inferences from the text.

6. **Levels of NLP.**

Ans: Natural Language Processing works on multiple levels and most often, these different areas synergize well with each other. This article will offer a brief overview of each and provide some example of how they are used in information retrieval.

Morphological

- The morphological level of linguistic processing deals with the study of word structures and word formation, focusing on the analysis of the individual components of words.
- The most important unit of morphology, defined as having the "minimal unit of meaning", is referred to as the *morpheme*.
- Taking, for example, the word: *"unhappiness"*. It can be broken down into three morphemes (prefix, stem, and suffix), with each conveying some form of meaning: the prefix *un-* refers to "not being", while the suffix *-ness* refers to "a state of being".

- The stem *happy* is considered as a *free morpheme* since it is a “word” in its own right. *Bound morphemes* (prefixes and suffixes) require a free morpheme to which it can be attached to, and can therefore not appear as a “word” on their own.
- In Information Retrieval, document and query terms can be stemmed to match the morphological variants of terms between the documents and query; such that the singular form of a noun in a query will match even with its plural form in the document, and vice versa, thereby increasing recall.

Lexical

- The lexical analysis in NLP deals with the study at the level of words with respect to their lexical meaning and part-of-speech. This level of linguistic processing utilizes a language’s *lexicon,* which is a collection of individual *lexemes*.
- A lexeme is a basic unit of lexical meaning; which is an abstract unit of morphological analysis that represents the set of forms or “senses” taken by a single morpheme.
- *“Duck”,* for example, can take the form of a noun or a verb but its part-of-speech and lexical meaning can only be derived in context with other words used in the phrase/sentence.
- This, in fact, is an early step towards a more sophisticated Information Retrieval system where precision is improved through part-of-speech tagging.

Syntactic

- The part-of-speech tagging output of the lexical analysis can be used at the syntactic level of linguistic processing to group words into the phrase and clause brackets.
- Syntactic Analysis also referred to as *“parsing”*, allows the extraction of phrases which convey more meaning than just the individual words by themselves, such as in a noun phrase.
- In Information Retrieval, parsing can be leveraged to improve indexing since phrases can be used as representations of documents which provide better information than just single-word indices.
- In the same way, phrases that are syntactically derived from the query offers better search keys to match with documents that are similarly parsed.
- Nevertheless, syntax can still be ambiguous at times as in the case of the news headline: *“Boy paralyzed after tumour fights back to gain black belt”* — which actually refers to how a boy was paralyzed because of a tumour but endured the fight against the disease and ultimately gained a high level of competence in martial arts.

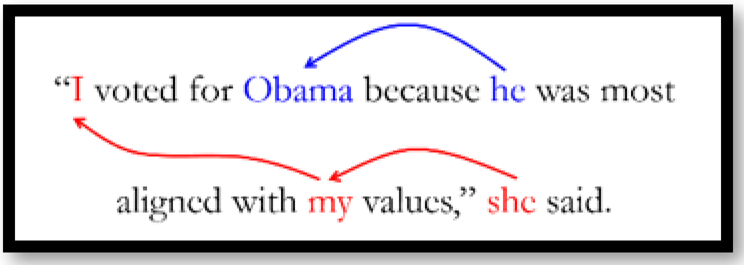
Semantic

- The semantic level of linguistic processing deals with the determination of what a sentence really means by relating syntactic features and disambiguating words with multiple definitions to the given context.
- This level entails the appropriate interpretation of the meaning of sentences, rather than the analysis at the level of individual words or phrases.

- In Information Retrieval, the query and document matching process can be performed on a conceptual level, as opposed to simple terms, thereby further increasing system precision.
- Moreover, by applying semantic analysis to the query, term expansion would be possible with the use of lexical sources, offering improved retrieval of the relevant documents even if exact terms are not used in the query.
- Precision may increase with query expansion, as with recall probably increasing as well.

Discourse

- The discourse level of linguistic processing deals with the analysis of structure and meaning of text beyond a single sentence, making connections between words and sentences.
- At this level, Anaphora Resolution is also achieved by identifying the entity referenced by an anaphor (most commonly in the form of, but not limited to, a pronoun).
- An example is shown below.



“I voted for Obama because he was most aligned with my values,” she said.

