Paper – V (A): Natural Language Processing

[4 HPW :: 4 Credits :: 100 Marks (External:80, Internal:20)]

Objective: The main objective of this course is to give a practical introduction to NLP. It deals with morphological processing, syntactic parsing, information extraction, probabilistic NLP and classification of text using Python's NLTK Library.

Outcomes:

At the end of the course the student will be able to

- Write Python programs to manipulate and analyze language data
- Understand key concepts from NLP and linguistics to describe and analyze language
- Understand the data structures and algorithms that are used in NLP
- Classify texts using machine learning and deep learning

Unit-I

Language Processing and Python: Computing with Language: Texts and Words, A Closer Look at Python: Texts as Lists of Words, Computing with Language: Simple Statistics, Back to Python: Making Decisions and Taking Control, Automatic Natural Language Understanding [Reference 1]

Accessing Text Corpora and Lexical Resources: Accessing Text Corpora, Conditional Frequency Distributions, Lexical Resources, WordNet [Reference 1]

Unit-II

Processing Raw Text: Accessing Text from the Web and from Disk, Strings: Text Processing at the Lowest Level, Text Processing with Unicode, Regular Expressions for Detecting Word Patterns, Useful Applications of Regular Expressions, Normalizing Text, Regular Expressions for Tokenizing Text, Segmentation, Formatting: From Lists to Strings. [Reference 1]

Categorizing and Tagging Words: Using a Tagger, Tagged Corpora, Mapping Words to Properties Using Python Dictionaries, Automatic Tagging, N-Gram Tagging, Transformation-Based Tagging, How to Determine the Category of a Word [Reference 1]

Unit-III

Learning to Classify Text: Supervised Classification, Evaluation, Naive Bayes Classifiers [Reference 1]

Deep Learning for NLP: Introduction to Deep Learning, Convolutional Neural Networks, Recurrent Neural Networks, Classifying Text with Deep Learning [Reference 2]

Unit-IV

Extracting Information from Text

Information Extraction, Chunking, Developing and Evaluating Chunkers, Recursion in Linguistic Structure, Named Entity Recognition, Relation Extraction. [Reference 1]

Analyzing Sentence Structure
Some Grammatical Dilemmas, What's the Use of Syntax. Context-Free Grammar, Parsing with Context-Free Grammar, [Reference 1]

References:

1. Natural Language Processingwith Python. Steven Bird, Ewan Klein, and Edward Lope, O'Reily, 2009
2. Natural Language Processing Recipes: Unlocking Text Data with Machine Learning and Deep Learning using Python. AkshayKulkarni,AdarshaShivananda, Apress, 2019

Suggested Reading:

3. Allen James, Natural Language Understanding, Benjamin/Cumming,1995.
4. Charniack, Eugene, Statistical Language Learning, MIT Press, 1993.

Practical – 5(A): Natural Language Processing (Lab)

[3 HPW:: 1 Credit :: 25 Marks]

Objective: The main objective of this laboratory is to write programs that manipulate and analyze language data using Python

This lab requires mentoring sessions from TCS.

Python Packages

Students are expected to know/ learn the following PythonNLP packages

- NLTK (www.nltk.org/ (http://www.nltk.org/))
- Spacy (https://spacy.io/)
- TextBlob (http://textblob.readthedocs.io/en/dev/
- Gensim (https://pypi.python.org/pypi/gensim)
- Pattern (https://pypi.python.org/pypi/Pattern)

Datasets:

1. NLTK includes a small selection of texts from the Project Gutenberg electronic text archive, which contains some 25,000 free electronic books, hosted at *http://www.gutenberg.org/*.
2. The Brown Corpus contains text from 500 sources, and the sources have been categorized by genre, such as *news*, *editorial*, and so on (*http://icame.uib.no/brown/bcm-los.html*).
3. Wikipedia Articles Or any other dataset of your choice

Reference:

Jacob Perkins. Python 3 Text Processing with NLTK 3 Cookbook. Packt Publishing. 2014

Exercises:

1. Text segmentation: Segment a text into linguistically meaningful units, such as paragraphs, sentences, or words. Write programs to segment text (in different formats) into tokens (words and word-like units) using regular expressions. Compare an automatic tokenization with a gold standard
2. Part-of-speech tagging: Label words (tokens) with parts of speech such as noun, adjective, and verb using a variety of tagging methods, e.g., default tagger, regular expression tagger, unigram tagger, and n-gram taggers.
3. Text classification: Categorize text documents into predefined classes using Naïve Bayes Classifier and the Perceptron model
4. Chunk extraction, or partial parsing: Extract short phrases from a part-of-speech tagged sentence. This is different from full parsing in that we're interested in standalone chunks, or phrases, instead of full parse trees
5. Parsing: parsing specific kinds of data, focusing primarily on dates, times, and HTML. Make use of the following preprocessing libraries:
   - dateutil which provides datetime parsing and timezone conversion
   - lxml and BeautifulSoup which can parse, clean, and convert HTML
   - charade and UnicodeDammit which can detect and convert text character encoding
6. Sentiment Analysis: Using Libraries TextBlob and nltk, give the sentiment of a document

