# Natural Language Processing

**## How to clean your text data?**

* Remove all irrelevant characters such as any non alphanumeric characters
* Tokenize your text by separating it into individual words
* Remove words that are not relevant, such as “@” twitter mentions or urls
* Convert all characters to lowercase((\*\*Case folding\*\*), in order to treat words such as “hello”, “Hello”, and “HELLO” the same. (Note)
* Consider combining misspelled or alternately spelled words to a single representation (e.g. “cool”/”kewl”/”cooool”)
* Consider lemmatization (reduce words such as “am”, “are”, and “is” to a common form such as “be”)
* Consider removing stopwords (such as a, an, the, be)etc.

Note : For tasks like speech recognition and information retrieval, everything is mapped to lower case. For sentiment analysis and other text classification tasks, information extraction, and machine translation, by contrast, case is quite helpful and case folding is generally not done (losing the difference, for example, between US the country and us the pronoun can out- weigh the advantage in generality that case folding provides)

**## What is Tokenization?**

Tokenization is the process of converting a sequence of characters into a sequence of tokens.Ex :RegexpTokenizer & Word Tokenize (scikit-learn)

**## What is stop words?**

Stop words are words that are particularly common in a text corpus and thus considered as rather un-informative.

**## What is Stemming and lemmatization?**

The goal of both stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form.

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. Different types of stemmers in NLTK are PorterStemmer, LancasterStemmer, SnowballStemmer.

Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma.

Note : It uses a knowledgebase called WordNet. Because of knowledge, lemmatization can even convert words which are different and cant be solved by stemmers, for example converting “came” to “come”.

**## Parts-of-Speech (POS) tagging :**

Part-of-speech tagging (POS tagging) is the task of tagging a word in a text with its part of speech. A part of speech is a category of words with similar grammatical properties. Common English parts of speech are noun, verb, adjective, adverb, pronoun, preposition, conjunction, etc.

**## Named Entity Recognition (NER):**

In the Named Entity Recognition (NER) task, systems are required to recognize the Named Entities occurring in the text. More specifically, the task is to find Person (PER), Organization (ORG), Location (LOC) and Geo-Political Entities (GPE). For instance, in the statement ”Shyam lives in India”, NER system extracts Shyam which refers to name of the person and India which refers to name of the country.

**## Corefrence resolution(CR):**

Coreference Resolution is the task which determines which noun phrases (including pronouns, proper names and common names) refer to the same entities in documents . For instance, in the sentence, ”I have seen the annual report. It shows that we have gained 15% profit in this financial year”. Here, ”I” refers to name of the person, ”It” refers to annual report and ”we” refers to the name of the company in which that person works.(Kong et al., 2010)

**## What is Bag of Words?**

\*\*Bag of words (BoW)\*\* builds a vocabulary of all the unique words in our dataset, and associate a unique index to each word in the vocabulary.It is called a "bag" of words, because it is a representation that completely ignores the order of words.

**## What is tf - idf?**

\*\*TF-IDF\*\* reveals what words are the most discriminating between different bodies of text. It is dependent on term frequency, how often a word appears, and Inverse document frequency, whether it is unique or common among all documents. It is particularly, helpful if you are trying to see the difference between words that occur a lot in one document, but fail to appear in others allowing you interpret something special about that document. Example:

Consider a document containing 100 words wherein the word cat appears 3 times. The term frequency (i.e., tf) for cat is then (3 / 100) = 0.03. Now, assume we have 10 million documents and the word cat appears in one thousand of these. Then, the inverse document frequency (i.e., idf) is calculated as log(10,000,000 / 1,000) = 4. Thus, the Tf-idf weight is the product of these quantities: 0.03 \* 4 = 0.12.

**## N gram**

* n-gram is a contiguous sequence of n items from a given sample of text or speech
* An n-gram of size 1 is referred to as a "unigram"; size 2 is a "bigram" size 3 is a "trigram". Larger sizes are sometimes referred to by the value of n in modern language, e.g., "four-gram", "five-gram", and so on.
* ngram model models sequence, i.e., predicts next word (n) given previous words (1, 2, 3, ..., n-1)
* multiple gram (bigram and above) captures \*\*context\*\*
* to choose n in n-gram requires experiments and making tradeoff between stability of the estimate against its appropriateness. Rule of thumb: trigram is a common choice with large training corpora (millions of words), whereas a bigram is often used with smaller ones.
* n-gram can be used as features for machine learning and downstream NLP tasks.

**## What is word2vec ?**

It is a shallow two-layer neural networks that are trained to construct linguistic context of words.It Takes as input a large corpus, and produce a vector space, typically of several hundred dimension, and each word in the corpus is assigned a vector in the space.The key idea is context: words that occur often in the same context should have same/opposite meanings.

Two Algorithms:

**Skip-Grams :** The skip-gram model does the exact opposite of the CBOW model, by predicting the surrounding context words given the central target word.

**Continuous Bag of Words (CBOW):** CBOW computes the conditional probability of a target word given the context words surrounding it across a window of size k.

**##Limitations:**

* When we want to obtain vector representations for phrases such as “hot potato” or “Boston Globe”. We can’t just simply combine the individual word vector representations since these phrases don’t represent the combination of meaning of the individual words. And it gets even more complicated when longer phrases and sentences are considered.
* use of smaller window sizes produce similar embeddings for contrasting words such as “good” and “bad”, which is not desirable especially for tasks where this differentiation is important such as sentiment analysis.At times these embeddings cluster semantically-similar words which have opposing sentiment polarities.

**## NLP Metrics :**

**Perplexity :** The perplexity (sometimes called PP for short) of a language model on a test set is the inverse probability of the test set, normalized by the number of words. (\*Smaller is better\*)

Note : Perplexity does not guarantee an (extrinsic) improvement in the performance of a language processing task like speech recognition or machine translation.

**1. What do you understand by Natural Language Processing?**

Natural Language Processing is a field of computer science that deals with communication between computer systems and humans. It is a technique used in Artificial Intelligence and Machine Learning. It is used to create automated software that helps understand human spoken languages to extract useful information from the data it gets in the form of audio. Techniques in NLP allow computer systems to process and interpret data in the form of natural languages.

**2. What are stop words?**

Stop words are said to be useless data for a search engine. Words such as articles, prepositions, etc. are considered as stop words. There are stop words such as was, were, is, am, the, a, an, how, why, and many more. In Natural Language Processing, we eliminate the stop words to understand and analyze the meaning of a sentence. The removal of stop words is one of the most important tasks for search engines. Engineers design the algorithms of search engines in such a way that they ignore the use of stop words. This helps show the relevant search result for a query.

**3. List any two real-life applications of Natural Language Processing.**

Two real-life applications of Natural Language Processing are as follows:

Google Translate: Google Translate is one of the famous applications of Natural Language Processing. It helps convert written or spoken sentences into any language. Also, we can find the correct pronunciation and meaning of a word by using Google Translate. It uses advanced techniques of Natural Language Processing to achieve success in translating sentences into various languages.

Chatbots: To provide a better customer support service, companies have started using chatbots for 24/7 service. Chatbots helps resolve the basic queries of customers. If a chatbot is not able to resolve any query, then it forwards it to the support team, while still engaging the customer. It helps make customers feel that the customer support team is quickly attending them. With the help of chatbots, companies have become capable of building cordial relations with customers. It is only possible with the help of Natural Language Processing.

**4. What is TF-IDF?**

TFIDF or Term Frequency-Inverse Document Frequency indicates the importance of a word in a set. It helps in information retrieval with numerical statistics. For a specific document, TF-IDF shows a frequency that helps identify the keywords in a document. The major use of TF-IDF in NLP is the extraction of useful information from crucial documents by statistical data. It is ideally used to classify and summarize the text in documents and filter out stop words.

TF helps calculate the ratio of the frequency of a term in a document and the total number of terms. Whereas, IDF denotes the importance of the term in a document.