Fig 1. Anaphora Resolution Illustration

- With the capability to recognize and resolve anaphora relationships, document and query representations are improved, since, at the lexical level, the implicit presence of concepts is accounted for throughout the document as well as in the query, while at the semantic and discourse levels, an integrated content representation of the documents and queries are generated.
- Structured documents also benefit from the analysis at the discourse level since sections can be broken down into (1) title, (2) abstract, (3) introduction, (4) body, (5) results, (6) analysis, (7) conclusion, and (8) references. Information Retrieval systems are significantly improved, as the specific roles of pieces of information are determined as for whether it is a conclusion, an opinion, a prediction, or a fact.

Pragmatic

- The pragmatic level of linguistic processing deals with the use of real-world knowledge and understanding of how this impacts the meaning of what is being communicated.
- By analyzing the contextual dimension of the documents and queries, a more detailed representation is derived.
- In Information Retrieval, this level of Natural Language Processing primarily engages query processing and understanding by integrating the user’s history and goals as well as the context upon which the query is being made.
- Contexts may include time and location.
- This level of analysis enables major breakthroughs in Information Retrieval as it facilitates the conversation between the IR system and the users, allowing the elicitation of the

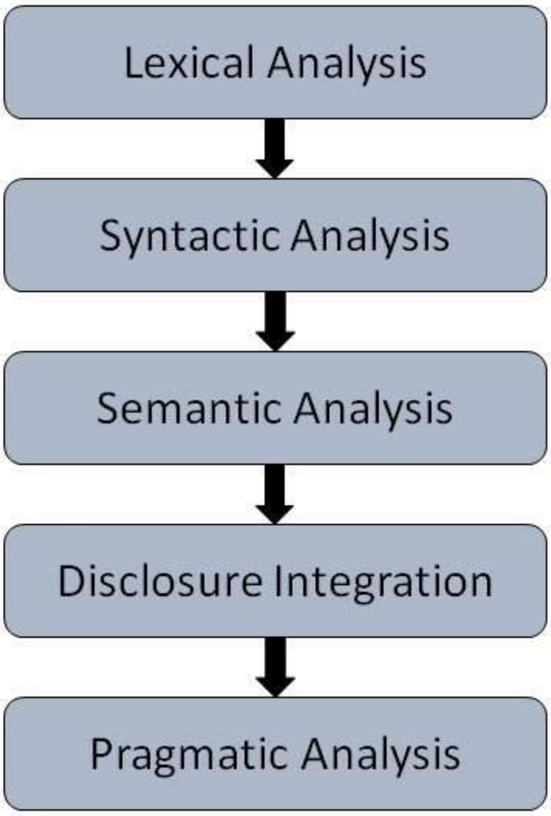
purpose upon which the information being sought is planned to be used, thereby ensuring that the information retrieval system is fit for purpose.

7. **Stages in NLP.**

Ans:

There are general five stages−

- **Lexical Analysis** − It involves identifying and analyzing the structure of words. Lexicon of a language means the collection of words and phrases in a language. Lexical analysis is dividing the whole chunk of txt into paragraphs, sentences, and words.
- **Syntactic Analysis (Parsing)** − It involves analysis of words in the sentence for grammar and arranging words in a manner that shows the relationship among the words. The sentence such as "The school goes to boy" is rejected by English syntactic analyzer.



- **Semantic Analysis** − It draws the exact meaning or the dictionary meaning from the text. The text is checked for meaningfulness. It is done by mapping syntactic structures and objects in the task domain. The semantic analyzer disregards sentence such as "hot ice-cream".
- **Discourse Integration** − The meaning of any sentence depends upon the meaning of the sentence just before it. In addition, it also brings about the meaning of immediately succeeding sentence.
- **Pragmatic Analysis** − During this, what was said is re-interpreted on what it actually meant. It involves deriving those aspects of language which require real world knowledge.

8. **Ambiguity in NLP.**

Ans:

- Ambiguity, generally used in natural language processing, can be referred to as the ability of being understood in more than one way.
- In simple terms, we can say that ambiguity is the capability of being understood in more than one way.
- Natural language is very ambiguous.

NLP has the following types of ambiguities –

Lexical Ambiguity

The ambiguity of a single word is called lexical ambiguity. For example, treating the word silver as a noun, an adjective, or a verb.

