Contents

[1. What do you understand by Natural Language Processing? 11](#_Toc168315265)

[3. What are stop words? 12](#_Toc168315266)

[4. What is NLTK? 12](#_Toc168315267)

[5. What is Syntactic Analysis? 13](#_Toc168315268)

[6. What is Semantic Analysis? 14](#_Toc168315269)

[7. List the components of Natural Language Processing. 15](#_Toc168315270)

[8. What is Latent Semantic Indexing (LSI)? 15](#_Toc168315271)

[9. What are Regular Expressions? 16](#_Toc168315272)

[10. What is Regular Grammar? 16](#_Toc168315273)

[11. What is Parsing in the context of NLP? 17](#_Toc168315274)

[12. What is TF-IDF? 18](#_Toc168315275)

[13. Define the terminology in NLP. 19](#_Toc168315276)

[14. Explain Dependency Parsing in NLP. 20](#_Toc168315277)

[15. What is the difference between NLP and NLU? 21](#_Toc168315278)

[16. What is the difference between NLP and CI? 21](#_Toc168315279)

[17. What is Pragmatic Analysis? 22](#_Toc168315280)

[18. What is Pragmatic Ambiguity? 22](#_Toc168315281)

[19. What are unigrams, bigrams, trigrams, and n-grams in NLP? 23](#_Toc168315282)

[20. What are the steps involved in solving an NLP problem? 24](#_Toc168315283)

[21. What is Feature Extraction in NLP? 24](#_Toc168315284)

[22. What are precision and recall? 25](#_Toc168315285)

[23. What is F1 score in NLP? 25](#_Toc168315286)

[**Advanced NLP Interview Questions** 26](#_Toc168315287)

[24. How to tokenize a sentence using the nltk package? 26](#_Toc168315288)

[25. Explain how we can do parsing. 27](#_Toc168315289)

[26. Explain Stemming with the help of an example. 28](#_Toc168315290)

[27. Explain Lemmatization with the help of an example. 29](#_Toc168315291)

[28. What is Parts-of-speech Tagging? 29](#_Toc168315292)

[29. Explain Named Entity Recognition by implementing it. 30](#_Toc168315293)

[30. How to check word similarity using the spaCy package 31](#_Toc168315294)

[1. What is NLP? 32](#_Toc168315295)

[2. What are the main challenges in NLP? 32](#_Toc168315296)

[3. What are the different tasks in NLP? 33](#_Toc168315297)

[What do you mean by Corpus in NLP? 34](#_Toc168315298)

[5. What do you mean by text augmentation in NLP and what are the different text augmentation techniques in NLP? 34](#_Toc168315299)

[6. What are some common pre-processing techniques used in NLP? 35](#_Toc168315300)

[7. What is text normalization in NLP? 35](#_Toc168315301)

[**8. What is tokenization in NLP?** 36](#_Toc168315302)

[**9. What is NLTK and How it’s helpful in NLP?** 37](#_Toc168315303)

[**10. What is stemming in NLP, and how is it different from lemmatization?** 37](#_Toc168315304)

[**11. How does part-of-speech tagging work in NLP?** 38](#_Toc168315305)

[**12. What is named entity recognition in NLP?** 38](#_Toc168315306)

[**13. What is parsing in NLP?** 39](#_Toc168315307)

[**14. What are the different types of parsing in NLP?** 39](#_Toc168315308)

[**15. What do you mean by vector space in NLP?** 40](#_Toc168315309)

[**16. What is the bag-of-words model?** 41](#_Toc168315310)

[**17. Define the Bag of N-grams model in NLP.** 42](#_Toc168315311)

[**18. What is the term frequency-inverse document frequency (TF-IDF)?** 42](#_Toc168315312)

[**19. Explain the concept of cosine similarity and its importance in NLP.** 43](#_Toc168315313)

[**20. What are the differences between rule-based, statistical-based and neural-based approaches in NLP?** 44](#_Toc168315314)

[**21. What do you mean by Sequence in the Context of NLP?** 45](#_Toc168315315)

[**22. What are the various types of machine learning algorithms used in NLP?** 46](#_Toc168315316)

[**23. What is Sequence Labelling in NLP?** 47](#_Toc168315317)

[**24.What is topic modelling in NLP?** 48](#_Toc168315318)

[**25. What is the GPT?** 49](#_Toc168315319)

[**Advanced NLP Interview Questions for Experienced** 49](#_Toc168315320)

[**26. What are word embeddings in NLP?** 49](#_Toc168315321)

[**27. What are the various algorithms used for training word embeddings?** 50](#_Toc168315322)

[**28. How to handle out-of-vocabulary (OOV) words in NLP?** 51](#_Toc168315323)

[**29. What is the difference between a word-level and character-level language model?** 52](#_Toc168315324)

[**Word-level** 53](#_Toc168315325)

[**Character-level** 53](#_Toc168315326)

[**30. What is word sense disambiguation?** 54](#_Toc168315327)

[**31. What is co-reference resolution?** 54](#_Toc168315328)

[**32.What is information extraction?** 54](#_Toc168315329)

[**33. What is the Hidden Markov Model, and How it’s helpful in NLP tasks?** 55](#_Toc168315330)

[**34. What is the conditional random field (CRF) model in NLP?** 56](#_Toc168315331)

[**35. What is a recurrent neural network (RNN)?** 57](#_Toc168315332)

[**36. How does the Backpropagation through time work in RNN?** 58](#_Toc168315333)

[**37. What are the limitations of a standard RNN?** 59](#_Toc168315334)

[**38. What is a long short-term memory (LSTM) network?** 59](#_Toc168315335)

[**39. What is the GRU model in NLP?** 60](#_Toc168315336)

[**40. What is the sequence-to-sequence (Seq2Seq) model in NLP?** 63](#_Toc168315337)

[**41. How does the attention mechanism helpful in NLP?** 64](#_Toc168315338)

[**42. What is the Transformer model?** 64](#_Toc168315339)

[**43. What is the role of the self-attention mechanism in Transformers?** 65](#_Toc168315340)

[**44. What is the purpose of the multi-head attention mechanism in Transformers?** 65](#_Toc168315341)

[**45. What are positional encodings in Transformers, and why are they necessary?** 66](#_Toc168315342)

[**46. Describe the architecture of the Transformer model.** 66](#_Toc168315343)

[**47. What is the difference between a generative and discriminative model in NLP?** 68](#_Toc168315344)

[**Generative Models** 68](#_Toc168315345)

[**Discriminative Models** 68](#_Toc168315346)

[**48. What is machine translation, and how does it is performed?** 69](#_Toc168315347)

[**49. What is the BLEU score?** 69](#_Toc168315348)

[**50. List out the popular NLP task and their corresponding evaluation metrics.** 71](#_Toc168315349)

[**Natural Language Processing(NLP) Tasks** 71](#_Toc168315350)

[**Evaluation Metric** 71](#_Toc168315351)

[**1. What are the stages in the lifecycle of a natural language processing (NLP) project?** 74](#_Toc168315352)

[**2. What are some of the common NLP tasks?** 75](#_Toc168315353)

[**3. What are the different approaches used to solve NLP problems?** 76](#_Toc168315354)

[**4. How do Conversational Agents work?** 77](#_Toc168315355)

[**5. What is meant by data augmentation? What are some of the ways in which data augmentation can be done in NLP projects?** 77](#_Toc168315356)

[**6. How can data be obtained for NLP projects?** 78](#_Toc168315357)

[**7. What do you mean by Text Extraction and Cleanup?** 78](#_Toc168315358)

[**8. What are the steps involved in preprocessing data for NLP?** 79](#_Toc168315359)

[**9. What do you mean by Stemming in NLP?** 79](#_Toc168315360)

[**10. What do you mean by Lemmatization in NLP?** 80](#_Toc168315361)

[**NLP Interview Questions for Experienced** 80](#_Toc168315362)

[**1. What is the meaning of Text Normalization in NLP?** 80](#_Toc168315363)

[**2. Explain the concept of Feature Engineering.** 81](#_Toc168315364)

[**3. What is an ensemble method in NLP?** 81](#_Toc168315365)

[**4. What do you mean by TF-IDF in Natural language Processing?** 82](#_Toc168315366)

[**5. What are the steps to follow when building a text classification system?** 82](#_Toc168315367)

[**6. Explain how parsing is done in NLP.** 83](#_Toc168315368)

[**7. What do you mean by a Bag of Words (BOW)?** 84](#_Toc168315369)

[**8. What do you mean by Parts of Speech (POS) tagging in NLP?** 84](#_Toc168315370)

[**9. What is Latent Semantic Indexing (LSI) in NLP?** 84](#_Toc168315371)

[**10. What is the difference between NLP and NLU?** 84](#_Toc168315372)

[**11. What are some metrics on which NLP models are evaluated?** 85](#_Toc168315373)

[**12. Explain the pipeline for Information extraction (IE) in NLP.** 85](#_Toc168315374)

[**13. What do you mean by Autoencoders?** 87](#_Toc168315375)

[**14. What do you mean by Masked language modelling?** 87](#_Toc168315376)

[**15. What is the meaning of Pragmatic Analysis in NLP?** 88](#_Toc168315377)

[**16. What is the meaning of N-gram in NLP?** 88](#_Toc168315378)

[**17. What do you mean by perplexity in NLP?** 88](#_Toc168315379)

[**1. What is Naive Bayes algorithm, When we can use this algorithm in NLP?** 89](#_Toc168315380)

[**2. Explain Dependency Parsing in NLP?** 89](#_Toc168315381)

[**3. What is text Summarization?** 89](#_Toc168315382)

[**4. What is NLTK? How is it different from Spacy?** 90](#_Toc168315383)

[**5. What is information extraction?** 90](#_Toc168315384)

[**6. What is Bag of Words?** 90](#_Toc168315385)

[**7. What is Pragmatic Ambiguity in NLP?** 90](#_Toc168315386)

[**8. What is Masked Language Model?** 91](#_Toc168315387)

[**9. What is the difference between NLP and CI(Conversational Interface)?** 91](#_Toc168315388)

[**10. What are the best NLP Tools?** 91](#_Toc168315389)

[**11. What is POS tagging?** 92](#_Toc168315390)

[**12. What is NES?** 92](#_Toc168315391)

[**NLP Interview Questions for Experienced** 92](#_Toc168315392)

[**13. Which of the following techniques can be used for keyword normalization in NLP, the process of converting a keyword into its base form?** 92](#_Toc168315393)

[**14. Which of the following techniques can be used to compute the distance between two-word vectors in NLP?** 93](#_Toc168315394)

[**15. What are the possible features of a text corpus in NLP?** 94](#_Toc168315395)

[**16. 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?** 94](#_Toc168315396)

[**17. Which of the text parsing techniques can be used for noun phrase detection, verb phrase detection, subject detection, and object detection in NLP.** 95](#_Toc168315397)

[**18. Dissimilarity between words expressed using cosine similarity will have values significantly higher than 0.5** 95](#_Toc168315398)

[**19. Which one of the following is keyword Normalization techniques in NLP** 95](#_Toc168315399)

[**20. Which of the below are NLP use cases?** 96](#_Toc168315400)

[**21. In a corpus of N documents, one randomly chosen document contains a total of T terms and the term “hello” appears K times.** 96](#_Toc168315401)

[**22. 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** 96](#_Toc168315402)

[**23. In NLP, The process of removing words like “and”, “is”, “a”, “an”, “the” from a sentence is called as** 97](#_Toc168315403)

[**24. In NLP, The process of converting a sentence or paragraph into tokens is referred to as Stemming** 97](#_Toc168315404)

[**25. In NLP, Tokens are converted into numbers before giving to any Neural Network** 97](#_Toc168315405)

[**26. Identify the odd one out** 98](#_Toc168315406)

[**27. TF-IDF helps you to establish?** 98](#_Toc168315407)

[**28. In NLP, The process of identifying people, an organization from a given sentence, paragraph is called** 100](#_Toc168315408)

[**29. Which one of the following is not a pre-processing technique in NLP** 100](#_Toc168315409)

[**30. In text mining, converting text into tokens and then converting them into an integer or floating-point vectors can be done using** 101](#_Toc168315410)

[**31. In NLP, Words represented as vectors are called Neural Word Embeddings** 102](#_Toc168315411)

[**32. In NLP, Context modeling is supported with which one of the following word embeddings** 102](#_Toc168315412)

[**33. In NLP, Bidirectional context is supported by which of the following embedding** 102](#_Toc168315413)

[**34. Which one of the following Word embeddings can be custom trained for a specific subject in NLP** 103](#_Toc168315414)

[**35. Word embeddings capture multiple dimensions of data and are represented as vectors** 103](#_Toc168315415)

[**36. In NLP, Word embedding vectors help establish distance between two tokens** 103](#_Toc168315416)

[**37. 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** 103](#_Toc168315417)

[**38. Which of the following will be a better choice to address NLP use cases such as semantic similarity, reading comprehension, and common sense reasoning** 104](#_Toc168315418)

[**39. Transformer architecture was first introduced with?** 104](#_Toc168315419)

[**40. Which of the following architecture can be trained faster and needs less amount of training data** 104](#_Toc168315420)

[**41. Same word can have multiple word embeddings possible with \_\_\_\_\_\_\_\_\_\_\_\_?** 105](#_Toc168315421)

[**42. For a given token, its input representation is the sum of embedding from the token, segment and position** 106](#_Toc168315422)

[**43. Trains two independent LSTM language model left to right and right to left and shallowly concatenates them.** 107](#_Toc168315423)

[**44. Uses unidirectional language model for producing word embedding.** 107](#_Toc168315424)

[**45. In this architecture, the relationship between all words in a sentence is modelled irrespective of their position. Which architecture is this?** 107](#_Toc168315425)

[**46. List 10 use cases to be solved using NLP techniques?** 108](#_Toc168315426)

[**47. Transformer model pays attention to the most important word in Sentence.** 108](#_Toc168315427)

[**48. Which NLP model gives the best accuracy amongst the following?** 108](#_Toc168315428)

[**49. Permutation Language models is a feature of** 109](#_Toc168315429)

[**50. Transformer XL uses relative positional embedding** 109](#_Toc168315430)

[**Natural Language Processing FAQs** 110](#_Toc168315431)

[**1. Why do we need NLP?** 110](#_Toc168315432)

[**2. What must a natural language program decide?** 110](#_Toc168315433)

[**3. Where can NLP be useful?** 110](#_Toc168315434)

[**4. How to prepare for an NLP Interview?** 110](#_Toc168315435)

[**5. What are the main challenges of NLP?** 110](#_Toc168315436)

[**6. Which NLP model gives best accuracy?** 110](#_Toc168315437)

[**7. What are the major tasks of NLP?** 111](#_Toc168315438)

[**8. What are stop words in NLP?** 111](#_Toc168315439)

[**9. What is stemming in NLP?** 111](#_Toc168315440)

[**10. Why is NLP so hard?** 111](#_Toc168315441)

[**11. What does a NLP pipeline consist of \*?** 111](#_Toc168315442)

[Scenario-based NLP Interview Questions 112](#_Toc168315443)

[Q1. What is natural language processing (NLP), and how does it relate to artificial intelligence? 112](#_Toc168315444)

[Q2. Can you explain the difference between syntax and semantics in NLP? 112](#_Toc168315445)

[Q3. How would you approach developing a machine learning model for a named entity recognition (NER) task? 112](#_Toc168315446)

[Q4. What are some common challenges or limitations of NLP, and how do you overcome them? 113](#_Toc168315447)

[Q5. How would you approach designing a chatbot for a customer service use case? 113](#_Toc168315448)

[Q6. How would you use machine learning to extract meaningful insights from unstructured text data? 113](#_Toc168315449)

[Q7. Can you explain the difference between supervised and unsupervised learning in NLP? 114](#_Toc168315450)

[Q8. How would you handle a situation where a text classification model is misclassifying a significant number of documents? 114](#_Toc168315451)

[Q9. How would you go about developing a system for sentiment analysis of social media posts? 114](#_Toc168315452)

[Q10. How would you go about evaluating the performance of an NLP model? 115](#_Toc168315453)

[NLP Algorithm Interview Questions with Answers 118](#_Toc168315454)

[The Most Common Use-cases of NLP are: 122](#_Toc168315455)

[Let Us Look At The Most Common NLP Terms 122](#_Toc168315456)

[NLP Interview Questions With Answers 123](#_Toc168315457)

[1. What is the NLG (Natural Language Generation)? 126](#_Toc168315458)

[2. What is the order of steps in natural language understanding? 126](#_Toc168315459)

[3. What is signal processing in NLP? 127](#_Toc168315460)

[4. What is pragmatic analysis in NLP? 127](#_Toc168315461)

[5. What is syntactic analysis in NLP? 127](#_Toc168315462)

[6. What is semantic analysis in NLP? 127](#_Toc168315463)

[7. What is sentiment analysis in NLP? 127](#_Toc168315464)

[8. What is discourse analysis in NLP? 128](#_Toc168315465)

[9. What is pragmatic ambiguity in NLP? 128](#_Toc168315466)

[10. What are the major applications of NLP? 128](#_Toc168315467)

[11. List any real-world application of NLP? 129](#_Toc168315468)

[12. What are the common NLP techniques? 129](#_Toc168315469)

[What are the components of NLP? 130](#_Toc168315470)

[14. What are the tools used for training NLP models? 130](#_Toc168315471)

[15. Which NLP technique uses a lexical knowledge base to obtain the correct base form of the words? 130](#_Toc168315472)

[16. List the models to reduce the dimensionality of data in NLP. 130](#_Toc168315473)

[17. List some open-source libraries for NLP. 130](#_Toc168315474)

[18. Explain the masked language model. 131](#_Toc168315475)

[19. What is the bag of words model? 131](#_Toc168315476)

[20. What is CBOW in NLP? 132](#_Toc168315477)

[21. What is TF-IDF and what are its uses? 132](#_Toc168315478)

[22. What are POS and tagging? 132](#_Toc168315479)

[23. What is n-gram in NLP? 133](#_Toc168315480)

[24. What is skip-gram? 133](#_Toc168315481)

[25. What is the corpus in NLP? 133](#_Toc168315482)

[26. What are the features of the text corpus in NLP? 133](#_Toc168315483)

[27. What is normalization in NLP? 133](#_Toc168315484)

[28. What is keyword normalization? 133](#_Toc168315485)

[29. What is lemmatization in NLP? 134](#_Toc168315486)

[30. What is stemming in NLP? 134](#_Toc168315487)

[31. What is ambiguity in NLP? 134](#_Toc168315488)

[32. What is tokenization in NLP? 134](#_Toc168315489)

[33. What are stop words in NLP? 134](#_Toc168315490)

[34. How to find word similarity in NLP? 134](#_Toc168315491)

[35. How to find sentence similarity in NLP? 135](#_Toc168315492)

[36. How to find document similarity in NLP? 135](#_Toc168315493)

[37. What are transformers? 135](#_Toc168315494)

[38. What are punctuations in NLP, and how can we remove them? 135](#_Toc168315495)

[39. What is latent semantic indexing (LSI)? 135](#_Toc168315496)

[40. What is named entity recognition (NER)? 136](#_Toc168315497)

[41. What is NLTK in NLP? 136](#_Toc168315498)

[42. What is spaCY? 136](#_Toc168315499)

[43. What is openNLP? 137](#_Toc168315500)

[44. What is the difference between NLTK and openNLP? 137](#_Toc168315501)

[45. What is parsing? 137](#_Toc168315502)

[46. What is dependency parsing? 137](#_Toc168315503)

[47. What is semantic parsing? 137](#_Toc168315504)

[48. What is constituency parsing? 137](#_Toc168315505)

[49. What is shallow parsing? 138](#_Toc168315506)

[50. What are the differences between dependency parsing and shallow parsing? 138](#_Toc168315507)

[51. What is language modeling? 138](#_Toc168315508)

[52. What is topic modeling? 138](#_Toc168315509)

[53. What is text summarization in NLP? 138](#_Toc168315510)

[54. What is the difference between a regular expression and regular grammar? 139](#_Toc168315511)

[55. What is perplexity in NLP? 139](#_Toc168315512)

[56. What is the Naive Bayes algorithm, and where is it used in NLP? 139](#_Toc168315513)

[57. What is the PageRank algorithm? 139](#_Toc168315514)

[58. What is noise removal? 139](#_Toc168315515)

[59. What is word embedding? 139](#_Toc168315516)

[60. What are the word embedding libraries? 140](#_Toc168315517)

[61. What is word2vec? 140](#_Toc168315518)

[61. What is doc2vec? 140](#_Toc168315519)

[63. What is a document-term matrix? 141](#_Toc168315520)

[64. What is wordnet? 141](#_Toc168315521)

[65. What is GloVe in NLP? 141](#_Toc168315522)

[66. What is a flexible string matching? 141](#_Toc168315523)

[67. What is cosine similarity? 141](#_Toc168315524)

[68. What is information extraction? 141](#_Toc168315525)

[69. What is object standardization, and when is it used? 142](#_Toc168315526)

[71. What is text generation, and when is it done? 142](#_Toc168315527)

[71. How can we estimate the entropy of the English language? 142](#_Toc168315528)

[72. What is Latent Dirichlet Allocation? 142](#_Toc168315529)

[73. What are the conditional random fields? 142](#_Toc168315530)

[74. What are the hidden Markov random fields? 142](#_Toc168315531)

[75. What is a coreference resolution? 143](#_Toc168315532)

[76. What is PAC learning? 143](#_Toc168315533)

[77. What is sequence learning? 143](#_Toc168315534)

[78. What is an ensemble method? 143](#_Toc168315535)

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

[Natural Language Processing](https://intellipaat.com/blog/what-is-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. Techniques in NLP allow computer systems to process and interpret data in the form of natural languages.