The formula for calculating TF-IDF:

TF(W) = (Frequency of W in a document)/(The total number of terms in the document)

IDF(W) = log\_e(The total number of documents/The number of documents having the term W)

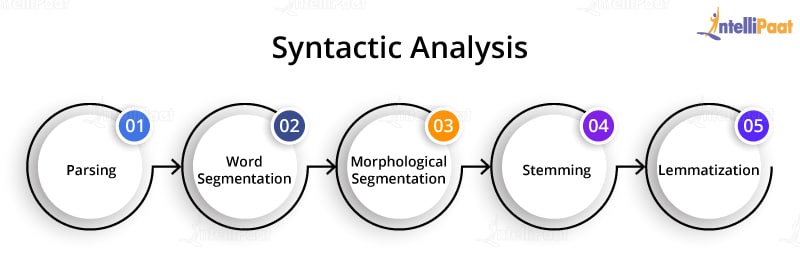
When TF\*IDF is high, the frequency of the term is less and vice versa.

Google uses TF-IDF to decide the index of search results according to the relevancy of pages. The design of the TF-IDF algorithm helps optimize the search results in Google. It helps quality content rank up in search results.

**5. What is Syntactic Analysis?**

Syntactic analysis is a technique of analyzing sentences to extract meaning from it. Using syntactic analysis, a machine can analyze and understand the order of words arranged in a sentence. NLP employs grammar rules of a language that helps in the syntactic analysis of the combination and order of words in documents.

The techniques used for syntactic analysis are as follows:



Parsing: It helps in deciding the structure of a sentence or text in a document. It helps analyze the words in the text based on the grammar of the language.

Word segmentation: The segmentation of words segregates the text into small significant units.

Morphological segmentation: The purpose of morphological segmentation is to break words into their base form.

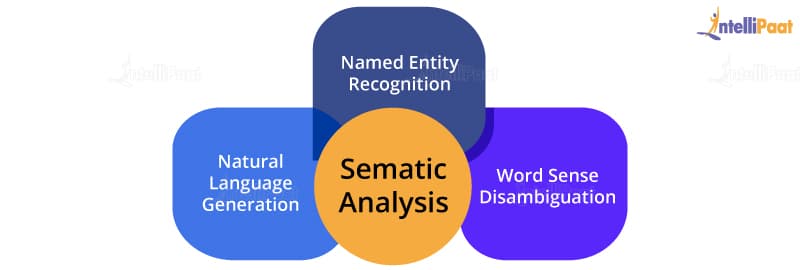
Stemming: It is the process of removing the suffix from a word to obtain its root word.

Lemmatization: It helps combine words using suffixes, without altering the meaning of the word.

**6. What is Semantic Analysis?**

Semantic analysis helps make a machine understand the meaning of a text. It uses various algorithms for the interpretation of words in sentences. It also helps understand the structure of a sentence.

Techniques used for semantic analysis are as given below:



Named entity recognition: This is the process of information retrieval that helps identify entities such as the name of a person, organization, place, time, emotion, etc.

Word sense disambiguation: It helps identify the sense of a word used in different sentences.

Natural language generation: It is a process used by the software to convert the structured data into human spoken languages. By using NLG, organizations can automate content for custom reports.

**7. What is NLTK?**

NLTK is a Python library, which stands for Natural Language Toolkit. We use NLTK to process data in human spoken languages. NLTK allows us to apply techniques such as parsing, tokenization, lemmatization, stemming, and more to understand natural languages. It helps in categorizing text, parsing linguistic structure, analyzing documents, etc.

A few of the libraries of the NLTK package that we often use in NLP are:

* SequentialBackoffTagger
* DefaultTagger
* UnigramTagger
* treebank
* wordnet
* FreqDist
* patterns
* RegexpTagger
* backoff\_tagger
* UnigramTagger, BigramTagger, and TrigramTagger

**8. How to tokenize a sentence using the nltk package?**

Tokenization is a process used in NLP to split a sentence into tokens. Sentence tokenization refers to splitting a text or paragraph into sentences.

For tokenizing, we will import sent\_tokenize from the nltk package:

from nltk.tokenize import sent\_tokenize<>

We will use the below paragraph for sentence tokenization:  
Para = “Hi Guys. Welcome to Intellipaat. This is a blog on the NLP interview questions and answers.”

sent\_tokenize(Para)

Output:

[ 'Hi Guys.' ,

'Welcome to Intellipaat. ',

'This is a blog on the NLP interview questions and answers. ' ]

Tokenizing a word refers to splitting a sentence into words.

Now, to tokenize a word, we will import word\_tokenize from the nltk package.

from nltk.tokenize import word\_tokenize

Para = “Hi Guys. Welcome to Intellipaat. This is a blog on the NLP interview questions and answers.”

word\_tokenize(Para)

Output:

[ 'Hi' , 'Guys' , ' . ' , 'Welcome' , 'to' , 'Intellipaat' , ' . ' , 'This' , 'is' , 'a', 'blog' , 'on' , 'the' , 'NLP' , 'interview' , 'questions' , 'and' , 'answers' , ' . ' ]

9. Explain how we can do parsing.

Parsing is the method to identify and understand the syntactic structure of a text. It is done by analyzing the individual elements of the text. The machine parses the text one word at a time, then two at a time, further three, and so on.

When the machine parses the text one word at a time, then it is a unigram.

When the text is parsed two words at a time, it is a bigram.

The set of words is a trigram when the machine parses three words at a time.

Look at the below diagram to understand unigram, bigram, and trigram.

Now, let’s implement parsing with the help of the nltk package.

import nltk

text = ”Top 30 NLP interview questions and answers”

We will now tokenize the text using word\_tokenize.

text\_token= word\_tokenize(text)

Now, we will use the function for extracting unigrams, bigrams, and trigrams.

list(nltk.unigrams(text))

Output:

[ "Top 30 NLP interview questions and answer"]

list(nltk.bigrams(text))

Output:

["Top 30", "30 NLP", "NLP interview", "interview questions", "questions and", "and answer"]

list(nltk.trigrams(text))

Output:

["Top 30 NLP", "NLP interview questions", "questions and answers"]

For extracting n-grams, we can use the function nltk.ngrams and give the argument n for the number of parsers.

list(nltk.ngrams(text,n))

10. Explain Stemming with the help of an example.

In Natural Language Processing, stemming is the method to extract the root word by removing suffixes and prefixes from a word.  
For example, we can reduce ‘stemming’ to ‘stem’ by removing ‘m’ and ‘ing.’  
We use various algorithms for implementing stemming, and one of them is PorterStemmer.  
First, we will import PorterStemmer from the nltk package.

from nltk.stem import PorterStemmer

Creating an object for PorterStemmer

pst=PorterStemmer()

pst.stem(“running”), pst.stem(“cookies”), pst.stem(“flying”)

Output:

(‘run’, ‘cooki', ‘fly’ )

11. Explain Lemmatization with the help of an example.

We use stemming and lemmatization to extract root words. However, stemming may not give the actual word, whereas lemmatization generates a meaningful word.  
In lemmatization, rather than just removing the suffix and the prefix, the process tries to find out the root word with its proper meaning.  
Example: ‘Bricks’ becomes ‘brick,’ ‘corpora’ becomes ‘corpus,’ etc.  
Let’s implement lemmatization with the help of some nltk packages.  
First, we will import the required packages.

from nltk.stem import wordnet

from nltk.stem import WordnetLemmatizer

Creating an object for WordnetLemmatizer()

lemma= WordnetLemmatizer()

list = [“Dogs”, “Corpora”, “Studies”]

for n in list:

print(n + “:” + lemma.lemmatize(n))

Output:

Dogs: Dog

Corpora: Corpus

Studies: Study

12. What is Parts-of-speech Tagging?

The parts-of-speech (POS) tagging is used to assign tags to words such as nouns, adjectives, verbs, and more. The software uses the POS tagging to first read the text and then differentiate the words by tagging. The software uses algorithms for the parts-of-speech tagging. POS tagging is one of the most essential tools in Natural Language Processing. It helps in making the machine understand the meaning of a sentence.  
We will look at the implementation of the POS tagging using stop words.  
Let’s import the required nltk packages.

import nltk

from nltk.corpus import stopwords

from nltk.tokenize import word\_tokenize, sent\_tokenize

stop\_words = set(stopwords.words('english'))

txt = "Sourav, Pratyush, and Abhinav are good friends."

