DATA SCIENCE - NATURAL LANGUAGE PROCESSING

QUICK NOTES

S.NO	Topic Name	Summary
1	**NLP**	"Natural Language Processing "
2	**What is NLP**	Natural language processing (NLP) refers to the branch of computer science—and more specifically, the branch of artificial intelligence or AI—concerned with giving computers the ability to understand text and spoken words in much the same way human beings can.
3	**Why NLP**	Natural language processing helps computers communicate with humans in their own language and scales other language-related tasks. For example, NLP makes it possible for computers to read text, hear speech, interpret it, measure sentiment and determine which parts are important.

4	**NLP=NLU+NLG**	At a high level, NLU and NLG are just components of NLP, Natural language understanding (NLU) focuses on machine reading comprehension through grammar and context, enabling it to determine the intended meaning of a sentence. Natural language generation (NLG) focuses on text generation, or the construction of text in English or other languages, by a machine and based on a given dataset.
5	**General Steps we follow in NLP**	1.Collection of data 2. Sentence Segmentation 3. Tokenization 4.Lowecase 5. Removing of Stopwords 6.Stemming or Lemmatizations 7.Words to vectors 8. ML/DL Algorithms
6	**Basic Terms used in NLP**	1.Tokenization 2.Stemming 3.Lemmatization 4.Corpus 5.Stop Words 6.Parts-of-speech (POS) Tagging 7.Statistical Language Modeling 8.Bag of Words 9.n-grams 10.Regular Expressions 11.Zipf's Law 12.Similarity Measures 13.Syntactic Analysis 14.Semantic Analysis 15.Sentiment Analysis 16.Information Retrieval

7	**Tokenization**	Tokenization is used in natural language processing to split paragraphs and sentences into smaller units that can be more easily assigned meaning.
		Sentence-nltk.sent_tokenize()
		Words-nltk.word_tokenize()

8	**Lowercase**	The words are converted to lower case as python is case sensitive language .lower()
9	**Stop_words**	The words which are generally filtered out before processing a natural language are called stop words
10	**One-hot encoding**	One hot encoding is a vector representation of words in a “vocabulary”. Each word in the vocabulary is represented by a vector of size 'n', where 'n' is the total number of words in the vocabularyand it has disadvantages of Sparse Matrix issue.
11	**Bag of Words**	Bag-of-words(BoW) - analyze text and documents based on word count. The model does not account for word order within a document. BoW can be implemented as a Python dictionary with each key set to a word and each value set to the number of times that word appears in a text and has disadvantages to capture the relationships between words in a sentence and the meaning they convey.
12	**Term Frequency and Inverse Document Frequency**	Term frequency refers to the number of times that a term t occurs in document d. The inverse document frequency is a measure of whether a term is common or rare in a given document corpus

13	**steps involved in Text preprocessing -1**	1. Tokenization: breaking the text into individual words or tokens
		2. Lowercasing: converting all text to lowercase to reduce the vocabulary size and avoid treating the same word differently due to its case
		3. Stopword removal: removing common words such as "the," "and," and "is" that do not contribute to the meaning of a sentence
		4. Stemming: reducing words to their base or root form to group words with the same stem together
		5. Lemmatization: reducing words to their base or dictionary form to obtain their actual meaning
		6. Part-of-speech (POS) tagging: labeling the words in a sentence with their corresponding part of speech such as noun, verb, adjective, or adverb.

14	**continuous bag of words**	Continuous Bag of Words (CBOW) is a type of neural network-based language model used in natural language processing (NLP). It is a method for representing words as vectors that capture the meaning of the words based on their context within a text corpus. It is often used as a baseline model for more complex language tasks, such as sentiment analysis or named entity recognition. However, CBOW has some limitations, such as the inability to handle out-of-vocabulary words and the fact that it does not capture the order of words in a sentence.
15	**skipgram**	Skip-gram is another type of neural network-based language model used in natural language processing (NLP) for word embedding. It is a method for representing words as vectors that capture the meaning of the words based on their context within a text corpus. The skip-gram model is based on predicting context words given a target word.
16	**magic of DL in CBOW**	the magic of deep learning in CBOW lies in its ability to learn high-quality distributed representations of words that capture semantic relationships between words,
17	**GENSIM**	Gensim is a popular open-source Python library for natural language processing (NLP) and topic modeling. It provides easy-to-use interfaces for various NLP tasks, such as building word embeddings, topic modeling, text summarization, and similarity detection.