## 3. What are stop words?

Stop words are said to be useless data for a search engine. Words such as articles, prepositions, etc. are considered 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.

## 4. 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:

1. SequentialBackoffTagger
2. DefaultTagger
3. UnigramTagger
4. treebank
5. wordnet
6. FreqDist
7. patterns
8. RegexpTagger
9. backoff\_tagger
10. UnigramTagger, BigramTagger, and TrigramTagger

## 5. What is Syntactic Analysis?

Syntactic analysis is a technique of analyzing sentences to extract meaning from them. 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:

A diagram of a diagram

Description automatically generated

1. **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.
2. **Word segmentation**: The segmentation of words segregates the text into small significant units.
3. **Morphological segmentation**: The purpose of morphological segmentation is to break words into their base form.
4. **Stemming**: It is the process of removing the suffix from a word to obtain its root word.
5. **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:

A diagram of a sematic analysis

Description automatically generated

1. **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.
2. **Word sense disambiguation:**It helps identify the sense of a word used in different sentences.
3. **Natural language generation:**It is a process used by the software to convert structured data into human-spoken languages. By using NLG, organizations can automate content for custom reports.

## 7. List the components of Natural Language Processing.

The major components of NLP are as follows:

A blue and purple rectangles with white text

Description automatically generated

* **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.

## 8. 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 is best suited 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 which helps improve the analysis and understanding of a text or a document.

## 9. 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.

## 10. What is Regular Grammar?

Regular grammar is used to represent a regular language.

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:

1. ‘N’ is used to represent the non-terminal set.
2. ‘∑’ represents the set of terminals.
3. ‘P’ stands for the set of productions.
4. ‘S € N’ denotes the start of non-terminal.

Regular grammar is of 2 types:  
(a) Left Linear Grammar(LLG)

(b) Right Linear Grammar(RLG)

## 11. What is Parsing in the context of NLP?

[Parsing in NLP](https://intellipaat.com/blog/what-is-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:

A diagram of words and a diagram of words

Description automatically generated

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

## 12. 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.

## 13. 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:

A diagram of a diagram

Description automatically generated

**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

## 14. 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:

A diagram of a graph

Description automatically generated

## 15. What is the difference between NLP and NLU?

The below table shows the difference between NLP and NLU:A screenshot of a computer screen

Description automatically generated

## 16. What is the difference between NLP and CI?

The below table shows the difference between NLP and CI:

A comparison of a conversation between a conversational language

Description automatically generated

## 17. 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.

## 18. 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.

## 19. 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:

A group of purple rectangular boxes with white text

Description automatically generated

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.

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

Below are the steps involved in solving an NLP problem:

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

## 21. 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.

## 22. What are 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.

A black text on a white background

Description automatically generated

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

A black text on a white background

Description automatically generated

## 23. What is F1 score in NLP?

[**F1 score**](https://intellipaat.com/blog/what-is-f1-score-in-machine-learning/) 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:

A black text with black text

Description automatically generated

## **Advanced NLP Interview Questions**

## 24. 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**:

|  |  |
| --- | --- |
| 1 | **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.”

|  |  |
| --- | --- |
| 1 | sent\_tokenize(Para) |

**Output**:

|  |  |
| --- | --- |
| 1  2  3 | [ '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.

|  |  |
| --- | --- |
| 1 | **from** nltk.tokenize **import** word\_tokenize |

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

|  |  |
| --- | --- |
| 1 | word\_tokenize(Para) |

**Output**:

|  |  |
| --- | --- |
| 1 | [ 'Hi' , 'Guys' , ' . ' , 'Welcome' , 'to' , 'Intellipaat' , ' . ' , 'This' , 'is' ,   'a', 'blog' , 'on' , 'the' , 'NLP' , 'interview' , 'questions' , 'and' , 'answers' , ' . ' ] |

## 25. 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.

A diagram of a question

Description automatically generated with medium confidence

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

|  |  |
| --- | --- |
| 1  2 | **import** nltk  text **=** ”Top 30 NLP interview questions **and** answers” |

We will now tokenize the text using **word\_tokenize**.

|  |  |
| --- | --- |
| 1 | text\_token**=** word\_tokenize(text) |

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

|  |  |
| --- | --- |
| 1 | list(nltk.unigrams(text)) |

**Output**:

|  |  |
| --- | --- |
| 1 | [ "Top 30 NLP interview questions and answer"] |
| 1 | list(nltk.bigrams(text)) |

**Output**:

|  |  |
| --- | --- |
| 1 | ["Top 30", "30 NLP", "NLP interview", "interview questions",   "questions and", "and answer"] |
| 1 | list(nltk.trigrams(text)) |

**Output**:

|  |  |
| --- | --- |
| 1 | ["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.

|  |  |
| --- | --- |
| 1 | list(nltk.ngrams(text,n)) |

## 26. 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.

|  |  |
| --- | --- |
| 1 | **from** nltk.stem **import** PorterStemmer |

Creating an object for PorterStemmer

|  |  |
| --- | --- |
| 1  2 | pst**=**PorterStemmer()  pst.stem(“running”), pst.stem(“cookies”), pst.stem(“flying”) |

**Output**:

|  |  |
| --- | --- |
| 1 | (‘run’, ‘cooki', ‘fly’ ) |

## 27. 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.

|  |  |
| --- | --- |
| 1  2 | **from** nltk.stem **import** wordnet  **from** nltk.stem **import** WordnetLemmatizer |

Creating an object for WordnetLemmatizer()

|  |  |
| --- | --- |
| 1  2  3  4 | lemma**=** WordnetLemmatizer()  list **=** [“Dogs”, “Corpora”, “Studies”]  **for** n **in** list:  print(n **+** “:” **+** lemma.lemmatize(n)) |

**Output**:

|  |  |
| --- | --- |
| 1  2  3 | Dogs: Dog  Corpora: Corpus  Studies: Study |

## 28. 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.

|  |  |
| --- | --- |
| 1  2  3  4  5 | **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

|  |  |
| --- | --- |
| 1 | tokenized\_text **=** sent\_tokenize(txt) |

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

|  |  |
| --- | --- |
| 1  2  3 | **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.

|  |  |
| --- | --- |
| 1  2 | tagged\_words **=** nltk.pos\_tag(wordsList)  print(tagged\_words) |

**Output**:

|  |  |
| --- | --- |
| 1 | [('Sourav', 'NNP'), ('Pratyush', 'NNP'), ('Abhinav', 'NNP'), ('good',  'JJ'), ('friends', 'NNS')] |

## 29. 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:

|  |  |
| --- | --- |
| 1  2  3  4  5 | **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**:

|  |  |
| --- | --- |
| 1  2  3 | Office 9 15 Place  Google 19 25 ORG  California 32 41 GPE |

Note: Office 9 15 Place means word starts at 9th position when tokenized and ends at 15, this is inclusive of spaces.

## 30. 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.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | **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**:

|  |  |
| --- | --- |
| 1  2  3 | hot  True 5.6898586 False  cold True6.5396233 False  Similarity: 0.597265 |

## 1. What is NLP?

NLP stands for **Natural Language Processing**. The subfield of [Artificial intelligence](https://www.geeksforgeeks.org/ai-algorithms/) and computational linguistics deals with the interaction between computers and human languages. It involves developing algorithms, models, and techniques to enable machines to understand, interpret, and generate natural languages in the same way as a human does.

NLP encompasses a wide range of tasks, including language translation, sentiment analysis, text categorization, information extraction, speech recognition, and natural language understanding. NLP allows computers to extract meaning, develop insights, and communicate with humans in a more natural and intelligent manner by processing and analyzing textual input.

## 2. What are the main challenges in NLP?

The complexity and variety of human language create numerous difficult problems for the study of Natural Language Processing (NLP). The primary challenges in NLP are as follows:

* **Semantics and Meaning:** It is a difficult undertaking to accurately capture the meaning of words, phrases, and sentences. The semantics of the language, including word sense disambiguation, metaphorical language, idioms, and other linguistic phenomena, must be accurately represented and understood by NLP models.
* **Ambiguity**: Language is ambiguous by nature, with words and phrases sometimes having several meanings depending on context. Accurately resolving this ambiguity is a major difficulty for NLP systems.
* **Contextual Understanding:**Context is frequently used to interpret language. For NLP models to accurately interpret and produce meaningful replies, the context must be understood and used. Contextual difficulties include, for instance, comprehending referential statements and resolving pronouns to their antecedents.
* **Language Diversity:** NLP must deal with the world’s wide variety of languages and dialects, each with its own distinctive linguistic traits, lexicon, and grammar. The lack of resources and knowledge of low-resource languages complicates matters.
* **Data Limitations and Bias:**The availability of high-quality labelled data for training NLP models can be limited, especially for specific areas or languages. Furthermore, biases in training data might impair model performance and fairness, necessitating careful consideration and mitigation.
* **Real-world Understanding:**NLP models often fail to understand real-world knowledge and common sense, which humans are born with. Capturing and implementing this knowledge into NLP systems is a continuous problem.

## 3. What are the different tasks in NLP?

Natural Language Processing (NLP) includes a wide range of tasks involving understanding, processing, and creation of human language. Some of the most important tasks in NLP are as follows:

* [Text Classification](https://www.geeksforgeeks.org/classification-of-text-documents-using-the-approach-of-naive-bayes/)
* [Named Entity Recognition (NER)](https://www.geeksforgeeks.org/python-named-entity-recognition-ner-using-spacy/)
* [Part-of-Speech Tagging (POS)](https://www.geeksforgeeks.org/nlp-part-of-speech-default-tagging/)
* [Sentiment Analysis](https://www.geeksforgeeks.org/what-is-sentiment-analysis/)
* [Language Modeling](https://www.geeksforgeeks.org/videos/what-is-language-modelling-in-nlp/)
* [Machine Translation](https://www.geeksforgeeks.org/machine-translation-of-languages-in-artificial-intelligence/)
* [Chatbots](https://www.geeksforgeeks.org/battle-of-ai-chatbots-which-chatbot-will-rule-the-present-and-future/)
* [Text Summarization](https://www.geeksforgeeks.org/python-extractive-text-summarization-using-gensim/)
* [Information Extraction](https://www.geeksforgeeks.org/difference-between-information-retrieval-and-information-extraction/)
* [Text Generation](https://www.geeksforgeeks.org/text-generation-using-recurrent-long-short-term-memory-network/)
* [Speech Recognition](https://www.geeksforgeeks.org/text-generation-using-recurrent-long-short-term-memory-network/)

## What do you mean by Corpus in NLP?

In NLP, a [corpus](https://www.geeksforgeeks.org/nlp-wordlist-corpus/) is a huge collection of texts or documents. It is a structured dataset that acts as a sample of a specific language, domain, or issue. A corpus can include a variety of texts, including books, essays, web pages, and social media posts. Corpora are frequently developed and curated for specific research or NLP objectives. They serve as a foundation for developing language models, undertaking linguistic analysis, and gaining insights into language usage and patterns.

## 5. What do you mean by text augmentation in NLP and what are the different text augmentation techniques in NLP?

[Text augmentation](https://www.geeksforgeeks.org/text-augmentation-techniques-in-nlp/) in NLP refers to the process that generates new or modified textual data from existing data to increase the diversity and quantity of training samples. Text augmentation techniques apply numerous alterations to the original text while keeping the underlying meaning.

Different text augmentation techniques in NLP include:

1. **Synonym Replacement:** Replacing words in the text with their synonyms to introduce variation while maintaining semantic similarity.
2. **Random Insertion/Deletion:**Randomly inserting or deleting words in the text to simulate noisy or incomplete data and enhance model robustness.
3. **Word Swapping:**Exchanging the positions of words within a sentence to generate alternative sentence structures.
4. **Back translation:** Translating the text into another language and then translating it back to the original language to introduce diverse phrasing and sentence constructions.
5. **Random Masking:** Masking or replacing random words in the text with a special token, akin to the approach used in masked language models like BERT.
6. **Character-level Augmentation:** Modifying individual characters in the text, such as adding noise, misspellings, or character substitutions, to simulate real-world variations.
7. **Text Paraphrasing:** Rewriting sentences or phrases using different words and sentence structures while preserving the original meaning.
8. **Rule-based Generation:** Applying linguistic rules to generate new data instances, such as using grammatical templates or syntactic transformations.

## 6. What are some common pre-processing techniques used in NLP?

[Natural Language Processing (NLP)](https://www.geeksforgeeks.org/natural-language-processing-nlp-pipeline/)preprocessing refers to the set of processes and techniques used to prepare raw text input for analysis, modelling, or any other NLP tasks. The purpose of preprocessing is to clean and change text data so that it may be processed or analyzed later.

Preprocessing in NLP typically involves a series of steps, which may include:

* [Tokenization](https://www.geeksforgeeks.org/tokenize-text-using-nltk-python/)
* [Stop Word Removal](https://www.geeksforgeeks.org/removing-stop-words-nltk-python/)
* [Text Normalization](https://www.geeksforgeeks.org/normalizing-textual-data-with-python/)
  + Lowercasing
  + Lemmatization
  + Stemming
  + Date and Time Normalization
* [Removal of Special Characters and Punctuation](https://www.geeksforgeeks.org/removing-punctuations-given-string/)
* [Removing HTML Tags or Markup](https://www.geeksforgeeks.org/program-to-remove-html-tags-from-a-given-string/)
* [Spell Correction](https://www.geeksforgeeks.org/correcting-words-using-nltk-in-python/)
* [Sentence Segmentation](https://www.geeksforgeeks.org/python-perform-sentence-segmentation-using-spacy/)

## 7. What is text normalization in NLP?

Text normalization, also known as text standardization, is the process of transforming text data into a standardized or normalized form It involves applying a variety of techniques to ensure consistency,  reduce variations, and simplify the representation of textual information.

The goal of text normalization is to make text more uniform and easier to process in Natural Language Processing (NLP) tasks. Some common techniques used in text normalization include:

* **Lowercasing**: Converting all text to lowercase to treat words with the same characters as identical and avoid duplication.
* **Lemmatization**: Converting words to their base or dictionary form, known as lemmas. For example, converting “running” to “run” or “better” to “good.”
* **Stemming**: Reducing words to their root form by removing suffixes or prefixes. For example, converting “playing” to “play” or “cats” to “cat.”
* **Abbreviation Expansion**: Expanding abbreviations or acronyms to their full forms. For example, converting “NLP” to “Natural Language Processing.”
* **Numerical Normalization**: Converting numerical digits to their written form or normalizing numerical representations. For example, converting “100” to “one hundred” or normalizing dates.
* **Date and Time Normalization**: Standardizing date and time formats to a consistent representation.

## **8. What is tokenization in NLP?**

[Tokenization](https://www.geeksforgeeks.org/tokenization-using-spacy-library/) is the process of breaking down text or string into smaller units called tokens. These tokens can be words, characters, or subwords depending on the specific applications. It is the fundamental step in many natural language processing tasks such as sentiment analysis, machine translation, and text generation. etc.

Some of the most common ways of tokenization are as follows:

* **Sentence tokenization:** In Sentence tokenizations, the text is broken down into individual sentences. This is one of the fundamental steps of tokenization.
* **Word tokenization:** In word tokenization, the text is simply broken down into words. This is one of the most common types of tokenization. It is typically done by splitting the text into spaces or punctuation marks.
* **Subword tokenization:** In subword tokenization, the text is broken down into subwords, which are the smaller part of words. Sometimes words are formed with more than one word, for example, Subword i.e Sub+ word, Here sub, and words have different meanings. When these two words are joined together, they form the new word “subword”, which means “a smaller unit of a word”. This is often done for tasks that require an understanding of the morphology of the text, such as stemming or lemmatization.
* **Char-label tokenization:** In Char-label tokenization, the text is broken down into individual characters. This is often used for tasks that require a more granular understanding of the text such as text generation, machine translations, etc.

## **9. What is NLTK and How it’s helpful in NLP?**

[NLTK](https://www.geeksforgeeks.org/python-nltk-tokenize-regexp/) stands for Natural Language Processing Toolkit. It is a suite of libraries and programs written in Python Language for symbolic and statistical natural language processing. It offers tokenization, stemming, lemmatization, POS tagging, Named Entity Recognization, parsing, semantic reasoning, and classification.

NLTK is a popular NLP library for Python. It is easy to use and has a wide range of features. It is also open-source, which means that it is free to use and modify.

## **10. What is stemming in NLP, and how is it different from lemmatization?**

Stemming and lemmatization are two commonly used word normalization techniques in NLP, which aim to reduce the words to their base or root word. Both have similar goals but have different approaches.

In [stemming](https://www.geeksforgeeks.org/python-stemming-words-with-nltk/), the word suffixes are removed using the heuristic or pattern-based rules regardless of the context of the parts of speech. The resulting stems may not always be actual dictionary words. Stemming algorithms are generally simpler and faster compared to lemmatization, making them suitable for certain applications with time or resource constraints.

In [lemmatization](https://www.geeksforgeeks.org/python-lemmatization-with-nltk/), The root form of the word known as lemma, is determined by considering the word’s context and parts of speech. It uses linguistic knowledge and databases (e.g., wordnet) to transform words into their root form. In this case, the output lemma is a valid word as per the dictionary. For example, lemmatizing “running” and “runner” would result in “run.” Lemmatization provides better interpretability and can be more accurate for tasks that require meaningful word representations.

## **11. How does part-of-speech tagging work in NLP?**

[Part-of-speech tagging](https://www.geeksforgeeks.org/part-speech-tagging-stop-words-using-nltk-python/) is the process of assigning a part-of-speech tag to each word in a sentence. The POS tags represent the syntactic information about the words and their roles within the sentence.

There are three main approaches for POS tagging:

* **Rule-based POS tagging:** It uses a set of handcrafted rules to determine the part of speech based on morphological, syntactic, and contextual patterns for each word in a sentence. For example, words ending with ‘-ing’ are likely to be a verb.
* **Statistical POS tagging:**The statistical model like Hidden Markov Model (HMMs) or Conditional Random Fields (CRFs) are trained on a large corpus of already tagged text. The model learns the probability of word sequences with their corresponding POS tags, and it can be further used for assigning each word to a most likely POS tag based on the context in which the word appears.
* **Neural network POS tagging:**The neural network-based model like RNN, LSTM, Bi-directional RNN, and transformer have given promising results in POS tagging by learning the patterns and representations of words and their context.

## **12. What is named entity recognition in NLP?**

[Named Entity Recognization (NER)](https://www.geeksforgeeks.org/named-entity-recognition/) is a task in natural language processing that is used to identify and classify the named entity in text. Named entity refers to real-world objects or concepts, such as persons, organizations, locations, dates, etc. NER is one of the challenging tasks in NLP because there are many different types of named entities, and they can be referred to in many different ways. The goal of NER is to extract and classify these named entities in order to offer structured data about the entities referenced in a given text.

