1. Write program demonstrates how to use regular expressions in Python to match and search for patterns in text.

2. Implement a basic finite state automaton that recognizes a specific language or pattern. In this example, we'll create a simple automaton to match strings ending with 'ab' using python.

3. Write program demonstrates how to perform morphological analysis using the NLTK library in Python.

4. Implement a finite-state machine for morphological parsing. In this example, we'll create a simple machine to generate plural forms of English nouns using python.

5. Use the Porter Stemmer algorithm to perform word stemming on a list of words using python libraries.

6. Implement a basic N-gram model for text generation. For example, generate text using a bigram model using python.

7. Write a program using the NLTK library to perform part-of-speech tagging on a text.

8. Implement a simple stochastic part-of-speech tagging algorithm using a basic probabilistic model to assign POS tags using python.

9. Implement a rule-based part-of-speech tagging system using regular expressions using python.

10. Implement transformation-based tagging using a set of transformation rules, apply a simple rule to tag words using python.

11. Implement a simple top-down parser for context-free grammars using python.

12. Implement an Earley parser for context-free grammars using a simple python program.

13. Generate a parse tree for a given sentence using a context-free grammar using python program.

14. Create a program in python to check for agreement in sentences based on a context-free grammar's rules.

15. Implement probabilistic context-free grammar parsing for a sentence using python.

16. Implement a Python program using the SpaCy library to perform Named Entity Recognition (NER) on a given text.

17. Write program demonstrates how to access WordNet, a lexical database, to retrieve synsets and explore word meanings in python.

18. Implement a simple FOPC parser for basic logical expressions using python program.

19. Create a program for word sense disambiguation using the Lesk algorithm using python.

20. Implement a basic information retrieval system using TF-IDF (Term Frequency-Inverse Document Frequency) for document ranking using python.

21. Create a python program that performs syntax-driven semantic analysis by extracting noun phrases and their meanings from a sentence.

22. Create a python program that performs reference resolution within a text.

23. Develop a python program that evaluates the coherence of a given text.

24. Create a python program that recognizes dialog acts in a given dialog or conversation.

25. Utilize the GPT-3 model to generate text based on a given prompt. Make sure to install the OpenAI GPT-3 library in python implementation.

26. Implement a machine translation program using the Hugging Face Transformers library, translate English text to French using python

Set-1

1. Write a Python program that demonstrates how to use regular expressions to validate and extract email addresses from a given text.

2. Implement a text classification program using the Naive Bayes algorithm to classify text documents into categories (e.g., spam detection).

3. Create a Python program to identify and extract key phrases or keywords from a given

text using techniques such as TF-IDF. The sentences = [

"Artificial intelligence (Al) is a field of computer science.",

"Machine learning is a subset of Al that focuses on training models to make predictions.",

"Deep learning is a type of machine learning that uses neural networks with multiple layers.",

"Neural networks are composed of interconnected nodes called neurons.",

"Recurrent neural networks (RNNs) are commonly used in natural language processing tasks.",]

4. Write a Python program that utilizes NLTK's named entity recognition to extract named entities (e.g., person names, locations) from a given text. Write a Python program tha performs reference resolution within a given text "Harvard University, located i Cambridge, Massachusetts, is a prestigious institution.".

Set2

1. Implement a Python program that defines a finite state automaton to recognize strings with an equal number of 'O's and 'I's.

2. Write a Python program using the SpaCy library to perform Named Entity Recognition for classifying named entities in text, on a given sentence "Barack Obama was the 44th President of the United States, and he was born in Honolulu, Hawaii".

3. Implement a Python-based named entity disambiguation program that resolves entity mentions to their corresponding Wikipedia entities using wikipediaapi. Use the following input sentences "Apple is a leading tech company.",

"I love apples as a fruit.",

"Python is a popular programming language.",

"The python is a non-venomous snake."

4. Develop a Python program that evaluates the coherence of a provided sample text =\*\*\*\* Once upon a time, there was a young boy named Peter.

He lived in a small village.