Tokenizing using sent\_tokenize

tokenized\_text = sent\_tokenize(txt)

To find punctuation and words in a string, we will use word\_tokenizer and then remove the stop words.

for n in tokenized\_text:

wordsList = nltk.word\_tokenize(i)

wordsList = [w for w in wordsList if not w instop\_words]

Now, we will use the POS tagger.

tagged\_words = nltk.pos\_tag(wordsList)

print(tagged\_words)

Output:

[('Sourav', 'NNP'), ('Pratyush', 'NNP'), ('Abhinav', 'NNP'), ('good', 'JJ'), ('friends', 'NNS')]

13. Explain Named Entity Recognition by implementing it.

Named Entity Recognition (NER) is an information retrieval process. NER helps classify named entities such as monetary figures, location, things, people, time, and more. It allows the software to analyze and understand the meaning of the text. NER is mostly used in NLP, Artificial Intelligence, and Machine Learning. One of the real-life applications of NER is chatbots used for customer support.  
Let’s implement NER using the spacy package.  
Importing the spacy package:

import spacy

nlp = spacy.load('en\_core\_web\_sm')

Text = "The head office of Google is in California"

document = nlp(text)for ent in document.ents:

print(ent.text, ent.start\_char, ent.end\_char, ent.label\_)

Output:

Office 9 15 Place

Google 19 25 ORG

California 32 41 GPE

14. How to check word similarity using the spacy package?

To find out the similarity among words, we use word similarity. We evaluate the similarity with the help of a number that lies between 0 and 1. We use the spacy library to implement the technique of word similarity.

import spacy

nlp = spacy.load('en\_core\_web\_md')

print("Enter the words")

input\_words = input()

tokens = nlp(input\_words)

for i in tokens:

print(i.text, i.has\_vector, i.vector\_norm, i.is\_oov)

token\_1, token\_2 = tokens[0], tokens[1]

print("Similarity between words:", token\_1.similarity(token\_2))

Output:

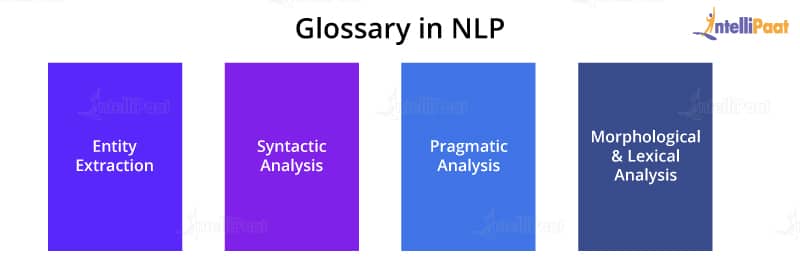
hot  True 5.6898586 False

cold True6.5396233 False

Similarity: 0.597265

This means that the similarity between the words ‘hot’ and ‘cold’ is just 59 percent.  
15. List the components of Natural Language Processing.

The major components of NLP are as follows:



Entity extraction: Entity extraction refers to the retrieval of information such as place, person, organization, etc. by the segmentation of a sentence. It helps in the recognition of an entity in a text.

Syntactic analysis: Syntactic analysis helps draw the specific meaning of a text.

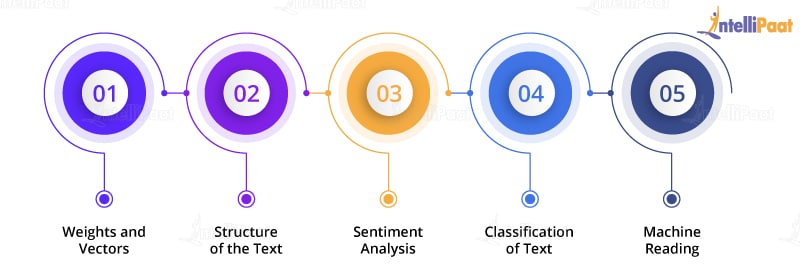
Pragmatic analysis: To find useful information from a text, we implement pragmatic analysis techniques.

Morphological and lexical analysis: It helps in explaining the structure of words by analyzing them through parsing.

16. Define the terminology in NLP.

This is one of the most often asked NLP interview questions.

The interpretation of Natural Language Processing depends on various factors, and they are:



* Weights and Vectors
* Use of TF-IDF for information retrieval
* Length (TF-IDF and doc)
* Google Word Vectors
* Word Vectors
* Structure of the Text
* POS tagging
* Head of the sentence
* Named Entity Recognition (NER)
* Sentiment Analysis
* Knowledge of the characteristics of sentiment
* Knowledge about entities and the common dictionary available for sentiment analysis
* Classification of Text
* Supervised learning algorithm
* Training set
* Validation set
* Test set
* Features of the text
* LDA
* Machine Reading
* Removal of possible entities
* Joining with other entities
* DBpedia
* FRED (lib) Pikes

17. What is Latent Semantic Indexing (LSI)?

Latent semantic indexing is a mathematical technique used to improve the accuracy of the information retrieval process. The design of LSI algorithms allows machines to detect the hidden (latent) correlation between semantics (words). To enhance information understanding, machines generate various concepts that associate with the words of a sentence.

The technique used for information understanding is called singular value decomposition. It is generally used to handle static and unstructured data. The matrix obtained for singular value decomposition contains rows for words and columns for documents. This method best suits to identify components and group them according to their types.

The main principle behind LSI is that words carry a similar meaning when used in a similar context. Computational LSI models are slow in comparison to other models. However, they are good at contextual awareness that helps improve the analysis and understanding of a text or a document.

18. What are Regular Expressions?

A regular expression is used to match and tag words. It consists of a series of characters for matching strings.

Suppose, if A and B are regular expressions, then the following are true for them:

If {ɛ} is a regular language, then ɛ is a regular expression for it.

If A and B are regular expressions, then A + B is also a regular expression within the language {A, B}.

If A and B are regular expressions, then the concatenation of A and B (A.B) is a regular expression.

If A is a regular expression, then A\* (A occurring multiple times) is also a regular expression.

19. What is Regular Grammar?

Regular grammar is used to represent a regular language.

A regular grammar comprises rules in the form of A -> a, A -> aB, and many more. The rules help detect and analyze strings by automated computation.

Regular grammar consists of four tuples:

‘N’ is used to represent the non-terminal set.

‘∑’ represents the set of terminals.

‘P’ stands for the set of productions.

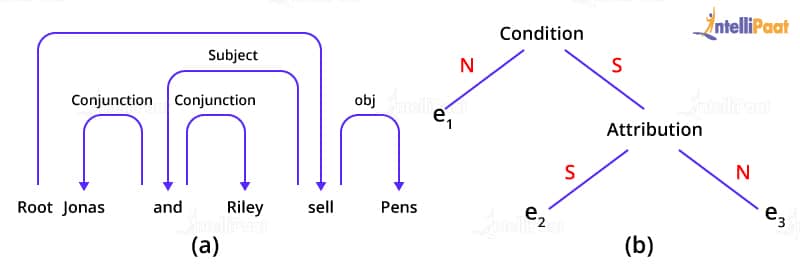
‘S € N’ denotes the start of non-terminal.

20. Explain Dependency Parsing in NLP.

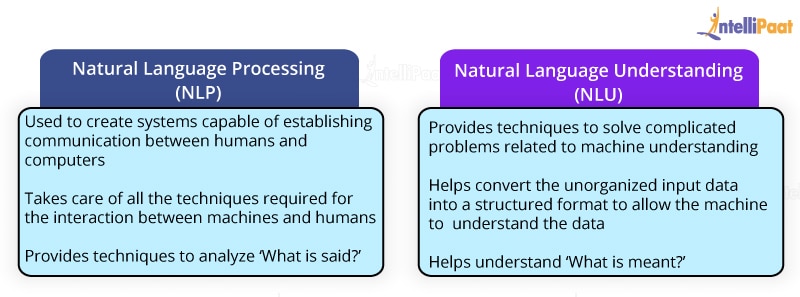
Dependency parsing helps assign a syntactic structure to a sentence. Therefore, it is also called syntactic parsing. Dependency parsing is one of the critical tasks in NLP. It allows the analysis of a sentence using parsing algorithms. Also, by using the parse tree in dependency parsing, we can check the grammar and analyze the semantic structure of a sentence.