The approach followed for Named Entity Recognization (NER) is the same as the POS tagging. The data used while training in NER is tagged with persons, organizations, locations, and dates.

## **13. What is parsing in NLP?**

In NLP, [parsing](https://www.geeksforgeeks.org/difference-between-top-down-parsing-and-bottom-up-parsing/) is defined as the process of determining the underlying structure of a sentence by breaking it down into constituent parts and determining the syntactic relationships between them according to formal grammar rules. The purpose of parsing is to understand the syntactic structure of a sentence, which allows for deeper learning of its meaning and encourages different downstream NLP tasks such as semantic analysis, information extraction, question answering, and machine translation. it is also known as syntax analysis or syntactic parsing.

The formal grammar rules used in parsing are typically based on Chomsky’s hierarchy. The simplest grammar in the Chomsky hierarchy is regular grammar, which can be used to describe the syntax of simple sentences. More complex grammar, such as context-free grammar and context-sensitive grammar, can be used to describe the syntax of more complex sentences.

## **14. What are the different types of parsing in NLP?**

In natural language processing (NLP), there are several types of parsing algorithms used to analyze the grammatical structure of sentences. Here are some of the main types of parsing algorithms:

* [**Constituency Parsing**](https://www.geeksforgeeks.org/constituency-parsing-and-dependency-parsing/): Constituency parsing in NLP tries to figure out a sentence’s hierarchical structure by breaking it into constituents based on a particular grammar. It generates valid constituent structures using context-free grammar. The parse tree that results represents the structure of the sentence, with the root node representing the complete sentence and internal nodes representing phrases. Constituency parsing techniques like as CKY, Earley, and chart parsing are often used for parsing. This approach is appropriate for tasks that need a thorough comprehension of sentence structure, such as semantic analysis and machine translation. When a complete understanding of sentence structure is required, constituency parsing, a classic parsing approach, is applied.
* [**Dependency Parsing**](https://www.geeksforgeeks.org/constituency-parsing-and-dependency-parsing/)**:** In NLP, dependency parsing identifies grammatical relationships between words in a sentence. It represents the sentence as a directed graph, with dependencies shown as labelled arcs. The graph emphasises subject-verb, noun-modifier, and object-preposition relationships. The head of a dependence governs the syntactic properties of another word. Dependency parsing, as opposed to constituency parsing, is helpful for languages with flexible word order. It allows for the explicit illustration of word-to-word relationships, resulting in a clear representation of grammatical structure.
* [**Top-down parsing:**](https://www.geeksforgeeks.org/difference-between-top-down-parsing-and-bottom-up-parsing/) Top-down parsing starts at the root of the parse tree and iteratively breaks down the sentence into smaller and smaller parts until it reaches the leaves. This is a more natural technique for parsing sentences. However, because it requires a more complicated language, it may be more difficult to implement.
* [**Bottom-up parsing:**](https://www.geeksforgeeks.org/difference-between-top-down-parsing-and-bottom-up-parsing/) Bottom-up parsing starts with the leaves of the parse tree and recursively builds up the tree from smaller and smaller constituents until it reaches the root. Although this method of parsing requires simpler grammar, it is frequently simpler to implement, even when it is less understandable.

## **15. What do you mean by vector space in NLP?**

In natural language processing (NLP), A [vector space](https://www.geeksforgeeks.org/web-information-retrieval-vector-space-model/) is a mathematical vector where words or documents are represented by numerical vectors form. The word or document’s specific features or attributes are represented by one of the dimensions of the vector. Vector space models are used to convert text into numerical representations that machine learning algorithms can understand.

Vector spaces are generated using techniques such as word embeddings, bag-of-words, and term frequency-inverse document frequency (TF-IDF). These methods allow for the conversion of textual data into dense or sparse vectors in a high-dimensional space. Each dimension of the vector may indicate a different feature, such as the presence or absence of a word, word frequency, semantic meaning, or contextual information.

## **16. What is the bag-of-words model?**

[Bag of Words](https://www.geeksforgeeks.org/bag-of-words-bow-model-in-nlp/) is a classical text representation technique in NLP that describes the occurrence of words within a document or not. It just keeps track of word counts and ignores the grammatical details and the word order.

Each document is transformed as a numerical vector, where each dimension corresponds to a unique word in the vocabulary. The value in each dimension of the vector represents the frequency, occurrence, or other measure of importance of that word in the document.

Let's consider two simple text documents:

Document 1: "I love apples."

Document 2: "I love mangoes too."

Step 1: Tokenization

Document 1 tokens: ["I", "love", "apples"]

Document 2 tokens: ["I", "love", "mangoes", "too"]

Step 2: Vocabulary Creation by collecting all unique words across the documents

Vocabulary: ["I", "love", "apples", "mangoes", "too"]

The vocabulary has five unique words, so each document vector will have five dimensions.

Step 3: Vectorization

Create numerical vectors for each document based on the vocabulary.

For Document 1:

- The dimension corresponding to "I" has a value of 1.

- The dimension corresponding to "love" has a value of 1.

- The dimension corresponding to "apples" has a value of 1.

- The dimensions corresponding to "mangoes" and "too" have values of 0 since they do not appear in Document 1.

Document 1 vector: [1, 1, 1, 0, 0]

For Document 2:

- The dimension corresponding to "I" has a value of 1.

- The dimension corresponding to "love" has a value of 1.

- The dimension corresponding to "mangoes" has a value of 1.

- The dimension corresponding to "apples" has a value of 0 since it does not appear in Document 2.

- The dimension corresponding to "too" has a value of 1.

Document 2 vector: [1, 1, 0, 1, 1]

The value in each dimension represents the occurrence or frequency of the corresponding word in the document. The BoW representation allows us to compare and analyze the documents based on their word frequencies.

## **17. Define the Bag of N-grams model in NLP.**

The [Bag of n-grams](https://www.geeksforgeeks.org/n-gram-language-modelling-with-nltk/) model is a modification of the standard bag-of-words (BoW) model in NLP. Instead of taking individual words to be the fundamental units of representation, the Bag of n-grams model considers contiguous sequences of n words, known as n-grams, to be the fundamental units of representation.

The Bag of n-grams model divides the text into n-grams, which can represent consecutive words or characters depending on the value of n. These n-grams are subsequently considered as features or tokens, similar to individual words in the BoW model.

The steps for creating a bag-of-n-grams model are as follows:

* The text is split or tokenized into individual words or characters.
* The tokenized text is used to construct N-grams of size n (sequences of n consecutive words or characters). If n is set to 1 known as uni-gram i.e. same as a bag of words, 2 i.e. bi-grams, and 3 i.e. tri-gram.
* A vocabulary is built by collecting all unique n-grams across the entire corpus.
* Similarly to the BoW approach, each document is represented as a numerical vector. The vector’s dimensions correspond to the vocabulary’s unique n-grams, and the value in each dimension denotes the frequency or occurrence of that n-gram in the document.

## **18. What is the term frequency-inverse document frequency (TF-IDF)?**

[Term frequency-inverse document frequency (TF-IDF)](https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/)is a classical text representation technique in NLP that uses a statistical measure to evaluate the importance of a word in a document relative to a corpus of documents. It is a combination of two terms: term frequency (TF) and inverse document frequency (IDF).

* **Term Frequency (TF):** Term frequency measures how frequently a word appears in a document. it is the ratio of the number of occurrences of a term or word (t ) in a given document (d) to the total number of terms in a given document (d). A higher term frequency indicates that a word is more important within a specific document.
* **Inverse Document Frequency (IDF):** Inverse document frequency measures the rarity or uniqueness of a term across the entire corpus. It is calculated by taking the logarithm of the ratio of the total number of documents in the corpus to the number of documents containing the term. it down the weight of the terms, which frequently occur in the corpus, and up the weight of rare terms.

The TF-IDF score is calculated by multiplying the term frequency (TF) and inverse document frequency (IDF) values for each term in a document. The resulting score indicates the term’s importance in the document and corpus. Terms that appear frequently in a document but are uncommon in the corpus will have high TF-IDF scores, suggesting their importance in that specific document.

## **19. Explain the concept of cosine similarity and its importance in NLP.**

The similarity between two vectors in a multi-dimensional space is measured using the cosine similarity metric. To determine how similar or unlike the vectors are to one another, it calculates the cosine of the angle between them.

In natural language processing (NLP), [Cosine similarity](https://www.geeksforgeeks.org/cosine-similarity/) is used to compare two vectors that represent text. The degree of similarity is calculated using the cosine of the angle between the document vectors. To compute the cosine similarity between two text document vectors, we often used the following procedures:

* Text Representation: Convert text documents into numerical vectors using approaches like bag-of-words, TF-IDF (Term Frequency-Inverse Document Frequency), or word embeddings like Word2Vec or GloVe.
* Vector Normalization: Normalize the document vectors to unit length. This normalization step ensures that the length or magnitude of the vectors does not affect the cosine similarity calculation.
* Cosine Similarity Calculation: Take the dot product of the normalised vectors and divide it by the product of the magnitudes of the vectors to obtain the cosine similarity.

Mathematically, the cosine similarity between two document vectors, 𝑎⃗   *a* and 𝑏⃗   *b* , can be expressed as:

Cosine Similarity(𝑎⃗,𝑏⃗)=𝑎⃗⋅𝑏⃗∣𝑎⃗∣∣𝑏⃗∣Cosine Similarity(*a*,*b*)=∣*a*∣∣*b*∣*a*⋅*b*​

Here,

* 𝑎⃗⋅𝑏⃗   *a*⋅*b* is the dot product of vectors a and b
* |a| and |b| represent the Euclidean norms (magnitudes) of vectors a and b, respectively.

The resulting cosine similarity score ranges from -1 to 1, where 1 represents the highest similarity, 0 represents no similarity, and -1 represents the maximum dissimilarity between the documents.

## **20. What are the differences between rule-based, statistical-based and neural-based approaches in NLP?**

[Natural language processing (NLP)](https://www.geeksforgeeks.org/natural-language-processing-nlp-pipeline/) uses three distinct approaches to tackle language understanding and processing tasks: rule-based, statistical-based, and neural-based.

1. **Rule-based Approach:**Rule-based systems rely on predefined sets of linguistic rules and patterns to analyze and process language.
   * Linguistic Rules are manually crafted rules by human experts to define patterns or grammar structures.
   * The knowledge in rule-based systems is explicitly encoded in the rules, which may cover syntactic, semantic, or domain-specific information.
   * Rule-based systems offer high interpretability as the rules are explicitly defined and understandable by human experts.
   * These systems often require manual intervention and rule modifications to handle new language variations or domains.
2. **Statistical-based Approach:**Statistical-based systems utilize statistical algorithms and models to learn patterns and structures from large datasets.
   * By examining the data’s statistical patterns and relationships, these systems learn from training data.
   * Statistical models are more versatile than rule-based systems because they can train on relevant data from various topics and languages.
3. **Neural-based Approach:** Neural-based systems employ deep learning models, such as neural networks, to learn representations and patterns directly from raw text data.
   * Neural networks learn hierarchical representations of the input text, which enable them to capture complex language features and semantics.
   * Without explicit rule-making or feature engineering, these systems learn directly from data.
   * By training on huge and diverse datasets, neural networks are very versatile and can perform a wide range of NLP tasks.
   * In many NLP tasks, neural-based models have attained state-of-the-art performance, outperforming classic rule-based or statistical-based techniques.

## **21. What do you mean by Sequence in the Context of NLP?**

A Sequence primarily refers to the sequence of elements that are analyzed or processed together. In [NLP](https://www.geeksforgeeks.org/natural-language-processing-nlp-pipeline/), a sequence may be a sequence of characters, a sequence of words or a sequence of sentences.

In general, sentences are often treated as sequences of words or tokens. Each word in the sentence is considered an element in the sequence. This sequential representation allows for the analysis and processing of sentences in a structured manner, where the order of words matters.

By considering sentences as sequences, NLP models can capture the contextual information and dependencies between words, enabling tasks such as part-of-speech tagging, named entity recognition, sentiment analysis, machine translation, and more.

## **22. What are the various types of machine learning algorithms used in NLP?**

There are various types of machine learning algorithms that are often employed in natural language processing (NLP) tasks. Some of them are as follows:

* [Naive Bayes:](https://www.geeksforgeeks.org/naive-bayes-classifiers/) Naive Bayes is a probabilistic technique that is extensively used in NLP for text classification tasks. It computes the likelihood of a document belonging to a specific class based on the presence of words or features in the document.
* [Support Vector Machines (SVM)](https://www.geeksforgeeks.org/support-vector-machine-algorithm/): SVM is a supervised learning method that can be used for text classification, sentiment analysis, and named entity recognition. Based on the given set of features, SVM finds a hyperplane that splits data points into various classes.
* [Decision Trees:](https://www.geeksforgeeks.org/decision-tree/) Decision trees are commonly used for tasks such as sentiment analysis, and information extraction. These algorithms build a tree-like model based on an order of decisions and feature conditions, which helps in making predictions or classifications.
* [Random Forests:](https://www.geeksforgeeks.org/random-forest-classifier-using-scikit-learn/) Random forests are a type of ensemble learning that combines multiple decision trees to improve accuracy and reduce overfitting.  They can be applied to the tasks like text classification, named entity recognition, and sentiment analysis.
* [Recurrent Neural Networks (RNN):](https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/) RNNs are a type of neural network architecture that are often used in sequence-based NLP tasks like language modelling, machine translation, and sentiment analysis. RNNs can capture temporal dependencies and context within a word sequence.
* [Long Short-Term Memory (LSTM)](https://www.geeksforgeeks.org/long-short-term-memory-lstm-rnn-in-tensorflow/): LSTMs are a type of recurrent neural network that was developed to deal with the vanishing gradient problem of RNN. LSTMs are useful for capturing long-term dependencies in sequences, and they have been used in applications such as machine translation, named entity identification, and sentiment analysis.
* [Transformer](https://www.geeksforgeeks.org/transformer-neural-network-in-deep-learning-overview/): Transformers are a relatively recent architecture that has gained significant attention in NLP. By exploiting self-attention processes to capture contextual relationships in text, transformers such as the BERT (Bidirectional Encoder Representations from Transformers) model have achieved state-of-the-art performance in a wide range of NLP tasks.

## **23. What is Sequence Labelling in NLP?**

Sequence labelling is one of the fundamental NLP tasks in which, categorical labels are assigned to each individual element in a sequence. The sequence can represent various linguistic units such as words, characters, sentences, or paragraphs.

Sequence labelling in NLP includes the following tasks.

* Part-of-Speech Tagging (POS Tagging): In which part-of-speech tags (e.g., noun, verb, adjective) are assigned to each word in a sentence.
* Named Entity Recognition (NER): In which named entities like person names, locations, organizations, or dates are recognized and tagged in the sentences.
* Chunking: Words are organized into syntactic units or “chunks” based on their grammatical roles (for example, noun phrase, verb phrase).
* Semantic Role Labeling (SRL): In which, words or phrases in a sentence are labelled based on their semantic roles like Teacher, Doctor, Engineer, Lawyer etc
* Speech Tagging: In speech processing tasks such as speech recognition or phoneme classification, labels are assigned to phonetic units or acoustic segments.

Machine learning models like Conditional Random Fields (CRFs), Hidden Markov Models (HMMs), recurrent neural networks (RNNs), or transformers are used for sequence labelling tasks. These models learn from the labelled training data to make predictions on unseen data.

## **24.What is topic modelling in NLP?**

Topic modelling is Natural Language Processing task used to discover hidden topics from large text documents. It is an unsupervised technique, which takes unlabeled text data as inputs and applies the probabilistic models that represent the probability of each document being a mixture of topics. For example, A document could have a 60% chance of being about neural networks, a 20% chance of being about Natural Language processing, and a 20% chance of being about anything else.

Where each topic will be distributed over words means each topic is a list of words, and each word has a probability associated with it. and the words that have the highest probabilities in a topic are the words that are most likely to be used to describe that topic. For example, the words like “neural”, “RNN”, and “architecture” are the keywords for neural networks and the words like ‘language”, and “sentiment” are the keywords for Natural Language processing.

There are a number of topic modelling algorithms but two of the most popular topic modelling algorithms are as follows:

* [**Latent Dirichlet Allocation (LDA)**](https://www.geeksforgeeks.org/latent-dirichlet-allocation/)**:**LDA is based on the idea that each text in the corpus is a mash-up of various topics and that each word in the document is derived from one of those topics. It is assumed that there is an unobservable (latent) set of topics and each document is generated by Topic Selection or Word Generation.
* [**Non-Negative Matrix Factorization (NMF)**](https://www.geeksforgeeks.org/non-negative-matrix-factorization/)**:** NMF is a matrix factorization technique that approximates the term-document matrix (where rows represent documents and columns represent words) into two non-negative matrices: one representing the topic-word relationships and the other the document-topic relationships. NMF aims to identify representative topics and weights for each document.

Topic modelling is especially effective for huge text collections when manually inspecting and categorising each document would be impracticable and time-consuming. We can acquire insights into the primary topics and structures of text data by using topic modelling, making it easier to organise, search, and analyse enormous amounts of unstructured text.

## **25. What is the GPT?**

[GPT](https://www.geeksforgeeks.org/gpt-4-vs-gpt-3/) stands for “Generative Pre-trained Transformer”. It refers to a collection of large language models created by OpenAI. It is trained on a massive dataset of text and code, which allows it to generate text, generate code, translate languages, and write many types of creative content, as well as answer questions in an informative manner. The GPT series includes various models, the most well-known and commonly utilised of which are the GPT-2 and GPT-3.

GPT models are built on the Transformer architecture, which allows them to efficiently capture long-term dependencies and contextual information in text. These models are pre-trained on a large corpus of text data from the internet, which enables them to learn the underlying patterns and structures of language.

**Advanced NLP Interview Questions for Experienced**

## **26. What are word embeddings in NLP?**

[Word embeddings](https://www.geeksforgeeks.org/word-embeddings-in-nlp/) in NLP are defined as the dense, low-dimensional vector representations of words that capture semantic and contextual information about words in a language. It is trained using big text corpora through unsupervised or supervised methods to represent words in a numerical format that can be processed by machine learning models.

The main goal of Word embeddings is to capture relationships and similarities between words by representing them as dense vectors in a continuous vector space. These vector representations are acquired using the distributional hypothesis, which states that words with similar meanings tend to occur in similar contexts. Some of the popular pre-trained word embeddings are Word2Vec, GloVe (Global Vectors for Word Representation), or FastText. The advantages of word embedding over the traditional text vectorization technique are as follows:

* It can capture the Semantic Similarity between the words
* It is capable of capturing syntactic links between words. Vector operations such as “king” – “man” + “woman” may produce a vector similar to the vector for “queen,” capturing the gender analogy.
* Compared to one-shot encoding, it has reduced the dimensionality of word representations. Instead of high-dimensional sparse vectors, word embeddings typically have a fixed length and represent words as dense vectors.
* It can be generalized to represent words that they have not been trained on i.e. out-of-vocabulary words. This is done by using the learned word associations to place new words in the vector space near words that they are semantically or syntactically similar to.