One day, he decided to explore the nearby forest.'"''

Set 3

1. Write a Python program that uses the NLTK library to perform morphological analysis on sentence "Unhappily, she ran quickly".

2. Develop a program for sentiment analysis (positive, negative, or neutral) using textblob library, (e.g., using a pre-trained model or building a custom model), to the sentences "I love this product! It's amazing.", "The weather is terrible today.".

3. Write a Python program that demonstrates how to access WordNet, a lexical database, to retrieve synesis and explore word meanings. Provides information about the meanings, relationships, and synonyms of words in the English language for the following "The cat sat on the mat.", "The dog barked loudly.", "She played the piano beautifully."

4. Create a Python program to recognize dialog acts in a given dialog or conversation = [

"Good morning! How's the weather today?"

"I heard it's going to be sunny and warm.",

"Could you please send me the report by 3 PM?",

"Of course, I'll send it over before the deadline.",

"Do you know where the nearest post office is?"

"The post office is two blocks down the street."].

Set 4

1. Implement a Python program that creates a finite-state machine for parsing and generating the past tense forms of English verbs using the sentences "She walked to the park yesterday", "He jumped over the fence".

2. Write a Python program to perform extractive text summarization on a longer text document with libraries NLTK or Gensim.

document = \*\*\*\*\*

Natural language processing (NLP) is a subfield of artificial intelligence (Al) that focuses on the interaction between computers and humans through natural language.

NLP technologies are used to process, analyze, and understand large amounts of natural language data.

One of the primary applications of NLP is sentiment analysis, which determines the sentiment or emotional tone of a piece of text. Sentiment analysis is widely used in social media monitoring, customer feedback analysis, and brand reputation management.

Text summarization is another important NLP task. Extractive summarization involves selecting a subset of sentences from a text to create a shorter version that retains the most critical information. Abstractive summarization, on the other hand, generates a

noting sut of s

summary by paraphrasing and rephrasing the original text.\*™™

3. Implement a simple First-Order Predicate Calculus (FOPC) parser for basic logical

expressions using Python. The expressions = [ "x and y", "x or (not y)", "x and (y or

(not x)) "].

4. Implement a Python program that utilizes the GPT-3 model to generate text based on user prompts.

Set 6

1. Write a Python program that uses NLTK to perform part-of-speech tagging on a given text "The sun is shining brightly", "I love reading interesting books".

2. Implement a Python program using the SpaCy library to perform Named Entity Recognition (NER) on a given sentences "Apple Inc. is headquartered in Cupertino, California, and its CEO, Tim Cook, often delivers keynote speeches.", "The Eiffel Tower in Paris, France, is a popular tourist attraction."

3

Implement a basic information retrieval system using TF-IDF (Term Frequency-

Inverse Document Frequency) for document ranking in Python using the sentences = [

"Climate change is a pressing global issue that requires immediate action.",

"Renewable energy sources, such as solar and wind power, are essential for reducing carbon emissions.",

"Greenhouse gases, like carbon dioxide and methane, contribute to global warming.",

"The Paris Agreement is an international treaty aimed at addressing climate change.",

"Sustainability and environmental conservation are crucial for the future of our planet.".

4. Write a Python program that takes a text and a regular expression pattern as input, and demonstrates how to use regular expressions to find and extract specific patterns in the

text. Using the pattern = r'b|w{3}\b' and input sentence "The quick brown fox jumps

over the lazy dog. The cat is also agile."

Set 7   
1.Create a Python program that demonstrates stochastic part-of-speech tagging for the given sentences "The red car stopped at the traffic light", "She quickly ran to catch the bus".

2. Create a Python program for abstractive text summarization, a more advanced technique that generates summaries by rewriting the content in a human-readable form.

Compose using following sentence "The World Health Organization (WHO) plays a vital role in global health. WHO is headquartered in Geneva, Switzerland, and it is responsible for coordinating international efforts to control and prevent the spread of diseases? Its mission is to promote and protect the health of people worldwide."