For implementing dependency parsing, we use the spacy package. It implements token properties to operate the dependency parse tree.

The below diagram shows the dependency parse tree:

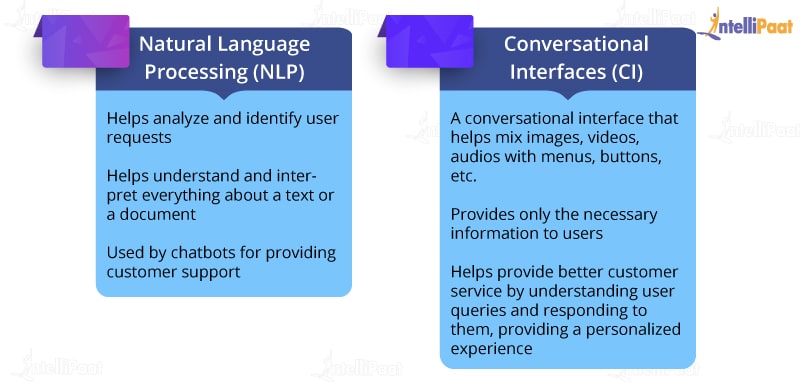


21. What is the difference between NLP and NLU?

The below table shows the difference between NLP and NLU:

22. What is the difference between NLP and CI?

The below table shows the difference between NLP and CI:



23. What is Pragmatic Analysis?

Pragmatic analysis is an important task in NLP for interpreting knowledge that is lying outside a given document. The aim of implementing pragmatic analysis is to focus on exploring a different aspect of the document or text in a language. This requires a comprehensive knowledge of the real world. The pragmatic analysis allows software applications for the critical interpretation of the real-world data to know the actual meaning of sentences and words.

Example:

Consider this sentence: ‘Do you know what time it is?’

This sentence can either be asked for knowing the time or for yelling at someone to make them note the time. This depends on the context in which we use the sentence.

24. What is Pragmatic Ambiguity?

Pragmatic ambiguity refers to the multiple descriptions of a word or a sentence. An ambiguity arises when the meaning of the sentence is not clear. The words of the sentence may have different meanings. Therefore, in practical situations, it becomes a challenging task for a machine to understand the meaning of a sentence. This leads to pragmatic ambiguity.

Example:

Check out the below sentence.

‘Are you feeling hungry?’

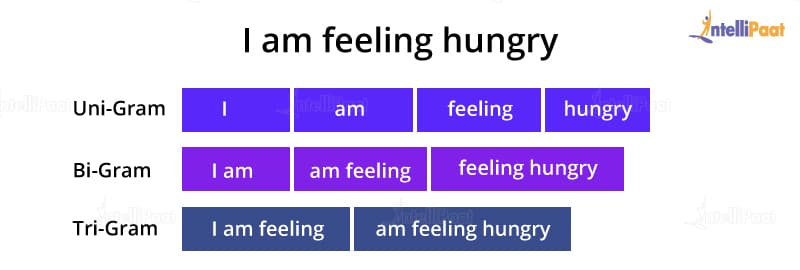
The given sentence could be either a question or a formal way of offering food.

25. What are unigrams, bigrams, trigrams, and n-grams in NLP?

When we parse a sentence one word at a time, then it is called a unigram. The sentence parsed two words at a time is a bigram.

When the sentence is parsed three words at a time, then it is a trigram. Similarly, n-gram refers to the parsing of n words at a time.

Example: To understand unigrams, bigrams, and trigrams, you can refer to the below diagram:



Therefore, parsing allows machines to understand the individual meaning of a word in a sentence. Also, this type of parsing helps predict the next word and correct spelling errors.

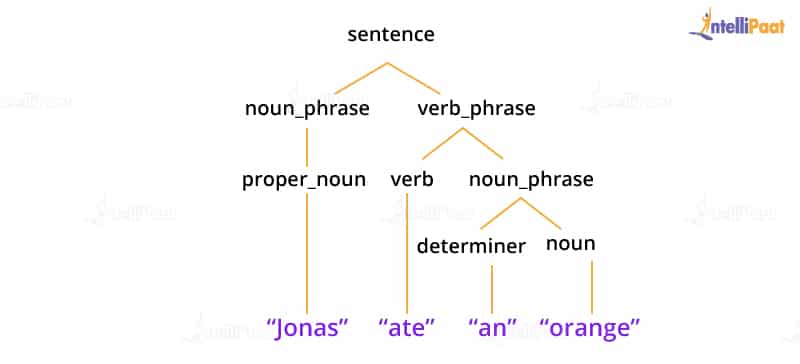
26. What are the steps involved in solving an NLP problem?

Below are the steps involved in solving an NLP problem:

* Gather the text from the available dataset or by web scraping
* Apply stemming and lemmatization for text cleaning
* Apply feature engineering techniques
* Embed using word2vec
* Train the built model using neural networks or other Machine Learning techniques
* Evaluate the model’s performance
* Make appropriate changes in the model
* Deploy the model

27. What is Parsing in the context of NLP?

Parsing in NLP refers to the understanding of a sentence and its grammatical structure by a machine. Parsing allows the machine to understand the meaning of a word in a sentence and the grouping of words, phrases, nouns, subjects, and objects in a sentence. Parsing helps analyze the text or the document to extract useful insights from it. To understand parsing, refer to the below diagram:



In this, ‘Jonas ate an orange’ is parsed to understand the structure of the sentence.

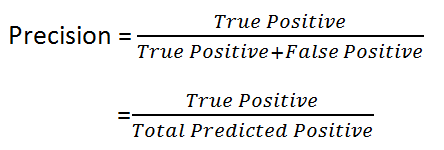
28. What is Feature Extraction in NLP?

Features or characteristics of a word help in text or document analysis. They also help in sentiment analysis of a text. Feature extraction is one of the techniques that are used by recommendation systems. Reviews such as ‘excellent,’ ‘good,’ or ‘great’ for a movie are positive reviews, recognized by a recommender system. The recommender system also tries to identify the features of the text that help in describing the context of a word or a sentence. Then, it makes a group or category of the words that have some common characteristics. Now, whenever a new word arrives, the system categorizes it as per the labels of such groups.

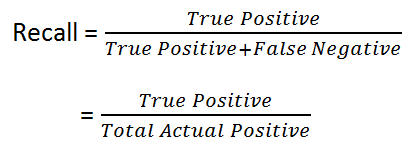
29. What is precision and recall?

The metrics used to test an NLP model are precision, recall, and F1. Also, we use accuracy for evaluating the model’s performance. The ratio of prediction and the desired output yields the accuracy of the model.

Precision is the ratio of true positive instances and the total number of positively predicted instances.

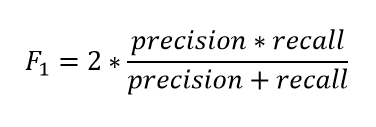


Recall is the ratio of true positive instances and the total actual positive instances.



30. What is F1 score in NLP?

F1 score evaluates the weighted average of recall and precision. It considers both false negative and false positive instances while evaluating the model. F1 score is more accountable than accuracy for an NLP model when there is an uneven distribution of class. Let us look at the formula for calculating F1 score:



1. Which of the following techniques can be used for keyword normalization in NLP, the process of converting a keyword into its base form?  
a. Lemmatization  
b. Soundex  
c. Cosine Similarity  
d. N-grams  
Answer: a)  
Lemmatization helps to get to the base form of a word, e.g. are playing -> play, eating -> eat, etc..  
Other options are meant for different purposes.  
  