## **27. What are the various algorithms used for training word embeddings?**

There are various approaches that are typically used for training word embeddings, which are dense vector representations of words in a continuous vector space. Some of the popular word embedding algorithms are as follows:

* [**Word2Vec**](https://www.geeksforgeeks.org/python-word-embedding-using-word2vec/): Word2vec is a common approach for generating vector representations of words that reflect their meaning and relationships. Word2vec learns embeddings using a shallow neural network and follows two approaches: CBOW and Skip-gram
  + CBOW (Continuous Bag-of-Words) predicts a target word based on its context words.
  + Skip-gram predicts context words given a target word.
* **GloVe**: GloVe (Global Vectors for Word Representation) is a word embedding model that is similar to Word2vec. GloVe, on the other hand, uses  objective function that constructs a co-occurrence matrix based on the statistics of word co-occurrences in a large corpus. The co-occurrence matrix is a square matrix where each entry represents the number of times two words co-occur in a window of a certain size. GloVe then performs matrix factorization on the co-occurrence matrix. Matrix factorization is a technique for finding a low-dimensional representation of a high-dimensional matrix. In the case of GloVe, the low-dimensional representation is a vector representation for each word in the corpus. The word embeddings are learned by minimizing a loss function that measures the difference between the predicted co-occurrence probabilities and the actual co-occurrence probabilities. This makes GloVe more robust to noise and less sensitive to the order of words in a sentence.
* [**FastText**](https://www.geeksforgeeks.org/fasttext-working-and-implementation/): FastText is a Word2vec extension that includes subword information. It represents words as bags of character n-grams, allowing it to handle out-of-vocabulary terms and capture morphological information. During training, FastText considers subword information as well as word context..
* [**ELMo**](https://www.geeksforgeeks.org/overview-of-word-embedding-using-embeddings-from-language-models-elmo/): ELMo is a deeply contextualised word embedding model that generates context-dependent word representations. It generates word embeddings that capture both semantic and syntactic information based on the context of the word using bidirectional language models.
* [**BERT**](https://www.geeksforgeeks.org/explanation-of-bert-model-nlp/): A transformer-based model called BERT (Bidirectional Encoder Representations from Transformers) learns contextualised word embeddings. BERT is trained on a large corpus by anticipating masked terms inside a sentence and gaining knowledge about the bidirectional context. The generated embeddings achieve state-of-the-art performance in many NLP tasks and capture extensive contextual information.

## **28. How to handle out-of-vocabulary (OOV) words in NLP?**

OOV words are words that are missing in a language model’s vocabulary or the training data it was trained on. Here are a few approaches to handling OOV words in NLP:

1. **Character-level models:**Character-level models can be used in place of word-level representations. In this method, words are broken down into individual characters, and the model learns representations based on character sequences. As a result, the model can handle OOV words since it can generalize from known character patterns.
2. **Subword tokenization:** Byte-Pair Encoding (BPE) and WordPiece are two subword tokenization algorithms that divide words into smaller subword units based on their frequency in the training data. This method enables the model to handle OOV words by representing them as a combination of subwords that it comes across during training.
3. **Unknown token:** Use a special token, frequently referred to as an “unknown” token or “UNK,” to represent any OOV term that appears during inference. Every time the model comes across an OOV term, it replaces it with the unidentified token and keeps processing. The model is still able to generate relevant output even though this technique doesn’t explicitly define the meaning of the OOV word.
4. **External knowledge:**When dealing with OOV terms, using external knowledge resources, like a knowledge graph or an external dictionary, can be helpful. We need to try to look up a word’s definition or relevant information in the external knowledge source when we come across an OOV word.
5. **Fine-tuning:**We can fine-tune using the pre-trained language model with domain-specific or task-specific data that includes OOV words. By incorporating OOV words in the fine-tuning process, we expose the model to these words and increase its capacity to handle them.

## **29. What is the difference between a word-level and character-level language model?**

The main difference between a word-level and a character-level language model is how text is represented. A character-level language model represents text as a sequence of characters, whereas a word-level language model represents text as a sequence of words.

Word-level language models are often easier to interpret and more efficient to train. They are, however, less accurate than character-level language models because they cannot capture the intricacies of the text that are stored in the character order. Character-level language models are more accurate than word-level language models, but they are more complex to train and interpret. They are also more sensitive to noise in the text, as a slight alteration in a character can have a large impact on the meaning of the text.

The key differences between word-level and character-level language models are:

|  | **Word-level** | **Character-level** |
| --- | --- | --- |
| **Text representation** | Sequence of words | Sequence of characters |
| **Interpretability** | Easier to interpret | More difficult to interpret |
| **Sensitivity to noise** | Less sensitive | More sensitive |
| **Vocabulary** | Fixed vocabulary of words | No predefined vocabulary |
| **Out-of-vocabulary (OOV) handling** | Struggles with OOV words | Naturally handles OOV words |
| **Generalization** | Captures semantic relationships between words | Better at handling morphological details |
| **Training complexity** | Smaller input/output space, less computationally intensive | Larger input/output space, more computationally intensive |
| **Applications** | Well-suited for tasks requiring word-level understanding | Suitable for tasks requiring fine-grained details or morphological variations |

## **30. What is word sense disambiguation?**

The task of determining which sense of a word is intended in a given context is known as [word sense disambiguation (WSD)](https://www.geeksforgeeks.org/word-sense-disambiguation-in-natural-language-processing/). This is a challenging task because many words have several meanings that can only be determined by considering the context in which the word is used.

For example, the word “bank” can be used to refer to a variety of things, including “a financial institution,” “a riverbank,” and “a slope.” The term “bank” in the sentence “I went to the bank to deposit my money” should be understood to mean “a financial institution.” This is so because the sentence’s context implies that the speaker is on their way to a location where they can deposit money.

## **31. What is co-reference resolution?**

Co-reference resolution is a natural language processing (NLP) task that involves identifying all expressions in a text that refer to the same entity. In other words, it tries to determine whether words or phrases in a text, typically pronouns or noun phrases, correspond to the same real-world thing. For example, the pronoun “he” in the sentence “Pawan Gunjan has compiled this article, He had done lots of research on Various NLP interview questions” refers to Pawan Gunjan himself. Co-reference resolution automatically identifies such linkages and establishes that “He” refers to “Pawan Gunjan” in all instances.

Co-reference resolution is used in information extraction, question answering, summarization, and dialogue systems because it helps to generate more accurate and context-aware representations of text data. It is an important part of systems that require a more in-depth understanding of the relationships between entities in large text corpora.

## **32.What is information extraction?**

[Information extraction](https://www.geeksforgeeks.org/difference-between-information-retrieval-and-information-extraction/) is a natural language processing task used to extract specific pieces of information like names, dates, locations, and relationships etc from unstructured or semi-structured texts.

Natural language is often ambiguous and can be interpreted in a variety of ways, which makes IE a difficult process. Some of the common techniques used for information extraction include:

* **Named entity recognition (NER):** In NER, named entities like people, organizations, locations, dates, or other specific categories are recognized from the text documents. For NER problems, a variety of machine learning techniques, including conditional random fields (CRF), support vector machines (SVM), and deep learning models, are frequently used.
* **Relationship extraction:**In relationship extraction, the connections between the stated text are identified. I figure out the relations different kinds of relationships between various things like “is working at”, “lives in” etc.
* **Coreference resolution:** Coreference resolution is the task of identifying the referents of pronouns and other anaphoric expressions in the text. A coreference resolution system, for example, might be able to figure out that the pronoun “he” in a sentence relates to the person “John” who was named earlier in the text.
* **Deep Learning-based Approaches:**To perform information extraction tasks, deep learning models such as recurrent neural networks (RNNs), transformer-based architectures (e.g., BERT, GPT), and deep neural networks have been used. These models can learn patterns and representations from data automatically, allowing them to manage complicated and diverse textual material.

## **33. What is the Hidden Markov Model, and How it’s helpful in NLP tasks?**

[Hidden Markov Model](https://www.geeksforgeeks.org/hidden-markov-model-in-machine-learning/) is a probabilistic model based on the Markov Chain Rule used for modelling sequential data like characters, words, and sentences by computing the probability distribution of sequences.

Markov chain uses the Markov assumptions which state that the probabilities future state of the system only depends on its present state, not on any past state of the system. This assumption simplifies the modelling process by reducing the amount of information needed to predict future states.

The underlying process in an HMM is represented by a set of hidden states that are not directly observable. Based on the hidden states, the observed data, such as characters, words, or phrases, are generated.

Hidden Markov Models consist of two key components:

1. Transition Probabilities: The transition probabilities in Hidden Markov Models(HMMs) represents the likelihood of moving from one hidden state to another. It captures the dependencies or relationships between adjacent states in the sequence. In part-of-speech tagging, for example, the HMM’s hidden states represent distinct part-of-speech tags, and the transition probabilities indicate the likelihood of transitioning from one part-of-speech tag to another.
2. Emission Probabilities: In HMMs, emission probabilities define the likelihood of observing specific symbols (characters, words, etc.) given a particular hidden state. The link between the hidden states and the observable symbols is encoded by these probabilities.
3. Emission probabilities are often used in NLP to represent the relationship between words and linguistic features such as part-of-speech tags or other linguistic variables. The HMM captures the likelihood of generating an observable symbol (e.g., word) from a specific hidden state (e.g., part-of-speech tag) by calculating the emission probabilities.

Hidden Markov Models (HMMs) estimate transition and emission probabilities from labelled data using approaches such as the Baum-Welch algorithm. Inference algorithms like Viterbi and Forward-Backward are used to determine the most likely sequence of hidden states given observed symbols. HMMs are used to represent sequential data and have been implemented in NLP applications such as part-of-speech tagging. However, advanced models, such as CRFs and neural networks, frequently beat HMMs due to their flexibility and ability to capture richer dependencies.

## **34. What is the conditional random field (CRF) model in NLP?**

[Conditional Random Fields](https://www.geeksforgeeks.org/conditional-random-fields-crfs-for-pos-tagging-in-nlp/) are a probabilistic graphical model that is designed to predict the sequence of labels for a given sequence of observations. It is well-suited for prediction tasks in which contextual information or dependencies among neighbouring elements are crucial.

CRFs are an extension of Hidden Markov Models (HMMs) that allow for the modelling of more complex relationships between labels in a sequence. It is specifically designed to capture dependencies between non-consecutive labels, whereas HMMs presume a Markov property in which the current state is only dependent on the past state. This makes CRFs more adaptable and suitable for capturing long-term dependencies and complicated label interactions.

In a CRF model, the labels and observations are represented as a graph. The nodes in the graph represent the labels, and the edges represent the dependencies between the labels. The model assigns weights to features that capture relevant information about the observations and labels.

During training, the CRF model learns the weights by maximizing the conditional log-likelihood of the labelled training data. This process involves optimization algorithms such as gradient descent or the iterative scaling algorithm.

During inference, given an input sequence, the CRF model calculates the conditional probabilities of different label sequences. Algorithms like the Viterbi algorithm efficiently find the most likely label sequence based on these probabilities.

CRFs have demonstrated high performance in a variety of sequence labelling tasks like named entity identification, part-of-speech tagging, and others.

## **35. What is a recurrent neural network (RNN)?**

[Recurrent Neural Networks](https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/) are the type of artificial neural network that is specifically built to work with sequential or time series data. It is utilised in natural language processing activities such as language translation, speech recognition, sentiment analysis, natural language production, summary writing, and so on. It differs from feedforward neural networks in that the input data in RNN does not only flow in a single direction but also has a loop or cycle inside its design that has “memory” that preserves information over time. As a result, the RNN can handle data where context is critical, such as natural languages.

RNNs work by analysing input sequences one element at a time while keeping track in a hidden state that provides a summary of the sequence’s previous elements. At each time step, the hidden state is updated based on the current input and the prior hidden state. RNNs can thus capture the temporal connections between sequence items and use that knowledge to produce predictions.

## **36. How does the Backpropagation through time work in RNN?**

[Backpropagation through time(BPTT)](https://www.geeksforgeeks.org/ml-back-propagation-through-time/) propagates gradient information across the RNN’s recurrent connections over a sequence of input data. Let’s understand step by step process for BPTT.

1. Forward Pass: The input sequence is fed into the RNN one element at a time, starting from the first element. Each input element is processed through the recurrent connections, and the hidden state of the RNN is updated.
2. Hidden State Sequence: The hidden state of the RNN is maintained and carried over from one time step to the next. It contains information about the previous inputs and hidden states in the sequence.
3. Output Calculation: The updated hidden state is used to compute the output at each time step.
4. Loss Calculation: At the end of the sequence, the predicted output is compared to the target output, and a loss value is calculated using a suitable loss function, such as mean squared error or cross-entropy loss.
5. Backpropagation: The loss is then backpropagated through time, starting from the last time step and moving backwards in time. The gradients of the loss with respect to the parameters of the RNN are calculated at each time step.
6. Weight Update: The gradients are accumulated over the entire sequence, and the weights of the RNN are updated using an optimization algorithm such as gradient descent or its variants.
7. Repeat: The process is repeated for a specified number of epochs or until convergence, during this the training data is iterated through several times.

During the backpropagation step, the gradients at each time step are obtained and used to update the weights of the recurrent connections. This accumulation of gradients over numerous time steps allows the RNN to learn and capture dependencies and patterns in sequential data.

## **37. What are the limitations of a standard RNN?**

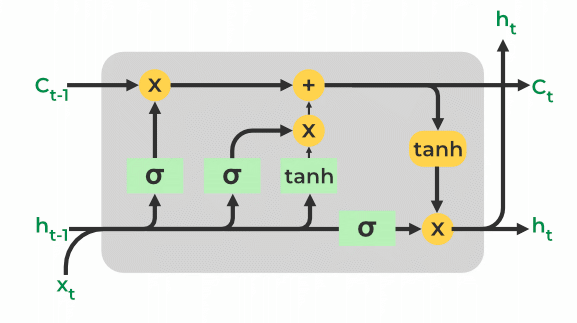
Standard [RNNs (Recurrent Neural Networks)](https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/) have several limitations that can make them unsuitable for certain applications:

1. Vanishing Gradient Problem: Standard RNNs are vulnerable to the vanishing gradient problem, in which gradients decrease exponentially as they propagate backwards through time. Because of this issue, it is difficult for the network to capture and transmit long-term dependencies across multiple time steps during training.
2. Exploding Gradient Problem: RNNs, on the other hand, can suffer from the expanding gradient problem, in which gradients get exceedingly big and cause unstable training. This issue can cause the network to converge slowly or fail to converge at all.
3. Short-Term Memory: Standard RNNs have limited memory and fail to remember information from previous time steps. Because of this limitation, they have difficulty capturing long-term dependencies in sequences, limiting their ability to model complicated relationships that span a significant number of time steps.

## **38. What is a long short-term memory (LSTM) network?**

A [Long Short-Term Memory (LSTM)](https://www.geeksforgeeks.org/long-short-term-memory-lstm-rnn-in-tensorflow/) network is a type of recurrent neural network (RNN) architecture that is designed to solve the vanishing gradient problem and capture long-term dependencies in sequential data. LSTM networks are particularly effective in tasks that involve processing and understanding sequential data, such as natural language processing and speech recognition.

The key idea behind LSTMs is the integration of a memory cell, which acts as a memory unit capable of retaining information for an extended period. The memory cell is controlled by three gates: the input gate, the forget gate, and the output gate.



The input gate controls how much new information should be stored in the memory cell. The forget gate determines which information from the memory cell should be destroyed or forgotten. The output gate controls how much information is output from the memory cell to the next time step. These gates are controlled by activation functions, which are commonly sigmoid and tanh functions, and allow the LSTM to selectively update, forget, and output data from the memory cell.

## **39. What is the GRU model in NLP?**

The [Gated Recurrent Unit (GRU)](https://www.geeksforgeeks.org/gated-recurrent-unit-networks/) model is a type of recurrent neural network (RNN) architecture that has been widely used in natural language processing (NLP) tasks. It is designed to address the vanishing gradient problem and capture long-term dependencies in sequential data.

GRU is similar to LSTM in that it incorporates gating mechanisms, but it has a simplified architecture with fewer gates, making it computationally more efficient and easier to train. The GRU model consists of the following components:

1. **Hidden State:** The hidden state ℎ𝑡−1  *ht*−1​ in GRU represents the learned representation or memory of the input sequence up to the current time step. It retains and passes information from the past to the present.
2. **Update Gate:** The update gate in GRU controls the flow of information from the past hidden state to the current time step. It determines how much of the previous information should be retained and how much new information should be incorporated.
3. **Reset Gate:** The reset gate in GRU determines how much of the past information should be discarded or forgotten. It helps in removing irrelevant information from the previous hidden state.
4. **Candidate Activation:**The candidate activation represents the new information to be added to the hidden state ℎ𝑡‘  *ht*‘​ . It is computed based on the current input and a transformed version of the previous hidden state using the reset gate.

GRU models have been effective in NLP applications like language modelling, sentiment analysis, machine translation, and text generation. They are particularly useful in situations when it is essential to capture long-term dependencies and understand the context. Due to its simplicity and computational efficiency, GRU makes it a popular choice in NLP research and applications.

## **40. What is the sequence-to-sequence (Seq2Seq) model in NLP?**

[Sequence-to-sequence (Seq2Seq)](https://www.geeksforgeeks.org/seq2seq-model-in-machine-learning/) is a type of neural network that is used for natural language processing (NLP) tasks. It is a type of recurrent neural network (RNN) that can learn long-term word relationships. This makes it ideal for tasks like machine translation, text summarization, and question answering.

The model is composed of two major parts: an encoder and a decoder. Here’s how the Seq2Seq model works:

1. **Encoder**: The encoder transforms the input sequence, such as a sentence in the source language, into a fixed-length vector representation known as the “context vector” or “thought vector”. To capture sequential information from the input, the encoder commonly employs recurrent neural networks (RNNs) such as Long Short-Term Memory (LSTM) or Gated Recurrent Units (GRU).
2. **Context Vector:**The encoder’s context vector acts as a summary or representation of the input sequence. It encodes the meaning and important information from the input sequence into a fixed-size vector, regardless of the length of the input.
3. **Decoder**: The decoder uses the encoder’s context vector to build the output sequence, which could be a translation or a summarised version. It is another RNN-based network that creates the output sequence one token at a time. At each step, the decoder can be conditioned on the context vector, which serves as an initial hidden state.

During training, the decoder is fed ground truth tokens from the target sequence at each step. Backpropagation through time (BPTT) is a technique commonly used to train Seq2Seq models. The model is optimized to minimize the difference between the predicted output sequence and the actual target sequence.

The Seq2Seq model is used during prediction or generation to construct the output sequence word by word, with each predicted word given back into the model as input for the subsequent step. The process is repeated until either an end-of-sequence token or a predetermined maximum length is achieved.

## **41. How does the attention mechanism helpful in NLP?**

An [attention mechanism](https://www.geeksforgeeks.org/ml-attention-mechanism/) is a kind of neural network that uses an additional attention layer within an Encoder-Decoder neural network that enables the model to focus on specific parts of the input while performing a task. It achieves this by dynamically assigning weights to different elements in the input, indicating their relative importance or relevance. This selective attention allows the model to focus on relevant information, capture dependencies, and analyze relationships within the data.

The attention mechanism is particularly valuable in tasks involving sequential or structured data, such as natural language processing or computer vision, where long-term dependencies and contextual information are crucial for achieving high performance. By allowing the model to selectively attend to important features or contexts, it improves the model’s ability to handle complex relationships and dependencies in the data, leading to better overall performance in various tasks.

## **42. What is the Transformer model?**

[Transformer](https://www.geeksforgeeks.org/transformer-neural-network-in-deep-learning-overview/) is one of the fundamental models in NLP based on the attention mechanism, which allows it to capture long-range dependencies in sequences more effectively than traditional recurrent neural networks (RNNs). It has given state-of-the-art results in various NLP tasks like word embedding, machine translation, text summarization, question answering etc.

Some of the key advantages of using a Transformer are as follows:

* **Parallelization**: The self-attention mechanism allows the model to process words in parallel, which makes it significantly faster to train compared to sequential models like RNNs.
* **Long-Range Dependencies:** The attention mechanism enables the Transformer to effectively capture long-range dependencies in sequences, which makes it suitable for tasks where long-term context is essential.
* **State-of-the-Art Performance:**Transformer-based models have achieved state-of-the-art performance in various NLP tasks, such as machine translation, language modelling, text generation, and sentiment analysis.

The key components of the Transformer model are as follows:

* Self-Attention Mechanism:
* Encoder-Decoder Network:
* Multi-head Attention:
* Positional Encoding
* Feed-Forward Neural Networks
* Layer Normalization and Residual Connections

## **43. What is the role of the self-attention mechanism in Transformers?**

The [self-attention mechanism](https://www.geeksforgeeks.org/self-attention-in-nlp/) is a powerful tool that allows the Transformer model to capture long-range dependencies in sequences. It allows each word in the input sequence to attend to all other words in the same sequence, and the model learns to assign weights to each word based on its relevance to the others. This enables the model to capture both short-term and long-term dependencies, which is critical for many NLP applications.