- She won two silver medals
- She made a silver speech
- His worries has silvered his hair

Syntactic Ambiguity

- This kind of ambiguity occurs when a sentence is parsed in different ways.
- For example, the sentence “The man saw the girl with the telescope”. It is ambiguous whether the man saw the girl carrying a telescope or he saw her through his telescope.

Semantic Ambiguity

- This kind of ambiguity occurs when the meaning of the words themselves can be misinterpreted even after syntax and the meaning of individual word have been resolved.
- In other words, semantic ambiguity happens when a sentence contains an ambiguous word or phrase.

Example 1

- “Seema loves her mother and shreya does too.”
- Here there are two meaning “seema lover her mother and shreya loves her own mother” and “seema lover her mother and shreya also loves seema mother”

Example 2

“The car hit the pole while it was moving” is having semantic ambiguity because the interpretations can be “The car, while moving, hit the pole” and “The car hit the pole while the pole was moving”.

Anaphoric Ambiguity

- This kind of ambiguity arises due to the use of anaphora entities in discourse.
- Anaphora : when same beginning of sentence is repeated several times
- Example of what anaphora is “my mother liked the house very much but she couldn’t buy it” here we are not repeating mother again and again we replaced it by she so here she is anaphora
- Now lets come back to anaphoric ambiguity and understand it with the help of example
- For example, the horse ran up the hill. It was very steep. It soon got tired. Here, the anaphoric reference of “it” in two situations can either be horse or hill which cause anaphoric ambiguity.

Pragmatic ambiguity

- It occurs when sentence gives it multiple interpretations or it is not specific.
- For example, the sentence “I like you too” can have multiple interpretations like I like you (just like you like me), I like you (just like someone else does).

9. **What is morphological analysis.**

Ans: Morphological analysis is the process of providing grammatical information about the word on the basis of properties of the morpheme it contains. It is an integral part of the larger natural language processing projects such as text to speech synthesis, information extraction and machine translation.

10. **What is stemming , lemmatization , difference between the two.**

Ans:

- **Stemming:** It is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words known as a lemma. Stemming is important in natural language processing (NLP).Stemming is a part of linguistic studies in morphology and artificial intelligence (AI) information retrieval and extraction. Stemming and AI knowledge extract meaningful information from vast sources like big data or the Internet since additional forms of a word related to a subject may need to be searched to get the best results. Stemming is also a part of queries and Internet search engines.

- **Lemmatization:** It is the process of grouping together the inflected forms of a word so they can be analysed as a single item, identified by the word's lemma, or dictionary form. Lemmatization is the algorithmic process of determining the lemma of a word based on its intended meaning. Unlike stemming, lemmatisation depends on correctly identifying the intended part of speech and meaning of a word in a sentence, as well as within the larger context surrounding that sentence, such as neighboring sentences or even an entire document.

- **Difference Between Stemming & Lemmatization**

Stemming	Lemmatization
Stemming usually operates on single word without knowledge of the context	Lemmatization usually considers words and the context of the word in the sentence
Stemming usually operates on single word without knowledge of the context	In lemmatization, we consider POS tags
Stemming is used to group words with a similar basic meaning together	Lemmatization concept is used to make dictionary or WordNet kind of dictionary.

11. **What is inflectional, derivational morphology.**

Ans:

Inflectional morphology

- Inflectional morphology is the study of processes, including affixation and vowel change, that distinguish word forms in certain grammatical categories.
- We may define inflectional morphology as the branch of morphology that deals with paradigms. It is therefore concerned with two thing: on the one hand, with the semantic oppositions among categories; and on the other, with the formal means, including inflections, that distinguish them." (Matthews, 1991, p.38).
- In addition to Matthews definition I would say what one should remember to understand inflectional morphology is that it changes the word form, it determines the grammar and it does not form a new lexeme but rather a variant of a lexeme that does not need its own entry in the dictionary.

Derivational Morphology:

- Derivational morphology is defined as morphology that creates new lexemes, either by changing the syntactic category (part of speech) of a base or by adding substantial, non-grammatical meaning or both.
- On the one hand, derivation may be distinguished from inflectional morphology, which typically does not change category but rather modifies lexemes to fit into various syntactic contexts; inflection typically expresses distinctions like number, case, tense, aspect, person, among others.
- On the other hand, derivation may be distinguished from compounding, which also creates new lexemes, but by combining two or more bases rather than by affixation, reduplication, subtraction, or internal modification of various sorts.
- Although the distinctions are generally useful, in practice applying them is not always easy.