2. Which of the following techniques can be used to compute the distance between two word vectors in NLP?  
a. Lemmatization  
b. Euclidean distance  
c. Cosine Similarity  
d. N-grams  
Answer: b) and c)  
Distance between two word vectors can be computed using Cosine similarity and Euclidean Distance.  Cosine Similarity establishes a cosine angle between the vector of two words. A cosine angle close to each other between two word vectors indicates the words are similar and vice a versa.  
E.g. cosine angle between two words “Football” and “Cricket” will be closer to 1 as compared to angle between the words “Football” and “New Delhi”  
  
Python code to implement CosineSimlarity function would look like this  
def cosine\_similarity(x,y):  
    return np.dot(x,y)/( np.sqrt(np.dot(x,x)) \* np.sqrt(np.dot(y,y)) )  
q1 = wikipedia.page(‘Strawberry’)  
q2 = wikipedia.page(‘Pineapple’)  
q3 = wikipedia.page(‘Google’)  
q4 = wikipedia.page(‘Microsoft’)  
cv = CountVectorizer()  
X = np.array(cv.fit\_transform([q1.content, q2.content, q3.content, q4.content]).todense())  
print (“Strawberry Pineapple Cosine Distance”, cosine\_similarity(X[0],X[1]))  
print (“Strawberry Google Cosine Distance”, cosine\_similarity(X[0],X[2]))  
print (“Pineapple Google Cosine Distance”, cosine\_similarity(X[1],X[2]))  
print (“Google Microsoft Cosine Distance”, cosine\_similarity(X[2],X[3]))  
print (“Pineapple Microsoft Cosine Distance”, cosine\_similarity(X[1],X[3]))  
Strawberry Pineapple Cosine Distance 0.8899200413701714  
Strawberry Google Cosine Distance 0.7730935582847817  
Pineapple Google Cosine Distance 0.789610214147025  
Google Microsoft Cosine Distance 0.8110888282851575  
Usually Document similarity is measured by how close semantically the content (or words) in the document are to each other. When they are close, the similarity index is close to 1, otherwise near 0.  
The Euclidean distance between two points is the length of the shortest path connecting them. Usually computed using Pythagoras theorem for a triangle.

3. What are the possible features of a text corpus in NLP?  
a. Count of the word in a document  
b. Vector notation of the word  
c. Part of Speech Tag  
d. Basic Dependency Grammar  
e. All of the above  
Answer: e)  
All of the above can be used as features of the text corpus.  
  
4. You created a document term matrix on the input data of 20K documents for a Machine learning model. Which of the following can be used to reduce the dimensions of data?

Keyword Normalization

Latent Semantic Indexing

Latent Dirichlet Allocation

a. only 1  
b. 2, 3  
c. 1, 3  
d. 1, 2, 3  
Answer: d)  
  
5. Which of the text parsing techniques can be used for noun phrase detection, verb phrase detection, subject detection, and object detection in NLP.  
a. Part of speech tagging  
b. Skip Gram and N-Gram extraction  
c. Continuous Bag of Words  
d. Dependency Parsing and Constituency Parsing  
Answer: d)  
  
6. Dissimilarity between words expressed using cosine similarity will have values significantly higher than 0.5  
a. True  
b. False  
Ans: a)  
  
7. Which one of the following are keyword Normalization techniques in NLP  
a.  Stemming  
b.  Part of Speech  
c. Named entity recognition  
d. Lemmatization  
Answer: a) and d)  
Part of Speech (POS) and Named Entity Recognition(NER) are not keyword Normalization techniques. Named Entity help you extract Organization, Time, Date, City, etc..type of entities from the given sentence, whereas Part of Speech helps you extract Noun, Verb, Pronoun, adjective, etc..from the given sentence tokens.  
  
8. Which of the below are NLP use cases?  
a. Detecting objects from an image  
b. Facial Recognition  
c. Speech Biometric  
d. Text Summarization  
Ans: d)  
a) And b) are Computer Vision use cases, and c) is Speech use case.  
Only d) Text Summarization is an NLP use case.  
  
9. In a corpus of N documents, one randomly chosen document contains a total of T terms and the term “hello” appears K times.  
What is the correct value for the product of TF (term frequency) and IDF (inverse-document-frequency), if the term “hello” appears in approximately one-third of the total documents?  
a. KT \* Log(3)  
b. T \* Log(3) / K  
c. K \* Log(3) / T  
d. Log(3) / KT  
Answer: (c)  
formula for TF is K/T  
formula for IDF is log(total docs / no of docs containing “data”)  
= log(1 / (⅓))  
= log (3)  
Hence correct choice is Klog(3)/T  
  
10. In NLP, The algorithm decreases the weight for commonly used words and increases the weight for words that are not used very much in a collection of documents  
a. Term Frequency (TF)  
b. Inverse Document Frequency (IDF)  
c. Word2Vec  
d. Latent Dirichlet Allocation (LDA)  
Ans: b)

Deep Learning Interview Questions

Deep Learning is a subset of machine learning. It has taken over the world as a leading technology today, and imitates the complex functionalities of the human brain to use unstructured data to decipher meaning and teach machines. Be it the automotive, healthcare, or content creation industry, the applications of deep learning are on the rise. We have complied a list of Deep Learning Interview Questions to help you prepare.

What are the different types of Activation Functions?

Linear or Identity,  
Unit or Binary Step,  
Sigmoid or Logistic,  
Tanh,  
ReLU, and  
Softmax, [read more.](https://www.mygreatlearning.com/blog/deep-learning-interview-questions/)

What is backpropagation?

Backpropagation is a training algorithm which is used for a multilayer neural network. In backpropagation, the error is moved from the end of the network to all weights. This allows efficient computing of the gradient. [Read more.](https://www.mygreatlearning.com/blog/deep-learning-interview-questions/)

11. In NLP, The process of removing words like “and”, “is”, “a”, “an”, “the” from a sentence is called as  
a. Stemming  
b. Lemmatization  
c. Stop word  
d. All of the above  
Ans: c)   
In Lemmatization, all the stop words such as a, an, the, etc.. are removed. One can also define custom stop words for removal.  
  
12. In NLP, The process of converting a sentence or paragraph into tokens is referred to as Stemming  
a. True  
b. False  
Ans: b)  
The statement describes the process of tokenization and not stemming, hence it is False.  
  
13. In NLP, Tokens are converted into numbers before giving to any Neural Network  
a. True  
b. False  
Ans: a)  
In NLP, all words are converted into a number before feeding to a Neural Network.  
  
14. identify the odd one out  
a. nltk  
b. scikit learn  
c. SpaCy  
d. BERT  
Ans: d)  
All the ones mentioned are NLP libraries except BERT, which is a word embedding  
  
15. TF-IDF helps you to establish?  
a. most frequently occurring word in the document  
b. most important word in the document  
Ans: b)

TF-IDF helps to establish how important a particular word is in the context of the document corpus. TF-IDF takes into account the number of times the word appears in the document and offset by the number of documents that appear in the corpus.

TF is the frequency of term divided by a total number of terms in the document.

IDF is obtained by dividing the total number of documents by the number of documents containing the term and then taking the logarithm of that quotient.

Tf.idf is then the multiplication of two values TF and IDF.

Suppose that we have term count tables of a corpus consisting of only two documents, as listed here

|  |  |  |
| --- | --- | --- |
| Term | Document 1 Frequency | Document 2 Frequency |
| This | 1 | 1 |
| is | 1 | 1 |
| a | 2 |  |
| Sample | 1 |  |
| another |  | 2 |
| example |  | 3 |

The calculation of tf–idf for the term “this” is performed as follows:  
for “this”  
———–  
tf(“this”, d1) = 1/5 = 0.2  
tf(“this”, d2) = 1/7 = 0.14  
idf(“this”, D) = log (2/2) =0  
hence tf-idf  
tfidf(“this”, d1, D) = 0.2\* 0 = 0  
tfidf(“this”, d2, D) = 0.14\* 0 = 0  
for “example”  
————  
tf(“example”, d1) = 0/5 = 0  
tf(“example”, d2) = 3/7 = 0.43  
idf(“example”, D) = log(2/1) = 0.301  
tfidf(“example”, d1, D) = tf(“example”, d1) \* idf(“example”, D) = 0 \* 0.301 = 0  
tfidf(“example”, d2, D) = tf(“example”, d2) \* idf(“example”, D) = 0.43 \* 0.301 = 0.129  
In its raw frequency form, TF is just the frequency of the “this” for each document. In each document, the word “this” appears once; but as document 2 has more words, its relative frequency is smaller.  
An IDF is constant per corpus, and accounts for the ratio of documents that include the word “this”. In this case, we have a corpus of two documents and all of them include the word “this”. So TF–IDF is zero for the word “this”, which implies that the word is not very informative as it appears in all documents.  
The word “example” is more interesting – it occurs three times, but only in the second document.  
  