## **44. What is the purpose of the multi-head attention mechanism in Transformers?**

The purpose of the [multi-head attention mechanism](https://www.geeksforgeeks.org/ml-attention-mechanism/) in Transformers is to allow the model to recognize different types of correlations and patterns in the input sequence. In both the encoder and decoder, the Transformer model uses multiple attention heads. This enables the model to recognise different types of correlations and patterns in the input sequence. Each attention head learns to pay attention to different parts of the input, allowing the model to capture a wide range of characteristics and dependencies.

The multi-head attention mechanism helps the model in learning richer and more contextually relevant representations, resulting in improved performance on a variety of natural language processing (NLP) tasks.

## **45. What are positional encodings in Transformers, and why are they necessary?**

The [transformer](https://www.geeksforgeeks.org/transformer-neural-network-in-deep-learning-overview/) model processes the input sequence in parallel, so that lacks the inherent understanding of word order like the sequential model recurrent neural networks (RNNs), LSTM possess. So, that. it requires a method to express the positional information explicitly.

Positional encoding is applied to the input embeddings to offer this positional information like the relative or absolute position of each word in the sequence to the model. These encodings are typically learnt and can take several forms, including sine and cosine functions or learned embeddings. This enables the model to learn the order of the words in the sequence, which is critical for many NLP tasks.

## **46. Describe the architecture of the Transformer model.**

The architecture of the Transformer model is based on self-attention and feed-forward neural network concepts. It is made up of an encoder and a decoder, both of which are composed of multiple layers, each containing self-attention and feed-forward sub-layers. The model’s design encourages parallelization, resulting in more efficient training and improved performance on tasks involving sequential data, such as natural language processing (NLP) tasks.

The architecture can be described in depth below:

1. [**Encoder**](https://www.geeksforgeeks.org/difference-between-encoder-and-decoder/):
   * Input Embeddings: The encoder takes an input sequence of tokens (e.g., words) as input and transforms each token into a vector representation known as an embedding. Positional encoding is used in these embeddings to preserve the order of the words in the sequence.
   * Self-Attention Layers: An encoder consists of multiple self-attention layers and each self-attention layer is used to capture relationships and dependencies between words in the sequence.
   * Feed-Forward Layers: After the self-attention step, the output representations of the self-attention layer are fed into a feed-forward neural network. This network applies the non-linear transformations to each word’s contextualised representation independently.
   * Layer Normalization and Residual Connections: Residual connections and layer normalisation are used to back up the self-attention and feed-forward layers. The residual connections in deep networks help to mitigate the vanishing gradient problem, and layer normalisation stabilises the training process.
2. [**Decoder**](https://www.geeksforgeeks.org/difference-between-encoder-and-decoder/):
   * Input Embeddings: Similar to the encoder, the decoder takes an input sequence and transforms each token into embeddings with positional encoding.
   * Masked Self-Attention: Unlike the encoder, the decoder uses masked self-attention in the self-attention layers. This masking ensures that the decoder can only attend to places before the current word during training, preventing the model from seeing future tokens during generation.
   * Cross-Attention Layers: Cross-attention layers in the decoder allow it to attend to the encoder’s output, which enables the model to use information from the input sequence during output sequence generation.
   * Feed-Forward Layers: Similar to the encoder, the decoder’s self-attention output passes through feed-forward neural networks.
   * Layer Normalization and Residual Connections: The decoder also includes residual connections and layer normalization to help in training and improve model stability.
3. **Final Output Layer:**
   * Softmax Layer: The final output layer is a softmax layer that transforms the decoder’s representations into probability distributions over the vocabulary. This enables the model to predict the most likely token for each position in the output sequence.

Overall, the Transformer’s architecture enables it to successfully handle long-range dependencies in sequences and execute parallel computations, making it highly efficient and powerful for a variety of sequence-to-sequence tasks. The model has been successfully used for machine translation, language modelling, text generation, question answering, and a variety of other NLP tasks, with state-of-the-art results.

## **47. What is the difference between a generative and discriminative model in NLP?**

Both generative and discriminative models are the types of [machine learning](https://www.geeksforgeeks.org/machine-learning/) models used for different purposes in the field of natural language processing (NLP).

[Generative models](https://www.geeksforgeeks.org/the-difference-between-generative-and-discriminative-machine-learning-algorithms/) are trained to generate new data that is similar to the data that was used to train them.  For example, a generative model could be trained on a dataset of text and code and then used to generate new text or code that is similar to the text and code in the dataset. Generative models are often used for tasks such as text generation, machine translation, and creative writing.

[Discriminative models](https://www.geeksforgeeks.org/the-difference-between-generative-and-discriminative-machine-learning-algorithms/) are trained to recognise different types of data. A discriminative model. For example, a discriminative model could be trained on a dataset of labelled text and then used to classify new text as either spam or ham. Discriminative models are often used for tasks such as text classification, sentiment analysis, and question answering.

The key differences between generative and discriminative models in NLP are as follows:

|  | **Generative Models** | **Discriminative Models** |
| --- | --- | --- |
| **Purpose** | Generate new data that is similar to the training data. | Distinguish between different classes or categories of data. |
| **Training** | Learn the joint probability distribution of input and output data to generate new samples. | Learn the conditional probability distribution of the output labels given the input data. |
| **Examples** | Text generation, machine translation, creative writing, Chatbots, text summarization, and language modelling. | Text classification, sentiment analysis, and named entity recognition. |

## **48. What is machine translation, and how does it is performed?**

[Machine translation](https://www.geeksforgeeks.org/machine-translation-of-languages-in-artificial-intelligence/) is the process of automatically translating text or speech from one language to another using a computer or machine learning model.

There are three techniques for machine translation:

* Rule-based machine translation (RBMT): RBMT systems use a set of rules to translate text from one language to another.
* Statistical machine translation (SMT): SMT systems use statistical models to calculate the probability of a given translation being correct.
* Neural machine translation (NMT): Neural machine translation (NMT) is a recent technique of machine translation have been proven to be more accurate than RBMT and SMT systems, In recent years, neural machine translation (NMT), powered by deep learning models such as the Transformer, are becoming increasingly popular.

## **49. What is the BLEU score?**

[BLEU](https://www.geeksforgeeks.org/nlp-bleu-score-for-evaluating-neural-machine-translation-python/) stands for “Bilingual Evaluation Understudy”. It is a metric invented by IBM in 2001 for evaluating the quality of a machine translation. It measures the similarity between machine-generated translations with the professional human translation. It was one of the first metrics whose results are very much correlated with human judgement.

The BLEU score is measured by comparing the n-grams (sequences of n words) in the machine-translated text to the n-grams in the reference text. The higher BLEU Score signifies, that the machine-translated text is more similar to the reference text.

The BLEU (Bilingual Evaluation Understudy) score is calculated using n-gram precision and a brevity penalty.

* N-gram Precision: The n-gram precision is the ratio of matching n-grams in the machine-generated translation to the total number of n-grams in the reference translation. The number of unigrams, bigrams, trigrams, and four-grams (i=1,…,4) that coincide with their n-gram counterpart in the reference translations is measured by the n-gram overlap.  
  precision𝑖=Count of matching n-gramscount of all n-grams in the machine translation  precision*i*​=count of all n-grams in the machine translationCount of matching n-grams​   
  For BLEU score precision𝑖  precision*i*​ is calculated for the I ranging (1 to N). Usually, the N value will be up to 4.
* Brevity Penalty: Brevity Penalty measures the length difference between machine-generated translations and reference translations. While finding the BLEU score, It penalizes the machine-generated translations if that is found too short compared to the reference translation’s length with exponential decay.  
  brevity-penalty=min⁡(1,exp⁡(1−Reference lengthMachine translation length)))brevity-penalty=min(1,exp(1−Machine translation length)Reference length​))
* BLEU Score: The BLEU score is calculated by taking the geometric mean of the individual n-gram precisions and then adjusting it with the brevity penalty.  
  BLEU=brevity-penalty×exp⁡[∑𝑖=1𝑁log⁡(precision𝑖)𝑁]=brevity-penalty×exp⁡[log⁡(∏𝑖=1𝑁precision𝑖)𝑁]=brevity-penalty×(∏𝑖=1𝑁precision𝑖)1𝑁  BLEU​=brevity-penalty×exp[*N*∑*i*=1*N*​log(precision*i*​)​]=brevity-penalty×exp​*N*log(∏*i*=1*N*​precision*i*​)​​=brevity-penalty×(*i*=1∏*N*​precision*i*​)*N*1​​   
  Here, N is the maximum n-gram size, (usually 4).

The BLEU score goes from 0 to 1, with higher values indicating better translation quality and 1 signifying a perfect match to the reference translation

## **50. List out the popular NLP task and their corresponding evaluation metrics.**

Natural Language Processing (NLP) involves a wide range of tasks, each with its own set of objectives and evaluation criteria. Below is a list of common NLP tasks along with some typical evaluation metrics used to assess their performance:

| **Natural Language Processing(NLP) Tasks** | **Evaluation Metric** |
| --- | --- |
| Part-of-Speech Tagging (POS Tagging) or Named Entity Recognition (NER) | Accuracy, F1-score, Precision, Recall |
| Dependency Parsing | UAS (Unlabeled Attachment Score), LAS (Labeled Attachment Score) |
| Coreference resolution | B-CUBED, MUC, CEAF |
| Text Classification or Sentiment Analysis | Accuracy, F1-score, Precision, Recall |
| Machine Translation | BLEU (Bilingual Evaluation Understudy), METEOR (Metric for Evaluation of Translation with Explicit Ordering) |
| Text Summarization | ROUGE (Recall-Oriented Understudy for Gisting Evaluation), BLEU |
| Question Answering | F1-score, Precision, Recall, MRR(Mean Reciprocal Rank) |
| Text Generation | Human evaluation (subjective assessment), perplexity (for language models) |
| Information Retrieval | Precision, Recall, F1-score, Mean Average Precision (MAP) |
| Natural language inference (NLI) | Accuracy, precision, recall, F1-score, Matthews correlation coefficient (MCC) |
| Topic Modeling | Coherence Score, Perplexity |
| Speech Recognition | Word Error Rate (WER) |
| Speech Synthesis (Text-to-Speech) | Mean Opinion Score (MOS) |

The brief explanations of each of the evaluation metrics are as follows:

* **Accuracy**: Accuracy is the percentage of predictions that are correct.
* **Precision**: Precision is the percentage of correct predictions out of all the predictions that were made.
* **Recall**: Recall is the percentage of correct predictions out of all the positive cases.
* **F1-score**: F1-score is the harmonic mean of precision and recall.
* **MAP(Mean Average Precision)**: MAP computes the average precision for each query and then averages those precisions over all queries.
* **MUC(Mention-based Understudy for Coreference)**: MUC is a metric for coreference resolution that measures the number of mentions that are correctly identified and linked.
* **B-CUBED**: B-cubed is a metric for coreference resolution that measures the number of mentions that are correctly identified, linked, and ordered.
* **CEAF**: CEAF is a metric for coreference resolution that measures the similarity between the predicted coreference chains and the gold standard coreference chains.
* **ROC AUC:** ROC AUC is a metric for binary classification that measures the area under the receiver operating characteristic curve.
* **MRR**: MRR is a metric for question answering that measures the mean reciprocal rank of the top-k-ranked documents.
* **Perplexity**: Perplexity is a language model evaluation metric. It assesses how well a linguistic model predicts a sample or test set of previously unseen data. Lower perplexity values suggest that the language model is more predictive.
* **BLEU**: BLEU is a metric for machine translation that measures the n-gram overlap between the predicted translation and the gold standard translation.
* **METEOR**: METEOR is a metric for machine translation that measures the overlap between the predicted translation and the gold standard translation, taking into account synonyms and stemming.
* **WER(Word Error Rate)**: WER is a metric for machine translation that measures the word error rate of the predicted translation.
* **MCC**: MCC is a metric for natural language inference that measures the Matthews correlation coefficient between the predicted labels and the gold standard labels.
* **ROUGE**: ROUGE is a metric for text summarization that measures the overlap between the predicted summary and the gold standard summary, taking into account n-grams and synonyms.
* **Human Evaluation (Subjective Assessment)**: Human experts or crowd-sourced workers are asked to submit their comments, evaluations, or rankings on many elements of the NLP task’s performance in this technique.

**1. What are the stages in the lifecycle of a natural language processing (NLP) project?**

Following are the stages in the lifecycle of a natural language processing (NLP) project:

A diagram of a model

Description automatically generated

* **Data Collection:** The procedure of collecting, measuring, and evaluating correct insights for research using established approved procedures is referred to as data collection.
* **Data Cleaning:** The practice of correcting or deleting incorrect, corrupted, improperly formatted, duplicate, or incomplete data from a dataset is known as data cleaning.
* **Data Pre-Processing:** The process of converting raw data into a comprehensible format is known as data preparation.
* **Feature Engineering:** Feature engineering is the process of extracting features (characteristics, qualities, and attributes) from raw data using domain expertise.
* **Data Modeling:** The practice of examining data objects and their relationships with other things is known as data modelling. It's utilised to look into the data requirements for various business activities.
* **Model Evaluation:** Model evaluation is an important step in the creation of a model. It aids in the selection of the best model to represent our data and the prediction of how well the chosen model will perform in the future.
* **Model Deployment:** The technical task of exposing an ML model to real-world use is known as model deployment.
* **Monitoring and Updating:** The activity of measuring and analysing production model performance to ensure acceptable quality as defined by the use case is known as machine learning monitoring. It delivers alerts about performance difficulties and assists in diagnosing and resolving the core cause.

**2. What are some of the common NLP tasks?**

Some of the common tasks of NLP include:

* **Machine Translation:**This helps in translating a given piece of text from one language to another.
* **Text Summarization:** Based on a large corpus, this is used to give a short summary that gives an idea of the entire text in the document.
* **Language Modeling:** Based on the history of previous words, this helps uncover what the further sentence will look like. A good example of this is the auto-complete sentences feature in Gmail.
* **Topic Modelling:**This helps uncover the topical structure of a large collection of documents. This indicates what topic a piece of text is actually about.
* **Question Answering:**This helps prepare answers automatically based on a corpus of text, and on a question that is posed.
* **Conversational Agent:** These are basically voice assistants that we commonly see such as Alexa, Siri, Google Assistant, Cortana, etc.
* **Information Retrieval:** This helps in fetching relevant documents based on a user’s search query.
* **Information Extraction:**This is the task of extracting relevant pieces of information from a given text, such as calendar events from emails.
* **Text Classification:**This is used to create a bucket of categories of a given text, based on its content. This is used in a wide variety of AI-based applications such as sentiment analysis and spam detection.

A chart of information on a white background

Description automatically generated with medium confidence

Common NLP Tasks in order of Difficulty

**3. What are the different approaches used to solve NLP problems?**

There are multiple approaches to solving NLP problems. These usually come in 3 categories:

* Heuristics
* Machine learning
* Deep Learning

**4. How do Conversational Agents work?**

The following NLP components are used in Conversational Agents:

* **Speech Recognition and Synthesis:** In the first stage, speech recognition helps convert speech signals to their phonemes, and are then transcribed as words.
* **Natural Language Understanding (NLU):** Here, the transcribed text from stage one is further analysed through AI techniques within the natural language understanding system. Certain NLP tasks such as Named Entity Recognition, Text Classification, Language modelling, etc. come into play here.
* **Dialog Management:** Once the needed information from text is extracted, we move on to the stage of understanding the user’s intent. The user’s response can then be classified by using a text classification system as a pre-defined intent. This helps the conversational agent in figuring out what is actually being asked.
* **Generating Response:** Based on the above stages, the agent generates an appropriate response that is based on a semantic interpretation of the user’s intent.

A diagram of a speech language

Description automatically generated

**5. What is meant by data augmentation? What are some of the ways in which data augmentation can be done in NLP projects?**

NLP has some methods through which we can take a small dataset and use that in order to create more data. This is called data augmentation. In this, we use language properties to create text that is syntactically similar to the source text data.

Some of the ways in which data augmentation can be done in NLP projects are as follows:

* Replacing entities
* TF-IDF–based word replacement
* Adding noise to data
* Back translation
* Synonym replacement
* Bigram flipping

**6. How can data be obtained for NLP projects?**

There are multiple ways in which data can be obtained for NLP projects. Some of them are as follows:

* **Using publicly available datasets:**Datasets for NLP purposes are available on websites like Kaggle as well as Google Datasets.
* **By using data augmentation:** These are used to create additional datasets from existing datasets.
* **Scraping data from the web:** Using coding in Python or other languages once can scrape data from websites that are usually not readily available in a structured form.

**7. What do you mean by Text Extraction and Cleanup?**

The process of extracting raw text from the input data by getting rid of all the other non-textual information, such as markup, metadata, etc., and converting the text to the required encoding format is called **text extraction and cleanup**. Usually, this depends on the format of available data for the required project.

Following are the common ways used for Text Extraction in NLP:

* Named Entity Recognition
* Sentiment Analysis
* Text Summarization
* Aspect Mining
* Topic Modeling

**8. What are the steps involved in preprocessing data for NLP?**

Here are some common pre-processing steps used in NLP software:

* **Preliminaries:** This includes word tokenization and sentence segmentation.
* **Common Steps:** Stop word removal, stemming and lemmatization, removing digits/punctuation, lowercasing, etc.
* **Processing Steps:** Code mixing, normalization, language detection, transliteration, etc.
* **Advanced Processing:** Parts of Speech (POS) tagging, coreference resolution, parsing, etc.

A blue and white diagram

Description automatically generated

**9. What do you mean by Stemming in NLP?**

When we remove the suffixes from a word so that the word is reduced to its base form, this process is called stemming. When the word is reduced to its base form, all the different variants of that word can be represented by the same form (e.g., “bird” and “birds” are both reduced to “bird”).

We can do this by using a fixed set of rules. For instance:  if a word ends in “-es,” we can remove the “-es”).

Even though these rules might not really make sense as a linguistically correct base form, stemming is usually carried out to match user queries in search engines to relevant documents. And in text classification, is done to reduce the feature space to train our machine learning (ML) models.

The code snippet given below depicts the way to use a well known NLP algorithm for stemming called Porter Stemmer using NLTK:

from nltk.stem.porter import PorterStemmer

stemmer = PorterStemmer()

word1, word2 = "bikes", "revolution"

print(stemmer.stem(word1), stemmer.stem(word2))

This gives “bike” as the stemmed version for “bikes,” but “revolut” as the stemmed form of “revolution,” even though the latter is not linguistically correct. Even if this might not affect the performance of the search engine, a derivation of the correct linguistic form becomes useful in some other cases. This can be done by another process that is closer to stemming, known as lemmatization.

**10. What do you mean by Lemmatization in NLP?**

The method of mapping all the various forms of a word to its base word (also called “lemma”) is known as Lemmatization. Although this may appear close to the definition of stemming, these are actually different. For instance, the word “better,” after stemming, remains the same. However, upon lemmatization, this should become “good,”. Lemmatization needs greater linguistic knowledge. Modelling and developing efficient lemmatizers still remains an open problem in NLP research.

The application of a lemmatizer based on WordNet from NLTK is shown in the code snippet below:

from nltk.stem import WordNetLemmatizer

lemmatizer = WordnetLemmatizer()

print(lemmatizer.lemmatize("better", pos="a")) #a is for adjective

**NLP Interview Questions for Experienced**

**1. What is the meaning of Text Normalization in NLP?**

Consider a situation in which we’re operating with a set of social media posts to find information events. Social media textual content may be very exceptional from the language we’d see in, say, newspapers. A phrase may be spelt in multiple ways, such as in shortened forms, (for instance, with and without hyphens), names are usually in lowercase, and so on. When we're developing NLP tools to work with such kinds of data, it’s beneficial to attain a canonical representation of textual content that captures these kinds of variations into one representation. This is referred to as text normalization.

Converting all text to lowercase or uppercase, converting digits to text (e.g., 7 to seven), expanding abbreviations, and so on are some frequent text normalisation stages.

**2. Explain the concept of Feature Engineering.**

After a variety of pre-processing procedures and their applications, we need a way to input the pre-processed text into an NLP algorithm later when we employ ML methods to complete our modelling step. The set of strategies that will achieve this goal is referred to as feature engineering. Feature extraction is another name for it. The purpose of feature engineering is to convert the text's qualities into a numeric vector that NLP algorithms can understand. This stage is called "text representation".