3. Write a Python program that performs syntax-driven semantic analysis by extracting

1oun phrases and their meanings from the sentence "The quick brown fox jumps over the lazy dog.", "She is an excellent chef and loves to cook delicious meals.",

"The Eiffel

Tower in Paris is a famous landmark.".

4. Write a Python program that showcases how to perform morphological analysis using the NLTK library. Use a sample word and show how to lemmatize it "The quick brown foxes jumped over the lazy dogs.", "I am running in the park with my friends."

Set 8

1. Implement a Python program that performs rule-based part-of-speech tagging using regular expressions using the following rules.

patterns = [

(г|b(?: The|the)\b', 'DET'), # Matches 'The' or 'the' as determiners (r\b?:cat|dog)\b', 'NOUN'), # Matches 'cat' or 'dog' as nouns (г\b(?:is/am/are)\b', 'VERB'), # Matches 'is,' 'am,' or 'are' as verbs (г'b?:quickly|brightly)\b', 'ADV'), # Matches 'quickly' or 'brightly' as adverbs (b(?:[A-Za-z]+)\b', 'NOUN) # Matches any other words as nouns]

2. Develop a Python program that performs Named Entity Recognition (NER) on a given text using popular libraries or models for the sentence "The capital of France is Paris, and it's known for the Eiffel Tower."

3. Develop a Python program for parsing sentences using a Probabilistic Context-Free Grammar (PCFG). Use the PCNF grammar S -> NP VP [1.0], NP -> Det N [0.5] | NP PP [0.4] | 'the' [0.1], VP -> V NP [0.7] | VP PP [0.3], PP > P NP [1.01, Det -> 'the' [0.7] l'a' [0.3], N →> 'fox' [0.4] | 'dog' [0.3] | 'cat' [0.2] | 'bird' [0.1], V →> 'jumps' [0.5] | 'runs'

[0.3] | 'sits' [0.2], P →> 'over' [0.6] | 'on' [0.4] and use the input string "The quick brown fox jumps over the lazy dog."

4. Write a Python program that employs the Porter Stemmer algorithm from a Python library to perform word stemming on a list of words. The sample sentences are "Coding with Python is very enjoyable.", "I had a delicious meal at the restaurant."

Set 9

1. Develop a Python program that uses a PCFG to parse a sentence using "The cat chased the mouse", the probabilities in square brackets indicate the likelihood of each rule need to apply.

S -> NP VP [0.6]

NP -> Det N [0.3]

NP →> NP PP [0.2]

VP →> V NP [0.5]

VP →> VP PP [0.4]

PP -> P NP [0.7]

2. Create a Python program for sentiment analysis which can be positive, negative, or neutral on a set of text data "I love this product! It's amazing.".

3. Create a Python program to recognize dialog acts in a given dialog or conversation = [

"Hello! How are you today?",

"I'm doing well, thank you. How about you?",

"Can you please pass the salt?",

"Sure, here you go.",

"What time is the meeting tomorrow?",

"The meeting is at 2:00 PM."].

4. Implement a Python program that constructs a finite state automaton capable of recognizing dates in a specific format (e.g., DD/MM/YYYY).

Set 10

Write a Python program that performs information retrieval using the TF-IDF score.

Sample Documents:

Document 1: "Natural language processing (NLP) is a field of study in artificial intelligence."

Document 2: "NLP techniques are used in various applications like machine translation and sentiment analysis."

Document 3: "The development of NLP tools and libraries has made text analysis easier."

2. Develop a Python program for dependency parsing of a sentence using a dependency parser to the sentences "John and Mary went to the store.", "The big brown dog chased the small black cat.".

3. Implement a machine translation program using the Hugging Face Transformers library in Python using model "Helsinki-NLP/opus-mt-en-de" to translate English to German.

4. Implement a simple First-Order Predicate Calculus (FOPC) parser for basic logical

expressions using Python. The variables = {'p': True, 'q: True, T': False) and

expressions = [ "p and a", "p or r", "not p", "q and (r or p)"].