16. In NLP, The process of identifying people, an organization from a given sentence, paragraph is called  
a. Stemming  
b. Lemmatization  
c. Stop word removal  
d. Named entity recognition  
Ans: d)  
  
17. Which one of the following is not a pre-processing technique in NLP  
a. Stemming and Lemmatization  
b. converting to lowercase  
c. removing punctuations  
d. removal of stop words  
e. Sentiment analysis  
Ans: e)  
Sentiment Analysis is not a pre-processing technique. It is done after pre-processing and is an NLP use case. All other listed ones are used as part of statement pre-processing.  
  
18. In text mining, converting text into tokens and then converting them into an integer or floating-point vectors can be done using  
a. CountVectorizer  
b.  TF-IDF  
c. Bag of Words  
d. NERs  
Ans: a)  
CountVectorizer helps do the above, while others are not applicable.  
text =[“Rahul is an avid writer, he enjoys studying understanding and presenting. He loves to play”]  
vectorizer = CountVectorizer()  
vectorizer.fit(text)  
vector = vectorizer.transform(text)  
print(vector.toarray())

output   
[[1 1 1 1 2 1 1 1 1 1 1 1 1 1]]  
The second section of the interview questions covers advanced NLP techniques such as Word2Vec, GloVe  word embeddings, and advanced models such as GPT, ELMo, BERT, XLNET based questions, and explanations.  
  
19. In NLP, Words represented as vectors are called as Neural Word Embeddings  
a. True  
b. False  
Ans: a)  
Word2Vec, GloVe based models build word embedding vectors that are multidimensional.  
  
20. In NLP, Context modeling is supported with which one of the following word embeddings

a. Word2Vec

b) GloVe

c) BERT

d) All of the above

Ans: c)  
Only BERT (Bidirectional Encoder Representations from Transformer) supports context modelling where the previous and next sentence context is taken into consideration. In Word2Vec, GloVe only word embeddings are considered and previous and next sentence context is not considered.  
  
21. In NLP, Bidirectional context is supported by which of the following embedding  
a. Word2Vec  
b. BERT  
c. GloVe  
d. All the above  
Ans: b)  
Only BERT provides a bidirectional context. The BERT model uses the previous and the next sentence to arrive at the context.Word2Vec and GloVe are word embeddings, they do not provide any context.  
  
22. Which one of the following Word embeddings can be custom trained for a specific subject in NLP  
a. Word2Vec  
b. BERT  
c. GloVe  
d. All the above  
Ans: b)  
BERT allows Transform Learning on the existing pre-trained models and hence can be custom trained for the given specific subject, unlike Word2Vec and GloVe where existing word embeddings can be used, no transfer learning on text is possible.  
  
23. Word embeddings capture multiple dimensions of data and are represented as vectors  
a. True  
b. False  
Ans: a)  
  
24. In NLP, Word embedding vectors help establish distance between two tokens  
a. True  
b. False  
Ans: a)  
One can use Cosine similarity to establish distance between two vectors represented through Word Embeddings  
  
25. Language Biases are introduced due to historical data used during training of word embeddings, which one amongst the below is not an example of bias  
a. New Delhi is to India, Beijing is to China  
b. Man is to Computer, Woman is to Homemaker  
Ans: a)  
Statement b) is a bias as it buckets Woman into Homemaker, whereas statement a) is not a biased statement.  
  
26. Which of the following will be a better choice to address NLP use cases such as semantic similarity, reading comprehension, and common sense reasoning  
a. ELMo  
b. Open AI’s GPT  
c. ULMFit  
Ans: b)  
Open AI’s GPT is able to learn complex pattern in data by using the Transformer models Attention mechanism and hence is more suited for complex use cases such as semantic similarity, reading comprehensions, and common sense reasoning.  
  
27. Transformer architecture was first introduced with?  
a. GloVe  
b. BERT  
c. Open AI’s GPT  
d. ULMFit  
Ans: c)  
ULMFit has an LSTM based Language modeling architecture. This got replaced into Transformer architecture with Open AI’s GPT  
  
28. Which of the following architecture can be trained faster and needs less amount of training data  
a. LSTM based Language Modelling  
b. Transformer architecture  
Ans: b)  
Transformer architectures were supported from GPT onwards and were faster to train and needed less amount of data for training too.

**Python Interview Questions**

Python has emerged to be one of the most widely used programming languages today. Individuals with Python skills are in high-demand and recruiters are looking to hire professionals who possess Python Programming knowledge. Interviewing for Python can be intimidating. Thus, we have complied a list of the Top Python Interview Questions to help you prepare well.

What are Keywords in Python?

In Python, Keywords are reserved words that are used as identifiers, function name or variable name. They are used to define the structure and syntax of the language.

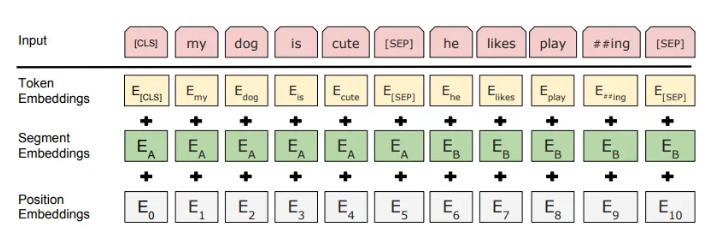
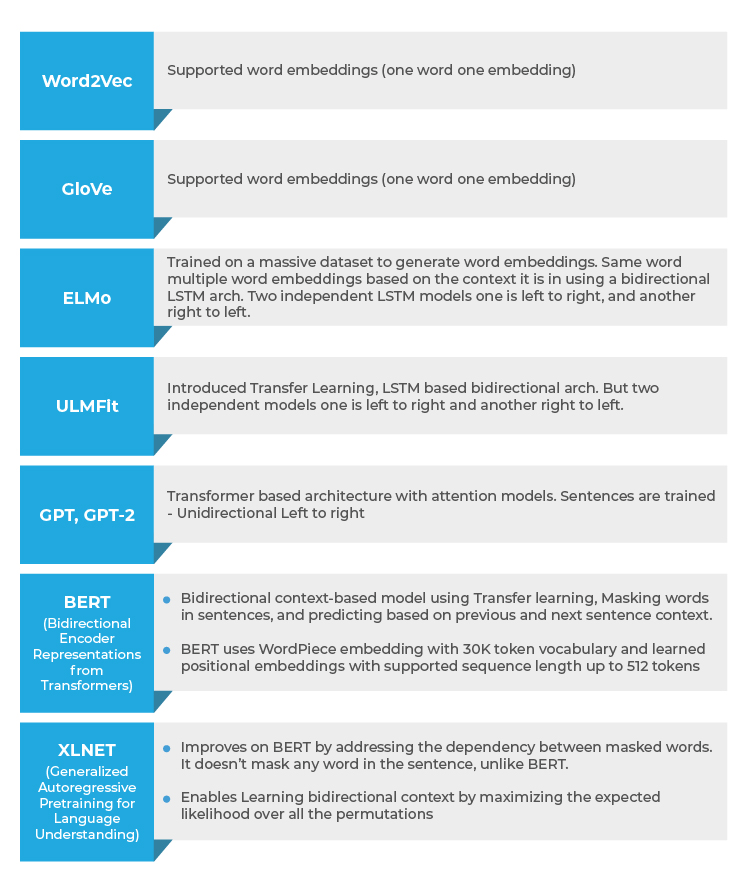
There are a total of 33 keywords in Python 3.7. A list of all the keywords is provided below. [Read More.](https://www.mygreatlearning.com/blog/python-interview-questions/)

What are dataframes?