Discover your path to a  **Successful Tech Career For FREE!**

Answer 4 simple questions & get a career plan tailored for you

Interview Process

CTC & Designation

Projects on the Job

[**Try It Out**](https://www.scaler.com/career-plan?utm_source=ib&utm_medium=top_nudge&utm_campaign=CareerPlan)

2 Lakh+ Roadmaps Created

**3. What is an ensemble method in NLP?**

An ensemble approach is a methodology that derives an output or makes predictions by combining numerous independent similar or distinct models/weak learners. An ensemble can also be created by combining various models such as random forest, SVM, and logistic regression.

Bias, variance, and noise, as we all know, have a negative impact on the mistakes and predictions of any machine learning model. Ensemble approaches are employed to overcome these drawbacks.

**4. What do you mean by TF-IDF in Natural language Processing?**

TF-IDF also called **Term Frequency-Inverse Document Frequency** helps us get the importance of a particular word relative to other words in the corpus. It's a common scoring metric in information retrieval (IR) and summarization. TF-IDF converts words into vectors and adds semantic information, resulting in weighted unusual words that may be utilised in a variety of NLP applications.

**5. What are the steps to follow when building a text classification system?**

When creating a text classification system, the following steps are usually followed:

* Gather or develop a labelled dataset that is appropriate for the purpose.
* Decide on an evaluation metric after splitting the dataset into two (training and test) or three parts: training, validation (i.e., development), and test sets (s).
* Convert unprocessed text into feature vectors.
* Utilize the feature vectors and labels from the training set to train a classifier.
* Benchmark the model's performance on the test set using the evaluation metric(s) from Step 2.
* Deploy the model and track its performance to serve a real-world use case.

A diagram of a training process

Description automatically generated

**6. Explain how parsing is done in NLP.**

Parsing is the process of identifying and understanding a text's syntactic structure. It is accomplished by examining the text's constituent pieces. The machine parses each word one by one, then two by two, three by three, and so on. It's a unigram when the system parses the text one word at a time. A bigram is a text that is parsed two words at a time. When the machine parses three words at a time, the set of words is called a **trigram**.

The following points will help us comprehend the importance of parsing in NLP:

* Any syntax errors are reported by the parser.
* It aids in the recovery of often occurring errors so that the remainder of the programme can be processed.
* A parser is used to generate the parse tree.
* The parser is used to construct a symbol table, which is crucial in NLP.
* In addition, a Parser is utilised to generate intermediate representations (IR).

**7. What do you mean by a Bag of Words (BOW)?**

The **Bag of Words** model is a popular one that uses word frequency or occurrences to train a classifier. This methodology generates a matrix of occurrences for documents or phrases, regardless of their grammatical structure or word order.

A bag-of-words is a text representation that describes the frequency with which words appear in a document. It entails two steps:

* A list of terms that are well-known.
* A metric for determining the existence of well-known terms.

Because any information about the sequence or structure of words in the document is deleted, it is referred to as a "bag" of words. The model simply cares about whether or not recognised terms appear in the document, not where they appear.

**8. What do you mean by Parts of Speech (POS) tagging in NLP?**

A Part-Of-Speech Tagger (POS Tagger) reads the text in a language and assigns parts of speech to each word (and other tokens), such as noun, verb, adjective, and so on.

To label terms in text bodies, PoS taggers employ an algorithm. With tags like "noun-plural" or even more complicated labels, these taggers create more complex categories than those stated as basic PoS.

**9. What is Latent Semantic Indexing (LSI) in NLP?**

**Latent Semantic Indexing** (LSI), also known as Latent Semantic Analysis, is a mathematical method for improving the accuracy of information retrieval. It aids in the discovery of hidden(latent) relationships between words (semantics) by generating a set of various concepts associated with the terms of a phrase in order to increase information comprehension. Singular value decomposition is the NLP technique utilised for this aim. It's best for working with small groups of static documents.

**10. What is the difference between NLP and NLU?**

| **Natural Language Processing (NLP)** | **Natural Language Understanding (NLU)** |
| --- | --- |
| NLP is a system that manages end-to-end conversations between computers and people at the same time. | NLU aids in the solving of Artificial Intelligence's complex problems. |
| Humans and machines are both involved in NLP. | NLU allows machines to interpret unstructured inputs by transforming them into structured text. |
| NLP focuses on interpreting language in its most literal sense, such as what was said. | NLU, on the other hand, concentrates on extracting context and meaning, or what was meant. |
| NLP can parse text-based on grammar, structure, typography, and point of view. | It'll be NLU that helps the machine deduce the meaning behind the language content. |

**11. What are some metrics on which NLP models are evaluated?**

The following are some metrics on which NLP models are evaluated:

* **Accuracy:**When the output variable is categorical or discrete, accuracy is used. It is the percentage of correct predictions made by the model compared to the total number of predictions made.
* **Precision:** Indicates how precise or exact the model's predictions are, i.e., how many positive (the class we care about) examples can the model correctly identify given all of them?
* **Recall:** Precision and recall are complementary. It measures how effectively the model can recall the positive class, i.e., how many of the positive predictions it generates are correct.
* **F1 score:**This metric combines precision and recall into a single metric that also represents the trade-off between accuracy and recall, i.e., completeness and exactness.  
  (2 Precision Recall) / (Precision + Recall) is the formula for F1.
* **AUC:**As the prediction threshold is changed, the AUC captures the number of correct positive predictions versus the number of incorrect positive predictions.

**12. Explain the pipeline for Information extraction (IE) in NLP.**

In comparison to text classification, the typical pipeline for IE necessitates more fine-grained NLP processing. For example, we'd need to know the part-of-speech tags of words to identify named entities (people, organisations, etc.). We would require coreference resolution to connect various references to the same entity (e.g., Albert Einstein, Einstein, the scientist, he, etc.). It's worth noting that none of these stages are required for creating a text classification system. As a result, IE is a more NLP-intensive operation than text categorization. Not all steps in the pipeline are required for all IE jobs, as shown in the diagram, and the figure shows which IE tasks necessitate which degrees of analysis.

Other than named entity recognition, all other IE tasks require deeper NLP pre-processing followed by models developed for those specific tasks. Key phrase extraction is the task that requires the least amount of NLP processing (some algorithms also do POS tagging before extracting key phrases), whereas all other IE tasks require deeper NLP pre-processing followed by models developed for those specific tasks. Standard evaluation sets are often used to assess IE tasks in terms of precision, recall, and F1 scores. Because of the various levels of NLP pre-processing required, the accuracy of these processing steps has an impact on IE jobs. All of these factors should be considered when collecting relevant training data and, if necessary, training our own models for IE.

A diagram of a diagram

Description automatically generated

**13. What do you mean by Autoencoders?**

A network that is used for learning a vector representation of the input in a compressed form, is called an autoencoder. It is a type of unsupervised learning since labels aren’t needed for the process. This is mainly used to learn the mapping function from the input. In order to make the mapping useful, the input is reconstructed from the vector representation. After training is complete, the vector representation that we get helps encode the input text as a dense vector. Autoencoders are generally used to make feature representations.

In the figure below, the hidden layer depicts a compressed representation of the source data that captures its essence. The input representation is reconstructed by the output layer called the decoder.

A diagram of a network

Description automatically generated

**14. What do you mean by Masked language modelling?**

Masked language modelling is an NLP technique for extracting the output from a contaminated input. Learners can use this approach to master deep representations in downstream tasks. Using this NLP technique, you may predict a word based on the other words in the sentence.

The following is the process for Masked language modelling:

* Our text is tokenized. We start with text tokenization, just as we would with transformers.
* Make a tensor of labels. We're using a labels tensor to calculate loss against — and optimise towards — as we train our model.
* Tokens in input ids are masked. We can mask a random selection of tokens now that we've produced a duplicate of input ids for labels.
* Make a loss calculation. We use our model to process the input ids and labels tensors and determine the loss between them.

**15. What is the meaning of Pragmatic Analysis in NLP?**

Pragmatic Analysis is concerned with outside word knowledge, which refers to information that is not contained in the documents and/or questions. The many parts of the language that require real-world knowledge are derived from a pragmatics analysis that focuses on what was described and reinterpreted by what it truly meant.

**16. What is the meaning of N-gram in NLP?**

Text N-grams are commonly used in text mining and natural language processing. They're essentially a collection of co-occurring words within a specific frame, and when computing the n-grams, you usually advance one word (although you can move X words forward in more advanced scenarios).

**17. What do you mean by perplexity in NLP?**

It's a statistic for evaluating the effectiveness of language models. It is described mathematically as a function of the likelihood that the language model describes a test sample. The perplexity of a test sample X = x1, x2, x3,....,xn is given by,

PP(X)=P(x1,x2,…,xN)-1N

The total number of word tokens is N.

The more perplexing the situation, the less information the language model conveys.

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

[Naive Bayes algorithm](https://www.mygreatlearning.com/blog/introduction-to-naive-bayes/) 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.

**2. 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.

**3. What is text Summarization?**

[Text summarization](https://www.mygreatlearning.com/blog/text-summarization-in-python/) 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

**4. 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

**5. 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

**6. What is Bag of Words?**

[Bag of Words](https://www.mygreatlearning.com/blog/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.

**7. 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.

**8. 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.

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

The difference between NLP and CI is as follows:

| **Natural Language Processing (NLP)** | **Conversational Interface (CI)** |
| --- | --- |
| 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. |

**10. What are the best NLP Tools?**

Some of the best NLP tools from open sources are:

* SpaCy
* TextBlob
* Textacy
* Natural language Toolkit ([NLTK](https://www.mygreatlearning.com/blog/nltk-tutorial-with-python/))
* Retext
* NLP.js
* Stanford NLP
* CogcompNLP

**11. What is POS tagging?**

Parts of speech tagging better known as [POS tagging](https://www.mygreatlearning.com/blog/pos-tagging/) refer to the process of identifying specific words in a document and grouping 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 general process used for word mapping is quite ineffective for POS tagging because of the same reason.

**12. What is NES?**

Name entity recognition is more commonly known as NER is the process of identifying specific entities in a text document that are more informative and have a unique context. These often denote places, people, organizations, 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 categorize them under different predefined classes. This step further helps in extracting information.

**NLP Interview Questions for Experienced**

**13. 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.

**14. 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 versa.

E.g. cosine angle between two words “Football” and “Cricket” will be closer to 1 as compared to the 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.

**15. 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.

**16. 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?**

1. Keyword Normalization
2. Latent Semantic Indexing
3. Latent Dirichlet Allocation

a. only 1  
b. 2, 3  
c. 1, 3  
d. 1, 2, 3

**Answer:** d)

**17. 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)

**18. Dissimilarity between words expressed using cosine similarity will have values significantly higher than 0.5**

a. True  
b. False

**Answer:**a)

**19. Which one of the following is 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) is not keyword Normalization techniques. Named Entity helps 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.

**20. 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 the Speech use case.  
Only d) Text Summarization is an NLP use case.

**21. 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, the correct choice is Klog(3)/T

**22. 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)

**Answer:** b)

**23. 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.

**24. In NLP, The process of converting a sentence or paragraph into tokens is referred to as Stemming**

a. True  
b. False

**Answer:** b)

The statement describes the process of tokenization and not stemming, hence it is False.

**25. In NLP, Tokens are converted into numbers before giving to any Neural Network**

a. True  
b. False

**Answer:** a)

In NLP, all words are converted into a number before feeding to a Neural Network.

**26. Identify the odd one out**

a. nltk  
b. scikit learn  
c. SpaCy  
d. BERT

**Answer:** d)

All the ones mentioned are NLP libraries except BERT, which is a word embedding.

**27. TF-IDF helps you to establish?**

a. most frequently occurring word in document  
b. the most important word in the document

**Answer:** 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 is offset by the number of documents that appear in the corpus.

* TF is the frequency of terms divided by the 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. To understand more about NLP, check out these [NLP projects](https://www.mygreatlearning.com/academy/learn-for-free/courses/natural-language-processing-projects).

**28. 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

**Answer:** d)

**29. 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

**Answer:** 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.

**30. 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

**Answer:** 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.*

**31. In NLP, Words represented as vectors are called Neural Word Embeddings**

a. True  
b. False

**Answer:** a)

Word2Vec, GloVe based models build word embedding vectors that are multidimensional.

**32. In NLP, Context modeling is supported with which one of the following word embeddings**

1. a. Word2Vec
2. b) GloVe
3. c) BERT
4. d) All of the above

**Answer:** 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.

**33. In NLP, Bidirectional context is supported by which of the following embedding**

a. Word2Vec  
b. BERT  
c. GloVe  
d. All the above

**Answer:** 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.

**34. 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

**Answer:** 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.

**35. Word embeddings capture multiple dimensions of data and are represented as vectors**

a. True  
b. False

**Answer:** a)

**36. In NLP, Word embedding vectors help establish distance between two tokens**

a. True  
b. False

**Answer: a)**

**One can use Cosine similarity to establish the**distance between two vectors represented through Word Embeddings

**37. 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

**Answer:** a)

Statement b) is a bias as it buckets Woman into Homemaker, whereas statement a) is not a biased statement.

**38. 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

**Answer:**b)

Open AI’s GPT is able to learn complex patterns 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.

**39. Transformer architecture was first introduced with?**

a. GloVe  
b. BERT  
c. Open AI’s GPT  
d. ULMFit

**Answer:**c)

ULMFit has an LSTM based Language modeling architecture. This got replaced into Transformer architecture with Open AI’s GPT.

**40. Which of the following architecture can be trained faster and needs less amount of training data**

a. LSTM-based Language Modelling  
b. Transformer architecture

**Answer:** b)

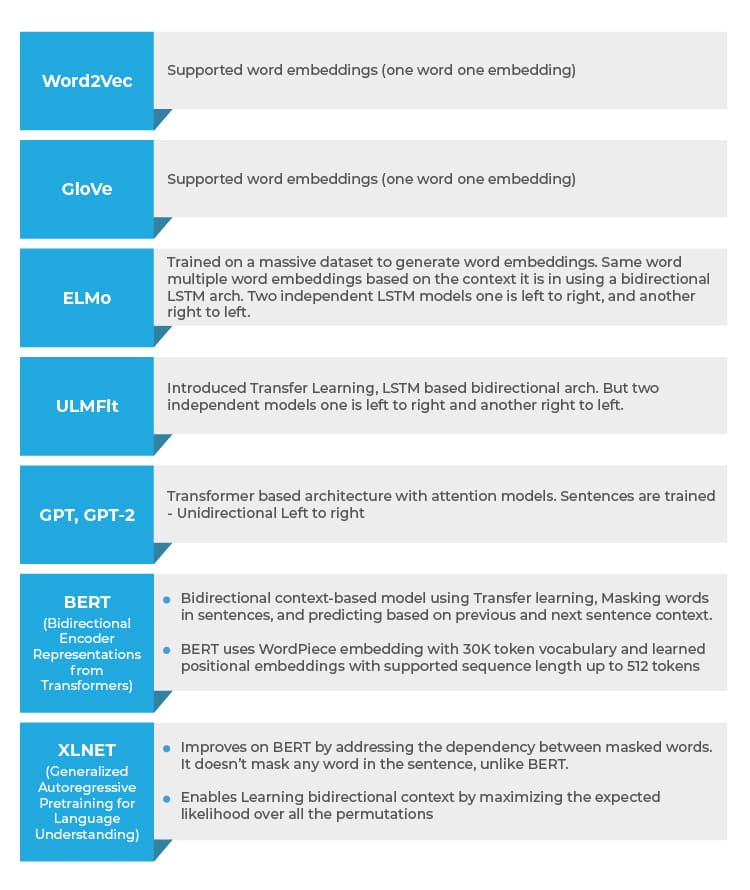
Transformer architectures were supported from GPT onwards and were faster to train and needed less amount of data for training too.

**41. Same word can have multiple word embeddings possible with \_\_\_\_\_\_\_\_\_\_\_\_?**

a. GloVe  
b. Word2Vec  
c. ELMo  
d. nltk

**Answer:**c)

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



**42. 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  
**Answer:** c)  
BERT uses token, segment and position embedding.

**43. 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  
**Answer:** 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.

**44. Uses unidirectional language model for producing word embedding.**

a. BERT  
b. GPT  
c. ELMo  
d. Word2Vec

**Answer:**b)

GPT is a bidirectional model and word embedding is produced by training on information flow from left to right. ELMo is bidirectional but shallow. Word2Vec provides simple word embedding.

**45. 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.

**46. 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

**47. 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.

**48. 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.

**49. 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.

**50. 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.

There, you have it – all the probable questions for your NLP interview. Now go, give it your best shot.

**Natural Language Processing FAQs**

**1. Why do we need NLP?**

One of the main reasons why NLP is necessary is because it helps computers communicate with humans in natural language. It also scales other language-related tasks. Because of NLP, it is possible for computers to hear speech, interpret this speech, measure it and also determine which parts of the speech are important.

**2. What must a natural language program decide?**

A natural language program must decide what to say and when to say something.

**3. Where can NLP be useful?**

NLP can be useful in communicating with humans in their own language. It helps improve the efficiency of the machine translation and is useful in emotional analysis too. It can be helpful in [sentiment analysis using python](https://www.mygreatlearning.com/academy/learn-for-free/courses/sentiment-analysis-using-python) too. It also helps in structuring highly unstructured data. It can be helpful in creating chatbots, Text Summarization and virtual assistants.

**4. How to prepare for an NLP Interview?**

The best way to prepare for an NLP Interview is to be clear about the basic concepts. Go through blogs that will help you cover all the key aspects and remember the important topics. Learn specifically for the interviews and be confident while answering all the questions.

**5. What are the main challenges of NLP?**

Breaking sentences into tokens, Parts of speech tagging, Understanding the context, Linking components of a created vocabulary, and Extracting semantic meaning are currently some of the main challenges of NLP.

**6. Which NLP model gives best accuracy?**

Naive Bayes Algorithm has the **highest accuracy** when it comes to NLP models. It gives up to 73% correct predictions.

**7. What are the major tasks of NLP?**

Translation, named entity recognition, relationship extraction, sentiment analysis, speech recognition, and topic segmentation are few of the major tasks of NLP. Under unstructured data, there can be a lot of untapped information that can help an organization grow.

**8. What are stop words in NLP?**

Common words that occur in sentences that add weight to the sentence are known as stop words. These stop words act as a bridge and ensure that sentences are grammatically correct. In simple terms, words that are filtered out before processing natural language data is known as a stop word and it is a common pre-processing method.

**9. What is stemming in NLP?**

The process of obtaining the root word from the given word is known as stemming. All tokens can be cut down to obtain the root word or the stem with the help of efficient and well-generalized rules. It is a rule-based process and is well-known for its simplicity.

**10. Why is NLP so hard?**

There are several factors that make the process of Natural Language Processing difficult. There are hundreds of natural languages all over the world, words can be ambiguous in their meaning, each natural language has a different script and syntax, the meaning of words can change depending on the context, and so the process of NLP can be difficult. If you choose to upskill and continue learning, the process will become easier over time.

**11. What does a NLP pipeline consist of \*?**

The overall architecture of an **NLP pipeline consists** of several layers: a user interface; one or several **NLP** models, depending on the use case; a Natural Language Understanding layer to describe the **meaning of** words and sentences; a preprocessing layer; microservices for linking the components together and of course.

# Scenario-based NLP Interview Questions

## **Q1. What is natural language processing (NLP), and how does it relate to artificial intelligence?**

This is one of the fundamental NLP Interview Questions that you must be aware of before proceeding to advanced level questions. [Natural Languagare Processing](https://www.shiksha.com/online-courses/articles/introduction-to-natural-language-processing/) is a subfield of [artificial intelligence](https://www.shiksha.com/online-courses/what-is-artificial-intelligence-st551-tg1405) that focuses on enabling computers to understand and analyze human language. It involves the use of computational techniques to process and analyze large amounts of natural language data, such as text or speech. The ultimate goal of NLP is to enable computers to interact with humans in a natural and intuitive way, as if they were communicating with another person.

NLP is closely related to artificial intelligence because it relies on many of the same techniques and approaches used in other AI fields, such as [machine learning](https://www.shiksha.com/online-courses/what-is-machine-learning-st553) and [deep learning](https://www.shiksha.com/online-courses/what-is-deep-learning-st551). By leveraging these techniques, NLP researchers and practitioners are able to develop sophisticated [algorithms](https://www.shiksha.com/online-courses/articles/algorithms/) and models that can process and analyze natural language data with high accuracy and efficiency.