12. **Types of stemmers.**

- **Porter's Stemmer**
  It is one of the most popular stemming methods proposed in 1980. It is based on the idea that the suffixes in the English language are made up of a combination of smaller and simpler suffixes.
  Example: EED -> EE means "if the word has at least one vowel and consonant plus EED ending, change the ending to EE" as 'agreed' becomes 'agree'.
  Advantage: It produces the best output as compared to other stemmers and it has less error rate.
  Limitation: Morphological variants produced are not always real words.
- **Lovins Stemmer**
  It is proposed by Lovins in 1968, that removes the longest suffix from a word then word is recoded to convert this stem into valid words.
  Example: sitting -> sitt -> sit
  Advantage: It is fast and handles irregular plurals like 'teeth' and 'tooth' etc.
  Limitation: It is time consuming and frequently fails to form words from stem.
- **Dawson Stemmer**
  It is extension of Lovins stemmer in which suffixes are stored in the reversed order indexed by their length and last letter.
  Advantage: It is fast in execution and covers more suffices.
  Limitation: It is very complex to implement.
- **Krovetz Stemmer**
  It was proposed in 1993 by Robert Krovetz. Following are the steps:
  1) Convert the plural form of a word to its singular form.
  2) Convert the past tense of a word to its present tense and remove the suffix 'ing'.
  Example: 'children' -> 'child'
  Advantage: It is light in nature and can be used as pre-stemmer for other stemmers.
  Limitation: It is inefficient in case of large documents.
- **Xerox Stemmer**
  Example:
  'children' -> 'child'
  'understood' -> 'understand'
  'whom' -> 'who'
  'best' -> 'good'
  Advantage: It works well in case of large documents and stems produced are valid.
  Limitation: It is language dependent and mainly implemented on english and over stemming may occur.
- **N-Gram Stemmer**
  An n-gram is a set of n consecutive characters extracted from a word in which similar words will have a high proportion of n-grams in common.
  Example: 'INTRODUCTIONS' for n=2 becomes : *I, IN, NT, TR, RO, OD, DU, UC, CT, TI, IO, ON, NS, S*
  Advantage: It is based on string comparisons and it is language dependent.
  Limitation: It requires space to create and index the n-grams and it is not time efficient

13. **Porter stemmer.**

The Porter stemming algorithm (or 'Porter stemmer') is a process for removing the commoner morphological and inflexional endings from words in English. Its main use is as part of a term normalisation process that is usually done when setting up Information Retrieval systems.

- A consonant in a word is a letter other than A, E, I, O or U, and other than Y preceded by a consonant. (The fact that the term **consonant** is defined to some extent in terms of itself does not make it ambiguous.) So in TOY the consonants are T and Y, and in SYZYGY they are S, Z and G. If a letter is not a consonant it is a vowel.
- A consonant will be denoted by c, a vowel by v. A list ccc... of length greater than 0 will be denoted by C, and a list vvv... of length greater than 0 will be denoted by V. Any word, or part of a word, therefore has one of the four forms:
- CVCV ... C
- CVCV ... V
- VCVC ... C
- VCVC ... V

These may all be represented by the single form

[C]VCVC ... [V]

where the square brackets denote arbitrary presence of their contents. Using (VC^{m}) to denote VC repeated m times, this may again be written as

[C](VC^{m})[V]

m will be called the measure of any word or word part when represented in this form. The case m = 0 covers the null word. Here are some examples:

- m=0 TR, EE, TREE, Y, BY.
- m=1 TROUBLE, OATS, TREES, IVY.
- m=2 TROUBLES, PRIVATE, OATEN, ORRERY.

The rules for removing a suffix will be given in the form

(condition) S1 -> S2

This means that if a word ends with the suffix S1, and the stem before S1 satisfies the given condition, S1 is replaced by S2. The condition is usually given in terms of m, e.g.

(m > 1) EMENT ->

Here S1 is 'EMENT' and S2 is null. This would map REPLACEMENT to REPLAC, since REPLAC is a word part for which m = 2.