A pandas dataframe in Python is a data structure that is mutable. Pandas has support for heterogeneous data that is arranged across two axes (rows and columns). [Read more.](https://www.mygreatlearning.com/blog/python-interview-questions/)

29. Same word can have multiple word embeddings possible with \_\_\_\_\_\_\_\_\_\_\_\_?  
a. GloVe  
b. Word2Vec  
c. ELMo  
d. nltk  
Ans: c)

EMLo word embeddings supports same word with multiple embeddings, this helps in using the same word in a different context and thus captures the context than just meaning of the word unlike in GloVe and Word2Vec. Nltk is not a word embedding.



30. For a given token, its input representation is the sum of embedding from the token, segment and position embedding  
a. ELMo  
b. GPT  
c. BERT  
d. ULMFit  
Ans: c)  
BERT uses token, segment and position embedding.  
  
  
31. Trains two independent LSTM language model left to right and right to left and shallowly concatenates them  
a. GPT  
b. BERT  
c. ULMFit  
d. ELMo  
Ans: d)  
ELMo tries to train two independent LSTM language models (left to right and right to left) and concatenates the results to produce word embedding.

32. Uses unidirectional language model for producing word embedding  
a. BERT  
b. GPT  
c. ELMo  
d. Word2Vec  
Ans: b)   
GPT is a idirectional model and word embedding are produced by training on information flow from left to right. ELMo is bidirectional but shallow. Word2Vec provides simple word embedding.

33. In this architecture, the relationship between all words in a sentence is modelled irrespective of their position. Which architecture is this?  
a. OpenAI GPT  
b. ELMo  
c. BERT  
d. ULMFit  
Ans: c)  
BERT Transformer architecture models the relationship between each word and all other words in the sentence to generate attention scores. These attention scores are later used as weights for a weighted average of all words’ representations which is fed into a fully-connected network to generate a new representation.  
  
34. List 10 use cases to be solved using NLP techniques?

Sentiment Analysis

Language Translation (English to German, Chinese to English, etc..)

Document Summarization

Question Answering

Sentence Completion

Attribute extraction (Key information extraction from the documents)

Chatbot interactions

Topic classification

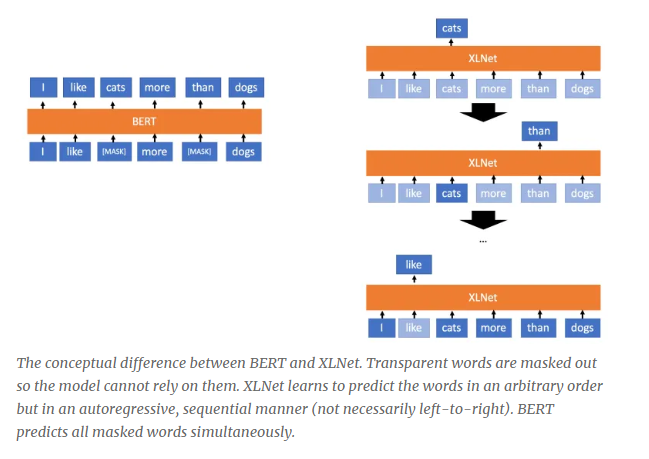
Intent extraction

Grammar or Sentence correction

Image captioning

Document Ranking

Natural Language inference



35. Transformer model pays attention to the most important word in Sentence  
a. True  
b. False  
Ans: a) Attention mechanisms in the Transformer model are used to model the relationship between all words and also provide weights to the most important word.  
  
36. Which NLP model gives the best accuracy amongst the following?  
a. BERT  
b. XLNET  
c. GPT-2  
d. ELMo  
Ans: b) XLNET  
XLNET has given best accuracy amongst all the models. It has outperformed BERT on 20 tasks and achieves state of art results on 18 tasks including sentiment analysis, question answering, natural language inference, etc.  
  
37. Permutation Language models is a feature of  
a. BERT  
b. EMMo  
c. GPT  
d. XLNET  
Ans: d)   
XLNET provides permutation-based language modelling and is a key difference from BERT. In permutation language modeling, tokens are predicted in a random manner and not sequential. The order of prediction is not necessarily left to right and can be right to left. The original order of words is not changed but a prediction can be random.   
The conceptual difference between BERT and XLNET can be seen from the following diagram.  
  
38. Transformer XL uses relative positional embedding  
a. True  
b. False  
Ans: a)  
Instead of embedding having to represent the absolute position of a word, Transformer XL uses an embedding to encode the relative distance between the words. This embedding is used to compute the attention score between any 2 words that could be separated by n words before or after.

39. What is Naive Bayes algorithm, When we can use this algorithm in NLP?

Naive Bayes algorithm is a collection of classifiers which works on the principles of the Bayes’ theorem. This series of NLP model forms a family of algorithms that can be used for a wide range of classification tasks including sentiment prediction, filtering of spam, classifying documents and more.

Naive Bayes algorithm converges faster and requires less training data. Compared to other discriminative models like logistic regression, Naive Bayes model it takes lesser time to train. This algorithm is perfect for use while working with multiple classes and text classification where the data is dynamic and changes frequently.

40. Explain Dependency Parsing in NLP?

Dependency Parsing, also known as Syntactic parsing in NLP is a process of assigning syntactic structure to a sentence and identifying its dependency parses. This process is crucial to understand the correlations between the “head” words in the syntactic structure.  
The process of dependency parsing can be a little complex considering how any sentence can have more than one dependency parses. Multiple parse trees are known as ambiguities. Dependency parsing needs to resolve these ambiguities in order to effectively assign a syntactic structure to a sentence.

Dependency parsing can be used in the semantic analysis of a sentence apart from the syntactic structuring.

41. What is text Summarization?

Text summarization is the process of shortening a long piece of text with its meaning and effect intact. Text summarization intends to create a summary of any given piece of text and outlines the main points of the document. This technique has improved in recent times and is capable of summarizing volumes of text successfully.

Text summarization has proved to a blessing since machines can summarise large volumes of text in no time which would otherwise be really time-consuming. There are two types of text summarization:

Extraction-based summarization

Abstraction-based summarization

42. What is NLTK? How is it different from Spacy?

NLTK or Natural Language Toolkit is a series of libraries and programs that are used for symbolic and statistical natural language processing. This toolkit contains some of the most powerful libraries that can work on different ML techniques to break down and understand human language. NLTK is used for Lemmatization, Punctuation, Character count, Tokenization, and Stemming. The difference between NLTK and Spacey are as follows:

While NLTK has a collection of programs to choose from, Spacey contains only the best-suited algorithm for a problem in its toolkit

NLTK supports a wider range of languages compared to Spacey (Spacey supports only 7 languages)

While Spacey has an object-oriented library, NLTK has a string processing library

Spacey can support word vectors while NLTK cannot

43. What is information extraction?

Information extraction in the context of Natural Language Processing refers to the technique of extracting structured information automatically from unstructured sources to ascribe meaning to it. This can include extracting information regarding attributes of entities, relationship between different entities and more. The various models of information extraction includes:

Tagger Module

Relation Extraction Module

Fact Extraction Module

Entity Extraction Module

Sentiment Analysis Module

Network Graph Module

Document Classification & Language Modeling Module

44. What is Bag of Words?

Bag of Words is a commonly used model that depends on word frequencies or occurrences to train a classifier. This model creates an occurrence matrix for documents or sentences irrespective of its grammatical structure or word order.

45. What is Pragmatic Ambiguity in NLP?

Pragmatic ambiguity refers to those words which have more than one meaning and their use in any sentence can depend entirely on the context. Pragmatic ambiguity can result in multiple interpretations of the same sentence. More often than not, we come across sentences which have words with multiple meanings, making the sentence open to interpretation. This multiple interpretation causes ambiguity and is known as Pragmatic ambiguity in NLP.

46. What is Masked Language Model?

Masked language models help learners to understand deep representations in downstream tasks by taking an output from the corrupt input. This model is often used to predict the words to be used in a sentence.