## **Q2. Can you explain the difference between syntax and semantics in NLP?**

In NLP, syntax refers to the structure of language, such as the rules and patterns that govern how words are combined to form sentences. This might include things like sentence structure, word order, and grammatical rules. Syntax is important because it provides a framework for understanding the meaning of sentences and the relationships between different words and phrases.

Semantics, on the other hand, refers to the meaning of language, such as the concepts and ideas that are conveyed by words and sentences. This might include things like word definitions, contextual meaning, and inferred meaning. Semantics is important because it allows us to understand the deeper meaning behind language and interpret the nuances and subtleties of human communication.

## **Q3. How would you approach developing a machine learning model for a named entity recognition (NER) task?**

Named entity recognition (NER) is a common NLP task that involves identifying and categorizing specific types of entities, such as people, organizations, and locations, in a piece of text. To develop a machine learning model for a NER task, we would first gather a large dataset of labeled examples, where each example contains a piece of text and the associated named entities.

Next, we would preprocess the text data by performing [tokenization](https://www.shiksha.com/online-courses/articles/tokenization-in-nlp/), [normalization](https://www.shiksha.com/online-courses/articles/normalization-and-standardization/), and other techniques to create a standardized format for the text. Then, we would choose a suitable [machine learning algorithm](https://www.shiksha.com/online-courses/articles/top-machine-learning-algorithms-for-beginners/), such as a conditional random field (CRF) or a recurrent neural network (RNN), and train the model on the labeled dataset.

During training, we would tune the hyperparameters of the model to optimize its performance on the validation set. Finally, we would evaluate the model on a test set to ensure that it is accurate and effective at identifying named entities in text.

## **Q4. What are some common challenges or limitations of NLP, and how do you overcome them?**

One common challenge in NLP is dealing with language ambiguity, which occurs when a word or phrase can have multiple possible meanings depending on the context. To overcome this challenge, we would leverage techniques like part-of-speech tagging, syntactic parsing, and named entity recognition to better understand the structure and meaning of the text.

Another challenge is dealing with out-of-vocabulary (OOV) words, which are words that the NLP model has not seen before and does not have a representation for. To overcome this challenge, we might use techniques like subword tokenization or character-level representations to enable the model to handle previously unseen words.

A final challenge is dealing with bias in the data or the model itself. To overcome this challenge, we would carefully analyze the data and the model to identify any biases or limitations, and take steps to mitigate or correct them. This might involve using techniques like adversarial training, data augmentation, or algorithmic fairness to ensure that the NLP model is accurate, reliable, and fair.

## **Q5. How would you approach designing a chatbot for a customer service use case?**

When designing a chatbot for customer service, the first step is to understand the customer’s needs and the types of queries they are likely to have. From there, we would develop a list of potential user intents and map them to appropriate responses. This would involve creating a set of rules or a decision tree to guide the chatbot’s behavior.

Next, we would choose a natural language processing (NLP) framework that can handle the specific requirements of the customer service use case. This might involve training the chatbot on a large corpus of customer interactions to help it recognize common phrases and language patterns.

Finally, we would test the chatbot extensively to ensure that it can handle a range of scenarios and provide accurate and helpful responses to users.

## **Q6. How would you use machine learning to extract meaningful insights from unstructured text data?**

To extract meaningful insights from unstructured text data using machine learning, we would first preprocess the data to remove any noise and standardize the format of the text. Then, we would use techniques like tokenization and lemmatization to break the text down into smaller, more manageable units.

From there, we would choose an appropriate machine learning algorithm such as a [clustering](https://www.shiksha.com/online-courses/articles/types-of-clustering-algorithm-scenario-you-must-know-as-a-data-scientist/) or classification algorithm to group similar pieces of text together or categorize them into different topics. we would train the algorithm on a labeled dataset and optimize its hyperparameters to achieve the best performance.

Finally, we would analyze the output of the algorithm to identify patterns and trends in the data and use these insights to make data-driven decisions.

## **Q7. Can you explain the difference between supervised and unsupervised learning in NLP?**

In [supervised learning](https://www.shiksha.com/online-courses/articles/supervised-learning-real-life-analogy-and-applications/), the machine learning model is trained on a labeled dataset, where each data point is associated with a predefined label or output. The goal of the model is to learn a mapping between the input features and the corresponding output labels, so that it can accurately predict the output for new, unseen data points.

In contrast, [unsupervised learning](https://www.shiksha.com/online-courses/articles/introduction-to-unsupervised-learning/) does not rely on labeled data. Instead, the model is trained on a dataset of unstructured or unlabeled data and is tasked with finding patterns or structure within the data itself. This might involve techniques like clustering or dimensionality reduction to group similar data points together or represent them in a lower-dimensional space.

In NLP, supervised learning is commonly used for tasks like [sentiment analysis](https://www.shiksha.com/online-courses/articles/learn-all-about-sentiment-analysis/) or named entity recognition, where the output label is already defined. Unsupervised learning is often used for tasks like topic modeling or word embeddings, where the goal is to identify underlying patterns in the data without any predefined labels.

## **Q8. How would you handle a situation where a text classification model is misclassifying a significant number of documents?**

This is one of the important scenario based NLP Interview Questions. If a text classification model is misclassifying a significant number of documents, there are several steps we would take to diagnose and address the issue.

First, we would evaluate the performance of the model on a validation set and calculate metrics like [precision and recall](https://www.shiksha.com/online-courses/articles/precision-and-recall/), and F1 score to determine which classes are being misclassified the most.

Next, we would examine the misclassified documents themselves to see if there are any patterns or commonalities that might explain why the model is struggling to classify them correctly. This might involve manually annotating the documents with the correct labels and retraining the model on the updated dataset.

I might also try experimenting with different feature representations or tweaking the hyperparameters of the model to see if that improves performance.

## **Q9. How would you go about developing a system for sentiment analysis of social media posts?**

To develop a system for sentiment analysis of social media posts, we would first gather a large dataset of social media posts and their associated sentiment labels (e.g., positive, negative, neutral).

Next, we would preprocess the text by removing any noise, such as URLs or emojis, and performing tokenization and normalization to create a standardized format for the text.

Then, we would choose a suitable NLP framework or library, such as spaCy or NLTK, to perform sentiment analysis on the text data. This might involve training a machine learning model using a supervised learning approach, where the model is trained on the labeled dataset to predict the sentiment of new, unseen social media posts.

Alternatively, we might use a pre-trained language model, such as BERT or [GPT](https://www.shiksha.com/online-courses/articles/how-to-use-chatgpt/), which has already been fine-tuned for sentiment analysis tasks. These models are trained on large amounts of data and can often provide state-of-the-art performance on sentiment analysis tasks.

Finally, we would test and evaluate the system on a holdout dataset to ensure that it is accurate and can generalize well to new, unseen data. we would also monitor the performance of the system over time and retrain or update it as needed to ensure that it remains effective in detecting sentiment in social media posts.

## **Q10. How would you go about evaluating the performance of an NLP model?**

To evaluate the performance of an NLP model, we would first define the evaluation metric or metrics that are most relevant to the task at hand. For example, if we are developing a sentiment analysis model, we might use metrics like accuracy, precision, recall, and F1 score to measure its performance.

Next, we would split my dataset into a training set, a validation set, and a test set. we would use the training set to train the model, the validation set to tune its hyperparameters and optimize its performance, and the test set to evaluate its final performance.

During evaluation, we would measure the model’s performance on the test set using the chosen evaluation metrics. we would also analyze the model’s outputs and errors to gain insights into its strengths and weaknesses, and to identify areas for improvement.

Finally, we would compare the performance of my NLP model to the performance of other state-of-the-art models or benchmarks in the field, to assess its relative effectiveness and identify opportunities for future improvement.

**4. What is the difference between a formal language and a natural language?**

|  |  |
| --- | --- |
| Formal Language | Natural Language |
| A formal language is a collection of strings, where each string contains symbols from a finite set called alphabets. | A natural language is a language that humans utilize to speak. It is usually a lot different from a formal language. These typically contain fragments of words and pause words like uh, um, etc. |

**5. What is the difference between stemming and lemmatization?**

Both stemming and lemmatization are keyword normalization techniques aiming to minimize the morphological variation in the words they encounter in a sentence. But, they are different from each other in the following way.

|  |  |
| --- | --- |
| Stemming | Lemmatization |
| This technique involves removing the affixes added to a word and leaving us with the rest of the word. | Lemmatization is the process of converting a word into its lemma from its inflected form. |
| Example: ‘Caring’→ ’Car’ | Example: ‘Caring’→ ’Care’ |

**6. What is NLU?**

 NLU stands for Natural Language Understanding. It is a subdomain of NLP that concerns making a machine learn the skills of reading comprehension. A few applications of NLU include Machine translation (MT), Newsgathering, and Text categorization. It often goes by the name Natural Language Interpretation (NLI) as well.

**Get Closer To Your Dream of Becoming a Data Scientist with 70+ Solved**[**End-to-End ML Projects**](https://www.projectpro.io/projects?utm_source=Blg439&utm_medium=ConvertLink&utm_campaign=TXTCTA1)

**7. List the differences between NLP and NLU.**

|  |  |
| --- | --- |
| Natural Language Processing | Natural Language Understanding |
| NLP is a branch of AI that deals with designing programs for machines that will allow them to process the language that humans use. The idea is to make machines imitate the way humans utilize language for communication. | In NLU, the aim is to improve a computer’s ability to understand and analyze human language. This aim is achieved by transforming unstructured data into a machine-readable format. |

**8. What do you know about Latent Semantic Indexing (LSI)?**

LSI is a technique that analyzes a set of documents to find the statistical coexistence of words that appear together. It gives an insight into the topics of those documents.

LSI is also known as Latent Semantic Analysis.

**9. List a few methods for**[**extracting features**](https://www.projectpro.io/recipes/load-features-from-dictionary-in-python)**from a corpus for NLP.**

1. Bag-of-Words

2. Word Embedding

**10. What are stop words?**

Stop words are the words in a document that are considered redundant by NLP engineers and are thus removed from the document before processing it. Few examples are ‘is’, ‘the’, ‘are, ‘am’.

**11. What do you know about Dependency Parsing?**

Dependency parsing is a technique that highlights the dependencies among the words of a sentence to understand its grammatical structure. It examines how the words of a sentence are linguistically linked to each other. These links are called dependencies.

**12. What is Text Summarization? Name its two types.**

[Text Summarization](https://www.projectpro.io/article/text-summarization-projects/693) is a method of converting a long-form text into a summary. The summary thus generated is expected to have critical ideas of the lengthy text. Two main types of Text Summarization are:

1. Extraction-based Summarization
2. Abstraction-based Summarization

**Get FREE Access to**[**Machine Learning Example Codes**](https://www.projectpro.io/recipes?utm_source=Blg439&utm_medium=RcpLink&utm_campaign=TXTCTA2)**for Data Cleaning, Data Munging, and Data Visualization**

**13. What are false positives and false negatives?**

If a machine learning algorithm falsely predicts a negative outcome as positive, then the result is labeled as a false negative.

And, if a machine learning algorithm falsely predicts a positive outcome as negative, then the result is labeled as a false positive.

**14. List a few methods for part-of-speech tagging.**

Rule-based tagging, HMM-tagging, transformation-based tagging, and memory-based tagging.

**15. What is a corpus?**

‘Corpus’ is a Latin word that means ‘body.’ Thus, a body of the written or spoken text is called a corpus.

**Recommended Reading:**[**10 NLP Techniques Every Data Scientist Should Know**](https://www.projectpro.io/article/10-nlp-techniques-every-data-scientist-should-know/415)

## **NLP Algorithm Interview Questions with Answers**

Most recruiters usually try to understand how well you know the models that are used widely in NLP. Take a look at these interview questions in NLP with answers that will help you upgrade your NLP algorithm skills.

**1. List a few real-world applications of the n-gram model.**

1. Augmentive Communication

2. Part-of-speech Tagging

3. Natural language generation

4. Word Similarity

5. Authorship Identification

6. Sentiment Extraction

7. Predictive Text Input

**2. What does TF\*IDF stand for? Explain its significance.**

TF\*IDF stands for Term-Frequency/Inverse-Document Frequency. It is an information-retrieval measure that encapsulates the semantic significance of a word in a particular document N, by degrading words that tend to appear in a variety of different documents in some huge background corpus with D documents.

Let nw denote the frequency of a word w in the document N, m represents the total number of documents in the corpus that contain w. Then, TF\*IDF is defined as

𝑇𝐹\*𝐼𝐷𝐹(𝑤)=𝑛𝑤×log𝑛𝑚

**3. What is perplexity in NLP?**

It is a metric that is used to test the performance of language models. Mathematically, it is defined as a function of the probability that the language model represents a test sample. For a test sample X = x1, x2, x3,....,xn, the perplexity is given by,

𝑃𝑃(𝑋)=𝑃(𝑥1,𝑥2,…,𝑥𝑁)-1𝑁

where N is the total number of word tokens.

Higher the perplexity, lesser is the information conveyed by the language model.

**4. Which algorithm in NLP supports bidirectional context?**

 BERT

**5. What is the Naive Bayes algorithm?**

Naive Bayes is a [classification machine learning algorithm](https://www.projectpro.io/article/7-types-of-classification-algorithms-in-machine-learning/435) that utilizes Baye’s Theorem for labeling a class to the input set of features. A vital element of this algorithm is that it assumes that all the feature values are independent.

Table of Contents

* [The Most Common Use-cases of NLP are:](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#The_Most_Common_Use-cases_of_NLP_are)
  + [Let Us Look At The Most Common NLP Terms](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#Let_Us_Look_At_The_Most_Common_NLP_Terms)
* [NLP Interview Questions With Answers](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#NLP_Interview_Questions_With_Answers)
  + [1. What is the NLG (Natural Language Generation)?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#1_What_is_the_NLG_Natural_Language_Generation)
  + [2. What is the order of steps in natural language understanding?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#2_What_is_the_order_of_steps_in_natural_language_understanding)
  + [3. What is signal processing in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#3_What_is_signal_processing_in_NLP)
  + [4. What is pragmatic analysis in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#4_What_is_pragmatic_analysis_in_NLP)
  + [5. What is syntactic analysis in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#5_What_is_syntactic_analysis_in_NLP)
  + [6. What is semantic analysis in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#6_What_is_semantic_analysis_in_NLP)
  + [7. What is sentiment analysis in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#7_What_is_sentiment_analysis_in_NLP)
  + [8. What is discourse analysis in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#8_What_is_discourse_analysis_in_NLP)
  + [9. What is pragmatic ambiguity in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#9_What_is_pragmatic_ambiguity_in_NLP)
  + [10. What are the major applications of NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#10_What_are_the_major_applications_of_NLP)
  + [11. List any real-world application of NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#11_List_any_real-world_application_of_NLP)
  + [12. What are the common NLP techniques?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#12_What_are_the_common_NLP_techniques)
  + [What are the components of NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#What_are_the_components_of_NLP)
  + [14. What are the tools used for training NLP models?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#14_What_are_the_tools_used_for_training_NLP_models)
  + [15. Which NLP technique uses a lexical knowledge base to obtain the correct base form of the words?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#15_Which_NLP_technique_uses_a_lexical_knowledge_base_to_obtain_the_correct_base_form_of_the_words)
  + [16. List the models to reduce the dimensionality of data in NLP.](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#16_List_the_models_to_reduce_the_dimensionality_of_data_in_NLP)
  + [17. List some open-source libraries for NLP.](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#17_List_some_open-source_libraries_for_NLP)
  + [18. Explain the masked language model.](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#18_Explain_the_masked_language_model)
  + [19. What is the bag of words model?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#19_What_is_the_bag_of_words_model)
  + [20. What is CBOW in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#20_What_is_CBOW_in_NLP)
  + [21. What is TF-IDF and what are its uses?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#21_What_is_TF-IDF_and_what_are_its_uses)
  + [22. What are POS and tagging?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#22_What_are_POS_and_tagging)
  + [23. What is n-gram in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#23_What_is_n-gram_in_NLP)
  + [24. What is skip-gram?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#24_What_is_skip-gram)
  + [25. What is the corpus in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#25_What_is_the_corpus_in_NLP)
  + [26. What are the features of the text corpus in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#26_What_are_the_features_of_the_text_corpus_in_NLP)
  + [27. What is normalization in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#27_What_is_normalization_in_NLP)
  + [28. What is keyword normalization?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#28_What_is_keyword_normalization)
  + [29. What is lemmatization in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#29_What_is_lemmatization_in_NLP)
  + [30. What is stemming in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#30_What_is_stemming_in_NLP)
  + [31. What is ambiguity in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#31_What_is_ambiguity_in_NLP)
  + [32. What is tokenization in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#32_What_is_tokenization_in_NLP)
  + [33. What are stop words in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#33_What_are_stop_words_in_NLP)
  + [34. How to find word similarity in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#34_How_to_find_word_similarity_in_NLP)
  + [35. How to find sentence similarity in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#35_How_to_find_sentence_similarity_in_NLP)
  + [36. How to find document similarity in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#36_How_to_find_document_similarity_in_NLP)
  + [37. What are transformers?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#37_What_are_transformers)
  + [38. What are punctuations in NLP, and how can we remove them?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#38_What_are_punctuations_in_NLP_and_how_can_we_remove_them)
  + [39. What is latent semantic indexing (LSI)?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#39_What_is_latent_semantic_indexing_LSI)
  + [40. What is named entity recognition (NER)?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#40_What_is_named_entity_recognition_NER)
  + [41. What is NLTK in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#41_What_is_NLTK_in_NLP)
  + [42. What is spaCY?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#42_What_is_spaCY)
  + [43. What is openNLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#43_What_is_openNLP)
  + [44. What is the difference between NLTK and openNLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#44_What_is_the_difference_between_NLTK_and_openNLP)
  + [45. What is parsing?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#45_What_is_parsing)
  + [46. What is dependency parsing?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#46_What_is_dependency_parsing)
* [47. What is semantic parsing?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#47_What_is_semantic_parsing)
  + [48. What is constituency parsing?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#48_What_is_constituency_parsing)
  + [49. What is shallow parsing?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#49_What_is_shallow_parsing)
  + [50. What are the differences between dependency parsing and shallow parsing?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#50_What_are_the_differences_between_dependency_parsing_and_shallow_parsing)
  + [51. What is language modeling?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#51_What_is_language_modeling)
  + [52. What is topic modeling?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#52_What_is_topic_modeling)
  + [53. What is text summarization in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#53_What_is_text_summarization_in_NLP)
  + [54. What is the difference between a regular expression and regular grammar?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#54_What_is_the_difference_between_a_regular_expression_and_regular_grammar)
  + [55. What is perplexity in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#55_What_is_perplexity_in_NLP)
  + [56. What is the Naive Bayes algorithm, and where is it used in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#56_What_is_the_Naive_Bayes_algorithm_and_where_is_it_used_in_NLP)
  + [57. What is the PageRank algorithm?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#57_What_is_the_PageRank_algorithm)
  + [58. What is noise removal?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#58_What_is_noise_removal)
  + [59. What is word embedding?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#59_What_is_word_embedding)
  + [60. What are the word embedding libraries?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#60_What_are_the_word_embedding_libraries)
  + [61. What is word2vec?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#61_What_is_word2vec)
  + [61. What is doc2vec?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#61_What_is_doc2vec)
  + [63. What is a document-term matrix?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#63_What_is_a_document-term_matrix)
  + [64. What is wordnet?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#64_What_is_wordnet)
  + [65. What is GloVe in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#65_What_is_GloVe_in_NLP)
  + [66. What is a flexible string matching?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#66_What_is_a_flexible_string_matching)
  + [67. What is cosine similarity?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#67_What_is_cosine_similarity)
  + [68. What is information extraction?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#68_What_is_information_extraction)
  + [69. What is object standardization, and when is it used?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#69_What_is_object_standardization_and_when_is_it_used)
  + [71. What is text generation, and when is it done?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#71_What_is_text_generation_and_when_is_it_done)
  + [71. How can we estimate the entropy of the English language?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#71_How_can_we_estimate_the_entropy_of_the_English_language)
  + [72. What is Latent Dirichlet Allocation?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#72_What_is_Latent_Dirichlet_Allocation)
  + [73. What are the conditional random fields?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#73_What_are_the_conditional_random_fields)
  + [74. What are the hidden Markov random fields?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#74_What_are_the_hidden_Markov_random_fields)
  + [75. What is a coreference resolution?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#75_What_is_a_coreference_resolution)
  + [76. What is PAC learning?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#76_What_is_PAC_learning)
  + [77. What is sequence learning?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#77_What_is_sequence_learning)
  + [78. What is an ensemble method?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#78_What_is_an_ensemble_method)
* [Tips To Crack An NLP Interview](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#Tips_To_Crack_An_NLP_Interview)

## **The Most Common Use-cases of NLP are:**

1. Sentiment Analysis
2. Text Summarization
3. Content Categorization
4. Speech-to-Text conversion and vice-versa
5. Machine Translation

## **Let Us Look At The Most Common NLP Terms**

* **Vocabulary:** The group of terms used in a text or speech.
* **Corpus, or Corpora(Plural):** It is a collection of text of similar type, for example, movie review, social media posts, etc.
* **Documents:** They are the body of text and collectively form a corpus.
* **Out of Vocabulary:** Terms that are not included in the vocabulary that we created during our model’s training are included in this category.
* **Preprocessing:** It is a method that attempts to remove unwanted text, or noise, from the given text and to make it “clean.” It is the first step of any NLP task. You may also like to read about [*Data Preprocessing in Machine Learning*](https://www.analytixlabs.co.in/blog/data-preprocessing-in-machine-learning/).
* **Tokenization:** **Tokenization in NLP**breaks down the large sets of text into small parts for easy readability and understanding. Each small part is referred to as ‘text’ and provides a piece of meaningful information.
* **Embeddings (Word):** It is the process of embedding each token as a vector before passing it into a machine learning model. Embeddings can also be done on phrases and characters as well, apart from words.
* **N-grams**: It is a continuous sequence (similar to the power set in number theory) of n-tokens of a given text.
* **Transformers:** They are deep learning architectures that can have the ability to parallelize computations. Transformers are used to learn long term dependencies.
* **Parts of Speech (POS):** They are the word’s functions, like a noun, verb, etc.
* **Parts of Speech Tagging:** It is the process of tagging words in the sentences into different parts of speech.
* **Stop Words:** It is the removal of unwanted text from further processing of text, for example, a, to, can, etc.
* **Normalization:** It is the process of mapping similar terms to a canonical form, i.e., a single entity.
* **Lemmatization:** Lemmatization in NLP is a type of normalization used to group similar terms to their base form based on the parts of speech. For example, talking and talking can be mapped to a single term, walk.
* **Stemming:** **Stemming in NLP** is also a type of normalization and is similar to lemmatization, but the difference here is that it segregates the words without the parts of speech tags. It is faster than lemmatization and can also be more accurate in some cases.