18	**SOME PRE-TRANIED MODEL IN NLP?**	BERT ,Word2Vec,GloVe,RNN, LSTM-RNN
19	**RNN**	RNN stands for Recurrent Neural NetworK designed to process sequential data, such as time series data or text data. The key feature of RNNs is that they have a feedback loop that allows information to be passed from one time step to the next.RNNs are particularly useful for tasks such as speech recognition, language translation, and sentiment analysis, where the input is a sequence of words or sentences. They can be trained using backpropagation through time (BPTT),One of the challenges of training RNNs is the vanishing gradient problem, where the gradients become very small as they are backpropagated through time
20	**LSTM**	LSTM stands for Long Short-Term Memory, which is a type of recurrent neural network (RNN) architecture that is designed to better handle long-term dependencies in sequential data.The key idea behind LSTM is the use of memory cells, which allow the network to selectively remember or forget information from previous time steps. Each memory cell contains three gates: an input gate, a forget gate, and an output gate

21	**GRU**	GRU stands for Gated Recurrent Unit, which is a type of recurrent neural network (RNN) GRU is designed to better handle long-term dependencies in sequential data. It achieves this through the use of gating mechanisms GRU has two gates: an update gate and a reset gate, It can be trained using backpropagation through time (BPTT) and has been shown to achieve state-of-the-art performance on various NLP tasks
22	**BI-DIRECTIONAL LSTM**	**Bidirectional LSTM** (BiLSTM) is a type of recurrent neural network (RNN) that allows information to be processed in both forward and backward directions. This is achieved by using two separate LSTM layers, one processing the input sequence in forward direction, and the other processing it in backward direction and useful for tasks such as language modeling, where the context of the surrounding words is important for predicting the next word. It has also been shown to be effective for tasks such as named entity recognition, sentiment analysis, and machine translation.
23	**SEQUENCE TO SEQUENCE IN NN?**	The basic idea behind **Seq2Seq** is to use two separate recurrent neural networks (RNNs), an encoder and a decoder. The encoder takes the input sequence and encodes it into a fixed-length vector, which is then passed to the decoder. The decoder uses this vector to generate the output sequence.

24	**ENCODER AND DECODER IN NLP?**	The **encoder** is a neural network that takes an input sequence (e.g., a sentence) and transforms it into a fixed-length vector or a sequence of vectors, which captures the important information in the input sequence. This encoding process is typically done using recurrent neural networks (RNNs), such as LSTM or GRU.
		The **decoder** is another neural network that takes the output of the encoder and generates an output sequence (e.g., a translation or a summary). The decoder is also typically implemented using RNNs, and it generates the output sequence one element at a time, using the previous element as inp
25	**BLEU SCORE?**	BLEU (Bilingual Evaluation Understudy) is a metric for evaluating the quality of machine translations compared to human translations. It was originally proposed for machine translation but has since been applied to other natural language processing tasks as well.
26	**CONTEXT OF INDEX IN NLP?**	In Natural Language Processing (NLP), an index refers to a data structure that is used to organize and retrieve information from a corpus or collection of documents efficiently.

27	**TRANSFORMERS IN NLP?**	Transformers are a type of neural network architecture that have been widely used in Natural Language Processing (NLP) tasks, particularly for tasks such as language modeling, machine translation, and text generation.Transformers use self-attention mechanisms, which enable the network to attend to different parts of the input sequence with varying degrees of importance, allowing for better capture of long-range dependencies and relationships between words in a sentence.The most famous implementation of a transformer is the Transformer model introduced in the paper "Attention is All You Need"(2017)
28	**WHAT IS EMBEDDINGS, STATIC AND CONTEXT BASED EMBEDDINGS?**	Static embeddings, also known as pre-trained embeddings, are fixed representations of words that are learned from large corpora of text data. Common examples of static embeddings include Word2Vec, GloVe, and FastText.
		Context-based embeddings, also known as contextualized embeddings, are learned by considering the context in which a word appears. These embeddings take into account the surrounding words in a sentence or document and generate a unique embedding for each occurrence of a word. Examples of context-based embeddings include ELMo, BERT, and GPT.