The 'condition' part may also contain the following:

- *S - the stem ends with S (and similarly for the other letters).
- *v* - the stem contains a vowel.
- m=2 TROUBLES, PRIVATE, OATEN, ORRERY.
- *d - the stem ends with a double consonant (e.g. -TT, -SS).
- *o - the stem ends cvc, where the second c is not W, X or Y (e.g. -WIL, -HOP).

And the condition part may also contain expressions with and, or and not, so that:

(m>1 and (*S or *T)) : tests for a stem with m>1 ending in S or T, while
(*d and not (*L or *S or *Z)) : tests for a stem ending witha double consonant other than L, S or Z.
Elaborate conditions like this are required only rarely.
In a set of rules written beneath each other, only one is obeyed, and this will be the one with the longest matching S1 for the given word. For example, with

- SSES -> SS
- IES -> I
- SS -> SS
- S ->

(here the conditions are all null) CARESSES maps to CARESS since SSES is the longest match for S1.
Equally CARESS maps to CARESS (S1=SS) and CARES to CARE (S1=S).
In the rules below, examples of their application, successful or otherwise, are given on the right in lower case. The algorithm now follows:

Step 1a :

- SSES -> SS (Example : caresses -> caress)
- IES -> I (Example : ponies -> poni ; ties -> ti)
- SS -> SS (Example : caress -> caress)
- S -> (Example : cats -> cat)

Step 1b :

- (m>0) EED -> EE (Example : feed -> feed ; agreed -> agree)
- (*v*) ED -> (Example : plastered -> plaster ; bled -> bled)
- (*v*) ING -> (Example : motoring -> motor ; sing -> sing)
- S -> (Example : cats -> cat)

If the second or third of the rules in Step 1b is successful, the following is done:

- AT -> ATE (Example : conflat(ed) -> conflate)

- BL -> BLE (Example : troubl(ed) -> trouble)
- IZ -> IZE (Example : siz(ed) -> size)
- S -> (Example : cats -> cat)
- (*d and not (*L or *S or *Z)) -> single letter (Example : hopp(ing) -> hop ; tann(ed) -> tan ; fall(ing) -> fall ; hiss(ing) -> hiss ; fizz(ed) -> fizz)
- (m=1 and *o) -> E (Example : fail(ing) -> fail ; fil(ing) -> file)
  The rule to map to a single letter causes the removal of one of the double letter pair. The -E is put back on -AT, -BL and -IZ, so that the suffixes -ATE, -BLE and -IZE can be recognised later. This E may be removed in step 4.

  Step 1c :
  (*v*) Y -> I (Example : happy -> happi ; sky -> sky)
  Step 1 deals with plurals and past participles. The subsequent steps are much more straightforward.
  Step 2 :
- (m>0) ATIONAL -> ATE (Example : relational -> relate
- (m>0) TIONAL -> TION (Example : conditional -> condition ; rational -> rational)
- (m>0) ENCI -> ENCE (Example : valenci -> valence)
- (m>0) ANCI -> ANCE (Example : hesitanci -> hesitance)
- (m>0) IZER -> IZE (Example : digitizer -> digitize)
- (m>0) ABLI -> ABLE (Example : conformabli -> conformable)
- (m>0) ALLI -> AL (Example : radicalli -> radical)
- (m>0) ENTLI -> ENT (differentli -> different)
- (m>0) ELI -> E (vileli -> vile)
- (m>0) OUSLI -> OUS (analogousli -> analogous)
- (m>0) IZATION -> IZE (vietnamization -> vietnamize)
- (m>0) ATION -> ATE (predication -> predicate)
- (m>0) ATOR -> ATE (operator -> operate)
- (m>0) ALISM -> AL (feudalism -> feudal)
- (m>0) IVENESS -> IVE (decisiveness -> decisive)
- (m>0) FULNESS -> FUL (hopefulness -> hopeful)
- (m>0) OUSNESS -> OUS (callousness -> callous)

- (m>0) ALITI -> AL (formaliti -> formal)
- (m>0) IVITI -> IVE (sensitiviti -> sensitive)
- (m>0) BILITI -> BLE (sensibiliti -> sensible)

The test for the string S1 can be made fast by doing a program switch on the penultimate letter of the word being tested. This gives a fairly even breakdown of the possible values of the string S1. It will be seen in fact that the S1-strings in step 2 are presented here in the alphabetical order of their penultimate letter. Similar techniques may be applied in the other steps.