## **NLP Interview Questions** With Answers

In this section, let us see the extensive set of NLP interview questions. **Following the question links, you can also find answers to these NLP interview questions.**

1. [What is the NLG (Natural Language Generation)?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q1)
2. [What is the order of steps in natural language understanding?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q2)
3. [What is signal processing in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q3)
4. [What is pragmatic analysis in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q4)
5. [What is syntactic analysis in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q5)
6. [What is semantic analysis in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q6)
7. [What is sentiment analysis in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q7)
8. [What is discourse analysis in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q8)
9. [What is pragmatic ambiguity in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q9)
10. [What are the major applications of NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q10)
11. [List any real-world application of NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q11)
12. [What are the commonNLP techniques?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q12)
13. [What are the components of NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q13)
14. [What are the tools used for training NLP models?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q14)
15. [Which NLP technique uses a lexical knowledge base to obtain the correct base form of the words?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q15)
16. [List the models to reduce the dimensionality of data in NLP.](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q16)
17. [List some open-source libraries for NLP.](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q17)
18. [Explain the masked language model.](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q18)
19. [What is the bagofwords (bag of words) model?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q19)
20. [What is CBOW in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q20)
21. [What is TF-IDF, and what are its uses?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q21)
22. [What are POS and tagging?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q22)
23. [What is n-gram in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q23)
24. [What is skip-gram?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q24)
25. [What is the corpus in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q25)
26. [What are the features of the text corpus in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q26)
27. [What is normalization in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q27)
28. [What is keyword normalization?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q28)
29. [What is lemmatization in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q29)
30. [What is stemming in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q30)
31. [What is ambiguity in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q31)
32. [What istokenization in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q32)
33. [What are stop words in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q33)
34. [How to find word similarity in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q34)
35. [How to find sentence similarity in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q35)
36. [How to find document similarity in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q36)
37. [What are transformers?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q37)
38. [What are punctuations in NLP, and how can we remove them?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q38)
39. [What is latent semantic indexing (LSI)?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q39)
40. [What is named entity recognition (NER)?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q40)
41. [What is NLTK in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q41)
42. [What is spaCY?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q42)
43. [What is openNLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q43)
44. [What is the difference between NLTK and openNLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q44)
45. [What is parsing?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q45)
46. [What is dependency parsing?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q46)
47. [What is semantic parsing?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q47)
48. [What is constituency parsing?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q48)
49. [What is shallow parsing?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q49)
50. [What are the differences between dependency parsing and shallow parsing?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q50)
51. [What is language modeling?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q51)
52. [What is topic modeling?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q52)
53. [What is text summarization in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q53)
54. [What is the difference between a regular expression (RegEx) and regular grammar?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q54)
55. [What is perplexity in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q55)
56. [What is the Naive Bayes algorithm, and where is it used in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q56)
57. [What is the PageRank algorithm?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q57)
58. [What is noise removal?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q58)
59. [What is word embedding?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q59)
60. [What are the word embedding libraries?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q60)
61. [What is word2vec?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q61)
62. [What is doc2vec?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q62)
63. [What is a document-term matrix?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q63)
64. [What is wordnet?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q64)
65. [What is GloVe in NLP?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q65)
66. [What is a flexible string matching?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q66)
67. [What is cosine similarity?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q67)
68. [What is information extraction?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q68)
69. [What is object standardization, and when is it used?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q69)
70. [What is text generation, and when is it done?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q70)
71. [How can we estimate the entropy of the English language?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q71)
72. [What is Latent Dirichlet Allocation?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q72)
73. [What are the conditional random fields?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q73)
74. [What are the hidden Markov random fields?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q74)
75. [What is a coreference resolution?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q75)
76. [What is PAC learning?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q76)
77. [What is sequence learning?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q77)
78. [What is an ensemble method?](https://www.analytixlabs.co.in/blog/nlp-interview-questions/#q78)

A diagram of a brain

Description automatically generated

## **1.** What is the NLG (Natural Language Generation)?

Natural Language Generation is a part of AI and generates natural language texts from structured data to produce an output. It can be seen as NLP’s reverse process, where NLP is used to understand and interpret the natural language to form data, and NLU is used to generate outputs in natural language from structured data.

## **2. What is the order of steps in natural language understanding?**

The order of steps that are to be followed in Natural Language Understanding is as follows:

1. Signal Processing
2. Syntactic Analysis
3. Semantic Analysis
4. Pragmatics

## **3. What is signal processing in NLP?**

 Signal processing is a method that enables software to analyze, modify, and synthesize signals. In NLP, these can be sound or text signals.

## **4. What is pragmatic analysis in NLP?**

The pragmatic analysis is the process of information extraction from the given text. It is a set of linguistic and logical tools that enable us to churn out the meaning of the given structure of a text.

## **5. What is syntactic analysis in NLP?**

  The syntactic analysis, also referred to as parsing and syntax analysis, is the phase in which we try to process the given text’s structure. This process tries to draw meaning from the text by comparing it to formal grammar rules or syntax.

## **6. What is semantic analysis in NLP?**

The semantic analysis is the process of understanding the meaning of the text in the way humans perceive and communicate. It focuses on larger parts of data for processing, as compared to other analysis techniques.

## **7. What is sentiment analysis in NLP?**

The sentiment analysis, also known as opinion mining and emotion AI, is a process of detecting the polarity of the opinion in the text or can be a part of it. It is majorly used to identify, extract, and quantify user or customer reviews’ polarity, survey responses, or social media opinions.

## **8. What is discourse analysis in NLP?**

Discourse is a structured group of the sentence. Discourse analysis can be termed as an approach to analyzing the discourse, i.e., text or language. It involves text interpretations and interactions.

## **9. What is pragmatic ambiguity in NLP?**

Pragmatic ambiguity can be referred to as a condition where words have multiple interpretations. This condition arises when the meaning of words is not specific; i.e., it can give different meanings.

## **10. What are the major applications of NLP?**

 The major applications of NLP are:

1. Machine Translation
2. Speech Recognition
3. Sentiment Analysis
4. Text Classification

Several different colored speech bubbles

Description automatically generated

## **11. List any real-world application of NLP?**

 The most used real-world application of NLP is speech recognition. Examples of speech recognition applications are Amazon Alexa, Google Assistant, Siri, HP Cortana.

## **12. What are the common NLP techniques?**

 The common NLP techniques for text extraction are:

1. Named Entity Recognition
2. Sentiment Analysis
3. Text Summarization
4. Aspect Mining
5. Text Modelling

## **What are the components of NLP?**

  The components of NLP are:

1. Lexical Analysis
2. Syntactic Analysis
3. Semantic Analysis
4. Discourse Integration
5. Pragmatic Analysis

## **14. What are the tools used for training NLP models?**

The tools used to train NLP models are NLTK, spaCY, PyTorch-NLP, openNLP.

## **15. Which NLP technique uses a lexical knowledge base to obtain the correct base form of the words?**

The **NLP techniques**that use lexical knowledge to obtain the correct base form are lemmatization and stemming.

## **16. List the models to reduce the dimensionality of data in NLP.**

 The commonly used models are TF-IDF, Word2vec/Glove, LSI, Topic Modelling, Elmo Embeddings.

## **17. List some open-source libraries for NLP.**

            The popular libraries are NLTK (Natural Language ToolKit), SciKit Learn, Textblob, CoreNLP, spaCY, Gensim.

A group of logos of companies

Description automatically generated

## **18. Explain the masked language model.**

Masked modeling is an example of autoencoding language modeling. Here the output is predicted from corrupted input. By this model, we can predict the word from other words present in the sentences.

## **19. What is the bag of words model?**

 The Bagofwords model is used for information retrieval. Here the text is represented as a multiset, i.e., a bag of words. We don’t consider grammar and word order, but we surely maintain the multiplicity.

A diagram of a bag with words

Description automatically generated

## **20. What is CBOW in NLP?**

CBOW or continuous bag of words is a model that tries to predict the target word from the available source context words, i.e., the surrounding words. Here the context words are taken into account as multiple words for a given target word.

## **21. What is TF-IDF and what are its uses?**

 TF-IDF is an abbreviation for the term frequency-inverse documentary frequency. It is used to provide a numeric value to a word to show how important it is in the document or a corpus.

## **22. What are POS and tagging?**

 Parts Of Speech (POS) are the functions of the word, like a noun, verb, etc., and tagging is labeling the words present in the sentences into different parts of speech.

## **23. What is n-gram in NLP?**

   N-grams are the continuous sequence (similar to the power set in number theory) of n-tokens of a given text.

## **24. What is skip-gram?**

 Skip gram is an unsupervised learning technique used to find the most related words to a target word. It is a reverse process of the continuous bag of words model.

## **25. What is the corpus in NLP?**

Corpus or corpora (plural), is a collection of the text of a similar type, for example, movie reviews, social media posts, etc.

## **26. What are the features of the text corpus in NLP?**

  The features of text corpus are:

1. Word count
2. Vector notation
3. Part of speech tag
4. Boolean feature
5. Dependency grammar

## **27. What is normalization in NLP?**

Normalization is the process of mapping similar terms to a canonical form, i.e., a single entity.

## **28. What is keyword normalization?**

  Keyword normalization is an **NLP technique**where we apply normalization on a word to condense it to its most basic form.

## **29. What is lemmatization in NLP?**

 Lemmatization is a type of normalization used to group similar terms to their base form-based on the parts of speech. For example, talking and talking can be mapped to a single term, walk.

## **30. What is stemming in NLP?**

Stemming in NLP is also a type of normalization and is similar to lemmatization, but the difference here is that it segregates the words without the parts of speech tags. It is faster than lemmatization and can also be more accurate in some cases.

## **31. What is ambiguity in NLP?**

 Ambiguity can be referred to as a condition when a word can have multiple interpretations and results in being misunderstood. Natural languages are ambiguous and can make it difficult to process**NLP techniques** on them, resulting in the wrong output.

## **32. What is tokenization in NLP?**

Tokenization is the process of breaking down large sets of text into small parts for easy readability and understanding. Each small part is referred to as ‘text’ and provides a piece of meaningful information.

## **33. What are stop words in NLP?**

 Stop words are the unwanted text that is present in the input. It is the process of removal of unwanted text from further processing of text, for example, a, to, can, etc.

## **34. How to find word similarity in NLP?**

 Word similarity in NLP is done by calculating the word vectors of the words in the vector space and then calculating the similarity on a scale of 0 to 1.

## **35. How to find sentence similarity in NLP?**

 Sentence similarity is done in NLP by finding the cosine similarity between the two sentences. It can be done by finding the cosine angle between the vectors of two sentences in the inner product space.

## **36. How to find document similarity in NLP?**

 Document similarity is done in NLP by converting the documents into the TF-IDF vectors form and finding their cosine similarity.

## **37. What are transformers?**

Transformers are deep learning architectures that can parallelize computations. They are used to learn long-term dependencies.

## **38. What are punctuations in NLP, and how can we remove them?**

Punctuations are the punctuations in the corpus or the input text. We can remove them by using the tokenizer function of NLTK. We can use nltk.RegexpTokenizer() to remove all punctuations.

## **39. What is latent semantic indexing (LSI)?**

Latent Semantic Indexing,, also referred to as latent semantic analysis, is an **NLP technique**used to remove stop words from processing the text into the text’s main content. It is used to find relationships between different words.

A cartoon of a person and a robot

Description automatically generated

## **40. What is named entity recognition (NER)?**

Named Entity Recognition is a part of information retrieval, a method to locate and classify the entities present in the unstructured data provided and convert them into predefined categories.

## **41. What is NLTK in NLP?**

 NTLK, an abbreviation of Natural Language Toolkit, is one of NLP’s most popular libraries. It was written in Python and contained libraries for tokenization, classification, tagging, stemming, parsing, and semantic reasoning.

## **42. What is spaCY?**

spaCY is an open-source library for natural language processing on an advanced level. It is mostly used for production-level usage and uses convolutional neural network models.

## **43. What is openNLP?**

openNLP is a java based library used for Natural Language Processing, and it supports most of the NLP tasks such as tokenization, language detection, etc.

## **44. What is the difference between NLTK and openNLP?**

There is a small difference between NTLK and openNLP, i.e., NLTK is written in python, and openNLP is based on java. One other difference is that NTLK has an option of downloading corpora by an in-built method.

## **45. What is parsing?**

Parsing is the method of analyzing the sentence automatically based on the syntactic structure.

## **46. What is dependency parsing?**

Dependency parsing, also called syntactic parsing, recognizes a dependency parse of a sentence and assigns a syntax structure to a sentence. It focuses on the relationship between different words.

## **47. What is semantic parsing?**

Semantic parsing is a method of conversion of natural language into machine-understandable form.

## **48. What is constituency parsing?**

 Constituency parsing is a method of division of sentences into sub-parts or constituencies. It aims to extract a constituency-based parse tree from the constituencies of the sentences.

## **49. What is shallow parsing?**

Shallow parsing, also known as light parsing and chunking, identifies constituents of sentences and then links them to different groups of grammatical meanings.

## **50. What are the differences between dependency parsing and shallow parsing?**

The difference between shallow parsing and dependency parsing is that shallow parsing is the parsing of limited parts of the information. In contrast, dependency parsing is the process of finding relations between all the different words.

## **51. What is language modeling?**

Language modeling is the process of creating a probability distribution of a sequence of words. It is used to provide probability to all the words present in the sequence.

## **52. What is topic modeling?**

 Topic modeling is a method of finding abstract topics in a document or set of documents to find hidden semantic structures.

## **53. What is text summarization in NLP?**

**Text summarization in NLP**is the process of conversion of large pieces of text to short text. It is intended to summarize the given text, keeping the main contents and overall meaning intact.

## **54. What is the difference between a regular expression and regular grammar?**

The difference between regular and regular grammar is that regular grammar is used to generate regular language, and regular expression is used to represent regular language.

## **55. What is perplexity in NLP?**

Perplexity is the condition when the system encounters something unaccountable or which is not meaningful.

## **56. What is the Naive Bayes algorithm, and where is it used in NLP?**

Naive Bayes algorithm is used to predict tags of text by calculating the probability for each tag for the text and then providing the one with the highest probability.

## **57. What is the PageRank algorithm?**

Google uses the PageRank algorithm. It is the algorithm to rank web pages in the search engine results.

## **58. What is noise removal?**

 Noise removal is one of the **NLP techniques**i.e., used to remove pieces of text from the corpus that is not necessary as they can hinder our text analysis.

## **59. What is word embedding?**

 Word embedding is the process of mapping words from the vocabulary to vectors of real numbers.

A group of people talking to a robot

Description automatically generated

## **60. What are the word embedding libraries?**

 The libraries that provide word embedding features are spaCY and genism.

## **61. What is word2vec?**

Word2vec is a collection of models that are used to generate word embeddings. These models are trained to reconstruct the linguistic context of the words in the corpus.

## **61. What is doc2vec?**

 Doc2vec is one of the unsupervised algorithms used to generate vectors of sentences or documents irrespective of their length.

## **63. What is a document-term matrix?**

The document-term matrix, also called the term-document matrix, is the matrix that describes the frequency of terms occurring in a document.

## **64. What is wordnet?**

 Wordnet can be described as a database created to store words from different languages connected by their semantic relationships.

## **65. What is GloVe in NLP?**

   The gloVebased on their pronunciation.

## **66. What is a flexible string matching?**

 Flexible string matching or fuzzy string matching is a method to find strings that are likely to match a specific pattern. It is also called approximate string matching as it uses an approximation to find patterns between strings.

## **67. What is cosine similarity?**

 Cosine similarity is the measure of cosine difference between two non-zero vectors in the inner product space. It is used to find the similarity between documents irrespective of their size.

## **68. What is information extraction?**

Information extraction is the process of extracting useful data in a structured way from a given unstructured set of data.

## **69. What is object standardization, and when is it used?**

Object standardization is the process of extracting useful information from abbreviations and other acronyms that can not be meaningful in lexical dictionaries.

## **71. What is text generation, and when is it done?**

   Text-generation is the process of generating natural language texts automatically in response to the communication. It uses artificial intelligence and computational linguistic knowledge to perform this task.

## **71. How can we estimate the entropy of the English language?**

N-grams can estimate the entropy of the English language. The entropy of a letter is calculated by knowing the entropy of all the previous letters.

## **72. What is Latent Dirichlet Allocation?**

 Latent Dirichlet Allocation is a topic modeling method where each topic represents a set of words, and every document is made of various words.

## **73. What are the conditional random fields?**

Conditional Random Fields (CRFs ) are a collection of statistical modeling methods. It is used for pattern recognition and structure prediction.

## **74. What are the hidden Markov random fields?**

Hidden Markov Random fields are a derivation of the Hidden Markov Model. It is a process generated by a Markov chain, whose state sequence can only be observed by a sequence of observations.

## **75. What is a coreference resolution?**

Coreference resolution is the process of collecting all the expressions that are referring to the same entity in a text. It is used in information extraction, document summarization, and question answering.

## **76. What is PAC learning?**

Probably Approximately Correct learning is a mathematical analysis framework. It is used for the analysis of generalization error of the learning algorithms.

## **77. What is sequence learning?**

Sequence learning is a method of learning where both input and output are sequences.

## **78. What is an ensemble method?**

 The ensemble method uses multiple learning algorithms to get enhanced and more accurate performance compared to the performance of an algorithm alone.

So these were the most frequently asked **NLP interview questions**, prepare them well, and increase your chances of getting selected.