29	**BERT AND ITS APPLICATIONS?**	BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained transformer-based language model designed to understand the context of words in a sentence and can generate context-based word embeddings that can be used for various downstream NLP tasks and its applications were : Text classification,Question answering,Named entity recognition,Natural language inference,Text generation:
30	**PRE-TRAINED BERT MODEL?**	The original BERT model (BERT-Base and BERT-Large) as well as smaller variants such as DistilBERT and TinyBERT
31	**PRE-TRAINING PROCESS OF BERT MODEL?**	1. **Masked Language Modeling** (MLM): The model is trained to predict masked words in a sentence. This involves randomly masking out some words in a sentence and training the model to predict the masked words based on the context of the surrounding words.

		2. **Next Sentence Prediction** (NSP): The model is trained to predict whether two sentences in a sequence are consecutive or not. This involves training the model to determine whether a second sentence in a sequence follows logically from the first sentence.
32	**WORD PIECE TOKENIZER?**	**Word Piece Tokenizer (WPT)** is a subword-level tokenizer used in Natural Language Processing (NLP) to split words into smaller subword units. The WPT algorithm was introduced by Google as part of the BERT (Bidirectional Encoder Representations from Transformers) model.

33	**LANGUAGE MODELLING IN NLP?**	**Language modeling** is a central task in Natural Language Processing (NLP) that involves predicting the probability of a sequence of words in a given language.Language modeling can be done in two ways: **Unconditional language modeling**: This involves training a model to generate text without any specific context. The model is trained to predict the probability of the next word in a sequence based solely on the previous words in the sequence. **Conditional language modeling:** This involves training a model to generate text based on a specific context, such as a prompt or a starting sentence. The model is trained to predict the probability of the next word in a sequence based on both the previous words in the sequence and the given context.
34	**BERT CASED AND UNCASED FORMATS?**	The cased version of BERT preserves the original case of the text, while the uncased version converts all text to lowercase. For example, the cased version would treat "Apple" and "apple" as two distinct tokens, while the uncased version would treat them as the same token.

35	**HUGGING FACE TRANSFORMERS?**	**Hugging Face Transformers** is an open-source library developed by Hugging Face,which can be fine-tuned on task-specific datasets for various NLP tasks such as text classification, question answering, and language translation.The library provides pre-trained models in PyTorch and TensorFlow, making it accessible to users of both frameworks.The library also includes utilities for tokenization, data preprocessing, and evaluation of NLP models.
36	**HIDDEN REP AND CLS HEAD IN NLP?**	In Natural Language Processing (NLP), the term "**hidden representation**" generally refers to the output of the final layer of a pre-trained language model that captures a contextualized encoding of the input text, where each token in the input sequence is represented as a high-dimensional vector that encodes its meaning and context in the input sequence.
		The "**CLS" head** in NLP stands for "classification head" and refers to a neural network layer that is typically added on top of the hidden representation of a pre-trained language model.The CLS head works by taking the hidden representation of the special [CLS] token that is added to the beginning of the input sequence in BERT, and passing it through a linear layer and activation function to produce the final output vector. This output vector is then used as input to a final classification layer that outputs the predicted class for the input text.

37	**Text classification**	Text classification: BERT can be fine-tuned for text classification tasks such as sentiment analysis, topic classification, and spam detection. Fine-tuning involves training the BERT model on a task-specific dataset to generate task-specific embeddings, which can then be used as inputs to a classification model.
38	**Question answering:**	Question answering: BERT can be used for question answering tasks, where the model is trained to generate answers to questions based on a given context. This has applications in tasks such as chatbots and customer service.
39	**Named entity recognition**	Named entity recognition: BERT can be fine-tuned for named entity recognition, where the model is trained to identify and classify named entities such as people, organizations, and locations in a text.
40	**Natural language inference**	Natural language inference: BERT can be used for natural language inference tasks, where the model is trained to determine the logical relationship between two sentences, such as whether one sentence entails, contradicts, or is neutral with respect to another.
41	**Text generation**	Text generation: BERT can be used for text generation tasks such as summarization, paraphrasing, and text completion, where the model generates text based on a given prompt or context.

`