Step 3 :

- (m>0) ICATE -> IC (Example : triplicate -> triplic)
- (m>0) ATIVE -> (Example : formative -> form)
- (m>0) ALIZE -> AL (Example : formalize -> formal)
- (m>0) ICITI -> IC (Example : electriciti -> electric))
- (m>0) ICAL -> IC (Example : electrical -> electric)
- (m>0) FUL -> (Example : hopeful -> hope)
- (m>0) NESS -> (Example : goodness -> good)

Step 4 :

- (m>1) AL -> (Example : revival -> reviv)
- (m>1) ANCE -> (Example : allowance -> allow)
- (m>1) ENCE -> (Example : inference -> infer)
- (m>1) ER -> (Example : airliner -> airlin)
- (m>1) IC -> (Example : gyroscopic -> gyroscop)
- (m>1) ABLE -> (Example : adjustable -> adjust)
- (m>1) IBLE -> (Example : defensible -> defens)
- (m>1) ANT -> (Example : irritant -> irrit)
- (m>1) EMENT -> (Example : replacement -> replac)
- (m>1) MENT -> (Example : adjustment -> adjust)
- (m>1) ENT -> (Example : dependent -> depend)
- (m>1 and (*S or *T)) ION -> (Example : adoption -> adopt)
- (m>1) OU -> (Example : homologou -> homolog)

- (m>1) ISM -> (Example : communism -> commun)
- (m>1) ATE -> (Example : activate -> activ)
- (m>1) ITI -> (Example : angulariti -> angular)
- (m>1) OUS -> (Example : homologous -> homolog)
- (m>1) IVE -> (Example : effective -> effect)
- (m>1) IZE -> (Example : bowdlerize -> bowdler)

The suffixes are now removed. All that remains is a little tidying up.

Step 5a :

- (m>1) E -> (Example : probate -> probat ; rate -> rate)
- (m=1 and not *o) E -> (Example : cease -> ceas)

Step 5b
(m > 1 and *d and *L) -> single letter
(Example : controll -> control ; roll -> roll)

14. **N gram model.**

- You can think of an N-gram as the sequence of N words, by that notion, a 2-gram (or bigram) is a two-word sequence of words like “please turn”, “turn your”, or ”your homework”, and a 3-gram (or trigram) is a three-word sequence of words like “please turn your”, or “turn your homework”
- Let’s start with equation P(w|h), the probability of word w, given some history, h. For example,

$$P(the \mid its\ water\ is\ so\ transperant\ that)$$

- Here,
  w = The
  h = its water is so transparent that
- And, one way to estimate the above probability function is through the relative frequency count approach, where you would take a substantially large corpus, count the number of times you see its water is so transparent that, and then count the number of times it is followed by the. In other words, you are answering the question:
- Out of the times you saw the history h, how many times did the word w follow it

$$P(the \mid its\ water\ is\ so\ transperant\ that) = C(\ its\ water\ is\ so\ transperant\ that\ the) / C(\ its\ water\ is\ so\ transperant\ that)$$

- Now, you can imagine it is not feasible to perform this over an entire corpus; especially it is of a significant a size.

- This shortcoming and ways to decompose the probability function using the chain rule serves as the base intuition of the N-gram model.
- Here, you, instead of computing probability using the entire corpus, would approximate it by just a few historical words

The Bigram Model

- As the name suggests, the bigram model approximates the probability of a word given all the previous words by using only the conditional probability of one preceding word. In other words, you approximate it with the probability:

  P(the | that)
- And so, when you use a bigram model to predict the conditional probability of the next word, you are thus making the following approximation:

$$P(w_n|w_1^{n-1}) \approx P(w_n|w_{n-1})$$

- This assumption that the probability of a word depends only on the previous word is also known as Markov assumption.
- Markov models are the class of probabilisitic models that assume that we can predict the probability of some future unit without looking too far in the past.
- You can further generalize the bigram model to the trigram model which looks two words into the past and can thus be further generalized to the N-gram model

15. **What is Part of Speech Tagging.**

Ans:

- Tagging is a kind of classification that may be defined as the automatic assignment of description to the tokens. Here the descriptor is called tag, which may represent one of the part-of-speech, semantic information and so on.

