Sr. NO	Practical Title	Hours	CLO
1.	Exploration of different frameworks, libraries and SDKs for Natural Language Processing	02 Hours	1
2.	a. Write a program that can fetch email addresses and URLs from the text. (Hint: Using RE module) b. Write a program to count words from the given input text file. (Hint: Using NLTK module) c. Write a program to implement sentence tokenizer using NLTK and spaCy.	02 Hours	
3.	a. Write a program to perform stemming at word level as well as sentence level using porterstemmer algorithm. b. Write a program to perform POS tagging for the given sentence/Paragraph.	02 Hours	
4.	To Explore and implement Text Preprocessing and Data Representation (Hint: Sentiment analysis using Word2vec and Glove encoding).	04 Hours	4
5.	Explore and understand Sequence to Sequence learning (RNN, LSTM, GRU based processing of text data)	02 hours	4
6.	Implement a technique for author attribution. Classify documents based on authors. A person's writing style is an example of a behavioral biometric. The words people use and the way they structure their sentences is distinctive, and can often be used to identify the author of a particular work. This is a widely studied problem, with hundreds of academic papers on the subject. There are two high-level ways to attack the chapter attribution problem: 1. Supervised learning: One approach would be to gather ground truth from external sources. For example, find works for each author from other publications, blogs, etc. These samples would be used to learn a model for each author's writing style. Determining who wrote each chapter would be a binary classification problem. 2. Unsupervised learning: A second approach is unsupervised, meaning that the analysis is conducted without ground truth. In this method, the chapters are analysed to find two subsets that appear to have been written by the same person. We use the second approach of unsupervised learning in this practical.	04 hours	4
7.	Implement the text summarization using any one of the techniques. Theory: Text summarization can broadly be divided into two categories — Extractive Summarization and Abstractive Summarization. Extractive Summarization: These methods rely on extracting several parts, such as phrases and sentences, from a piece of text and stack them together to create a summary. Therefore, identifying the right sentences for summarization is of utmost importance in an extractive method. Abstractive Summarization: These methods use advanced NLP techniques to generate an entirely new summary. Some parts of this summary may not even appear in the original text. In this program, we implement "Text Rank" algorithm which is an Extractive summarization technique.	04 Hours	4
8.	Take input one paragraph of 500 words and implement Natural Machine Translation	04 hoours	4
9.	Implement a stylish-text-generation system, that captures the author's writing style and generate the text based on it.	05 hours	4

	Text generation usually involves the following steps: 1. Importing Dependencies 2. Loading of Data 3. Creating Character/Word mappings 4. Data Preprocessing 5. Modelling 6. Generating text		
10.	NLP and Deep Learning based Project	05 hours	4