47. What is the difference between NLP and CI(Conversational Interface)?

The difference between NLP and CI is as follows:

|  |  |
| --- | --- |
| Natural Language Processing | Conversational Interface |
| NLP attempts to help machines understand and learn how language concepts work. | CI focuses only on providing users with an interface to interact with. |
| NLP uses AI technology to identify, understand, and interpret the requests of users through language. | CI uses voice, chat, videos, images and more such conversational aid to create the user interface. |

48. What are the best NLP Tools?

Some of the best NLP tools from open sources are:

SpaCy

TextBlob

Textacy

Natural language Toolkit

Retext

NLP.js

Stanford NLP

CogcompNLP

49. What is POS tagging?

Parts of speech tagging better known as POS tagging refers to the process of identifying specific words in a document and group them as part of speech, based on its context. POS tagging is also known as grammatical tagging since it involves understanding grammatical structures and identifying the respective component.

POS tagging is a complicated process since the same word can be different parts of speech depending on the context. The same generic process used for word mapping is quite ineffective for POS tagging because of the same reason.

50. What is NES?

Name entity recognition is more commonly known as NER is the process of identifying specific entities in a text document which are more informative and have a unique context. These often denote places, people, organisations, and more. Even though it seems like these entities are proper nouns, the NER process is far from identifying just the nouns. In fact, NER involves entity chunking or extraction wherein entities are segmented to categorise them under different predefined classes. This step further helps in extracting information.

|  |
| --- |
| What is NLP(natural language processing) ? |
| Natural language processing is a subfield of computer science, information engineering, and artificial intelligence concerned with the interactions between computers and human languages, in particular how to program computers to process and analyze large amounts of natural language data |
| What is applications of NLP ? |
| Text classification, Text summarization, Name entity recognization, part of speech tagging, language model building, Machine translation, Spell checking, speech recognization, character recognization. |
| What is tokenization ? |
| Splitting the sentence into words |
| What is stemming ? |
| Stemming is the process of reducing a word to its word stem that affixes to suffixes and prefixes. |
| What is lemmatizing ? |
| Lemmatizing is also same like stemming but the difference is lemmantizing words known with dictionary. |
| What is Normalization ? |
| Converting different range of values to same scale from 0 to 1. |
| What is POS (parts of speech) tagging ? |
| Tagging a word with noun, pronoun, adverd, adjective etc. |
| What is NER (name entity recognition)? |
| NER refers to name entiyy recognization like places, organizations, companies etc. |
| What are nlp libraries and tools ? |
|  |
| CoreNLP from Stanford group. |
| NLTK, the most widely-mentioned NLP library for Python. |
| TextBlob, a user-friendly and intuitive NLTK interface. |
| Gensim, a library for document similarity analysis. |
| SpaCy, an industrial-strength NLP library built for performance. |
|  |
| What are stop words ? |
| a, the , an etc like repeated words in text, that doesn’t give any additional value to context. we can filter those words by using nltk library standard function. |
| What are punctuation’s ? How can you remove it ? |
|  |
| What is Noise Removal ? |
| Remove unwanted data from corpus. Like if you are working sentiment analysis, we have to remove ?”! etc. |
| What is Wordnet ? |
| WordNet is a lexical database for the English language. It groups English words into sets of synonyms called synsets, provides short definitions and usage examples, and records a number of relations among these synonym sets or their members. |
| How can you find synonyms and antonyms for a word ? |
| [Refer here](https://onlinecoursetutorials.com/nlp/synonyms-and-antonyms-from-nltk-wordnet-in-python/) |
| What is NLG (Natural language Generation) ? |
| It’s about generating new text from understanding old data. |
| What is NLU (Natural language understanding) ? |
| It’s about understanding of natural language. How humans are communicating in different scenarios. |
| What is Corpus ? |
| It’s a collection of text documents. |
| What is N- Gram, Unigram, Bigram  and Trigram? |
| it’s about word analysis, unigram means single word, bigram means double words and trigram means tripple word. |
| What is Language modeling ? |
| A statistical language model is a probability distribution over sequences of words. Given such a sequence, say of length m, it assigns a probability to the whole sequence. The language model provides context to distinguish between words and phrases that sound simila |
| What is Latent semantic analysis ? |
| Latent semantic analysis is a technique in natural language processing, in particular distributional semantics, of analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms |
| What is word embedding ? |
| Word embedding is the collective name for a set of language modeling and feature learning techniques in natural language processing where words or phrases from the vocabulary are mapped to vectors of real numbers |
| What are word embedding libraries ? |
|  |
| Word2vec |
| Glove |
| Fasttext |
| genism |
|  |
| What is word2vec ? |
| Word2vec is a group of related models that are used to produce word embeddings. These models are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words |
| What is Glove ? |
| GloVe, coined from Global Vectors, is a model for distributed word representation. The model is an unsupervised learning algorithm for obtaining vector representations for words. This is achieved by mapping words into a meaningful space where the distance between words is related to semantic similarity. |
| What is Fasttext ? |
| fastText is a library for learning of word embeddings and text classification created by Facebook’s AI Research lab. The model allows to create an unsupervised learning or supervised learning algorithm for obtaining vector representations for words |
| What is Genism ? |
| Gensim is a production-ready open-source library for unsupervised topic modeling and natural language processing, using modern statistical machine learning. Gensim is implemented in Python and Cython for top performance and scalability |
| What is text mining ? |
| Text mining, also referred to as text data mining, roughly equivalent to text analytics, is the process of deriving high-quality information from text. High-quality information is typically derived through the devising of patterns and trends through means such as statistical pattern learning |
| What is Information Extraction ? |
| Information extraction is the task of automatically extracting structured information from unstructured and/or semi-structured machine-readable documents. In most of the cases this activity concerns processing human language texts by means of natural language processing |
| What is object standardization ? When it will be used ? |
| Text data often contains words or phrases which are not present in any standard lexical dictionaries. These pieces are not recognized by search engines and models. |
| What is text generation ? When we will do it ? |
| Generate new text from understanding old data. |
| What is text summarization ? When we will do it ? |
| Automatic summarization is the process of shortening a text document with software, in order to create a summary with the major points of the original document. Technologies that can make a coherent summary take into account variables such as length, writing style and syntax.  It’s widely used in news article sites. |
| What is Topic Modeling ? When we will do it ? |
| Topic modeling is a type of statistical modeling for discovering the abstract “topics” that occur in a collection of documents. Latent Dirichlet Allocation (LDA) is an example of topic model and is used to classify text in a document to a particulartopic |
| What is sentiment analysis ? When we will do it ? |
|  |
| What Term frequency(TF) ? |
|  |
| What is Inverse term frequency (IDF) ? |
|  |
| What is difference between NLTK and Spacy ? |
|  |
| What is difference between OpenNLP and NLTK ? |
|  |
| What is sequence modeling ? How it’s helpful in NLP ? |
|  |
| What is dependency parsing  ? |
|  |
| What is semantic parsing ? |
|  |
| What is constituency parsing ? |
|  |
| What is difference between shallow parsing and dependency parsing ? |
|  |
| How does the PageRank algorithm work? |
|  |
| What is Differentiate regular grammar and regular expression. |
|  |
| How will you estimate the entropy of the English language? |
|  |
| What is bagofwords model ? |
|  |
| What is cosine distance ? |
|  |
| What is doc2vec model ? |
|  |
| What is CBOW( continuous bag of words ) |
|  |
| What is Skip-gram ? |
|  |
| What are models to reduce dimensionality of data in nlp |
|  |
| Latent Dirichlet Allocation |
|  |
| Latent Semantic Indexing |
|  |
| Keyword Normalization |
|  |
| What is document-term matrix ? |
|  |
| A document-term matrix or term-document matrix is a mathematical matrix that describes the frequency of terms that occur in a collection of documents. |
|  |
| What is pragmatic analysis in NLP? |
|  |
| How can you find word similarity in nlp ? |
|  |
| How can you find sentence similarity in nlp ? |
|  |
| How can you find document similarity in nlp ? |
|  |
| What is NLP usage in recommendation engines ? |
|  |
| What are conditional random fields ? |
|  |
| What are hidden markov fields ? |
|  |
| What is Naive bayes algorithm, When we can use this algorithm in NLP ? |
|  |
| What is Text Matching / Similarity techniques ? |
|  |
| Levenshtein Distance |
| Phonetic Matching |
| Flexible String Matching |
| Cosine Similarity |
|  |
| What is Coreference Resolution ? |
|  |
| What is Ambiguity in NLP ? |