- Now, if we talk about Part-of-Speech (PoS) tagging, then it may be defined as the process of assigning one of the parts of speech to the given word. It is generally called POS tagging.

- In simple words, we can say that POS tagging is a task of labelling each word in a sentence with its appropriate part of speech. We already know that parts of speech include nouns, verb, adverbs, adjectives, pronouns, conjunction and their sub-categories.

16. **Rule based, Stochastic and Transformation based tagging**.

Ans:

Rule-based POS 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 −

- First stage − In the first stage, it uses a dictionary to assign each word a list of potential parts-of-speech.

- Second stage − 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.

Stochastic POS Tagging

- Another technique of tagging is Stochastic POS Tagging. Now, the question that arises here is which model can be stochastic.
- The model that includes frequency or probability (statistics) can be called stochastic.
- Any number of different approaches to the problem of part-of-speech tagging can be referred to as stochastic tagger.

The simplest stochastic tagger applies the following approaches for POS tagging −

Word Frequency Approach

In this approach, the stochastic taggers disambiguate the words based on the probability that a word occurs with a particular tag. We can also say that 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.

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.

Transformation-based Tagging

- Transformation based tagging is also called Brill tagging. 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.
- TBL, allows us to have linguistic knowledge in a readable form, transforms one state to another state by using transformation rules.
- It draws the inspiration from both the previous explained taggers − rule-based and stochastic.
- 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.
- 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.

17. **Homonymy, Polysemy, Synonymy, Hyponymy**

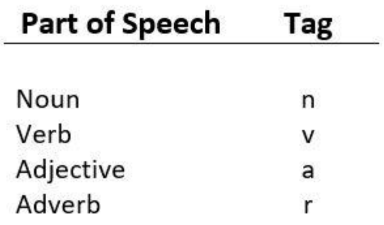
Ans:

- **Homonymy**: It may be defined as the words having same spelling or same form but having different and unrelated meaning. For example, the word “Bat” is a **homonymy** word because bat can be an implement to hit a ball or bat is a nocturnal flying mammal also.
- **Polysemy**:It is a Greek word, which means “many signs”. It is a word or phrase with different but related sense. In other words, we can say that polysemy has the same spelling but different and related meaning. For example, the word “bank” is a polysemy word having the following meanings –
  - A financial institution.
  - The building in which such an institution is located.
  - A synonym for “to rely on”.
- **Hyponymy:**It may be defined as the relationship between a generic term and instances of that generic term. Here the generic term is called hypernym and its instances are called hyponyms. For example, the word color is hypernym and the color blue, yellow etc. are hyponyms.

18. **What is Wordnet.**

Ans:

- **WordNet** is the lexical database i.e. dictionary for the English language, specifically designed for natural language processing.
- **Synset** is a special kind of a simple interface that is present in NLTK to look up words in WordNet.
- Synset instances are the groupings of synonymous words that express the same concept. Some of the words have only one Synset and some have several.



Part of Speech	Tag
Noun	n
Verb	v
Adjective	a
Adverb	r

19. **What is Word Sense Disambiguation and approaches to do the same.**

Ans:

- 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.
- Lexical ambiguity, syntactic or semantic, is one of the very first problem 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 "bass" −

- I can hear bass sound.
- He likes to eat grilled bass.

- The occurrence of the word bass clearly denotes the distinct meaning. In first sentence, it means frequency and in second, it means fish.
- Hence, if it would be disambiguated by WSD then the correct meaning to the above sentences can be assigned as follows −

- I can hear bass/frequency sound.
- He likes to eat grilled bass/fish.

Approaches and methods to WSD are classified according to the source of knowledge used in word disambiguation.

Let us now see the four conventional methods to WSD −

Dictionary-based or Knowledge-based Methods

- As the name suggests, for disambiguation, these methods primarily rely on dictionaries, treasures and lexical knowledge base.
- They do not use corpora evidences 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 surrounding sentence or paragraph.

Supervised Methods

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

Semi-supervised Methods

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

Unsupervised Methods

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

20. **Study all applications in detail(Machine translation, Information retrieval, Summarization, Sentiment Analysis, Text Categorization or Classification, Named Entity Recognition).**

Ans: (Refer Applications of NLP question)