(B): NoSQL Data Bases

[4 HPW :: 4 Credits :: 100 Marks (External:80, Internal:20)]

Objective: The main objective of this course is to cover core concepts of NoSQL databases, along with an example database for each ofthe key-value, document, column family, and graph databases

Outcomes:
At the end of the course the student will be able to

- Understand the need for NoSQL databases and their characteristics
- Understand the concepts of NoSQL databases
- Implement the concepts of NoSQL databases using four example databases: Redis for key-value databases, MongoDB for document databases, Cassandra for column-family databases, and Neo4J for graphdatabases.

Unit-I

Why NoSQL: The Value of Relational Databases, Impedance Mismatch, Application and Integration Databases, Attack of the Clusters, The Emergence of NoSQL
Aggregate Data Models: Aggregates, Column-Family Stores, Summarizing Aggregate-Oriented Databases
More Details on Data Models: Relationships, Graph Databases, Schemaless Databases, Materialized Views, Modeling for Data Access

Unit-II

Distribution Models: Single Server, Sharding, Master-Slave Replication, Peer-to-Peer Replication, Combining Sharding and Replication
Consistency: Update Consistency, Read Consistency, Relaxing Consistency, Relaxing Durability, Quorums
Version Stamps: Business and System Transactions, Version Stamps on Multiple Nodes
Map-Reduce: Basic Map-Reduce, Partitioning and Combining, Composing Map-Reduce Calculations

Unit-III

Key-Value Databases: What Is a Key-Value Store, Key-Value Store Features, Suitable Use Cases, When Not to Use
Document Databases: What Is a Document Database, Features, Suitable Use Cases, When Not to Use

Unit-IV

Column-Family Stores: What Is a Column-Family Data Store, Features, Suitable Use Cases, When Not to Use
Graph Databases: What Is a Graph Database, Features, Suitable Use Cases, When Not to Use

Reference:

1. Pramod J. Sadalage, Martin Fowler. NoSQL Distilled, Addison Wesley 2013

Suggested Reading

2. Luc Perkins, Eric Redmond, Jim R. Wilson. Seven Databases in Seven Weeks. The Pragmatic Bookshelf, 2018
3. Guy Harrison. Next GenerationDatabases: NoSQL, NewSQL, and Big Data. Apress, 2015

Practical – 5(B) : NoSQL Data Bases (Lab)

[3 HPW :: 1 Credit :: 25 Marks]

Objective: The main objective of this lab is to become familiar with the four NoSQL databases: Redis for key-value databases, MongoDB for document databases, Cassandra for column-family databases, and Neo4J for graphdatabases

NoSQL Databases:
Redis (http://redis.io)
MongoDB (http://www.mongodb.org)
Cassandra (http://cassandra.apache.org)
Neo4j (http://neo4j.com)

Exercises:
1. Installation of NoSQL Databases: Redis,MongoDB, Cassandra, Neo4j on Windows & Linux
2. Practice CRUD (*Create*, *Read*, *Update*, and *Delete)* operations on the four databases: Redis,MongoDB, Cassandra, Neo4j
3. Usage of Where Clause equivalent in MongoDB
4. Usage of operations in MongoDB – AND in MongoDB, OR in MongoDB, Limit Records and Sort Records. Usage of operations in MongoDB – Indexing, Advanced Indexing, Aggregation and Map Reduce.
5. Practice with ' macdonalds ' collection data for document oriented database. Import restaurants collection and apply some queries to get specified output.
6. Write a program to count the number of occurrences of a word using MapReduce

Paper – VI - GE: Data Structures and Algorithms

[4 HPW:: 4 Credits :: 100 Marks]

Objectives:

- To introduce the time and space complexities of algorithms.
- To discuss the linear and non-linear data structures and their applications.
- To introduce the creation, insertion and deletion operations on binary search trees and balanced binary searchtrees.
- To introduce various internal sorting techniques and their time complexities

Outcomes:

Students will be

- Able to analyze the time and space complexities of algorithms.
- Able to implement linear, non-linear data structures and balanced binarytrees
- Able to analyze and implement various kinds of searching and sorting techniques.
- Able to find a suitable data structure and algorithm to solve a real world problem.

UNIT–I

Performance and Complexity Analysis: Space Complexity, Time Complexity, Asymptotic Notation (Big-Oh), Complexity Analysis Examples.

Linear List-Array Representation: Vector Representation, Multiple Lists Single Array. **Linear List-Linked Representation:** Singly Linked Lists, Circular Lists, Doubly Linked Lists, Applications (Polynomial Arithmetic).

Arrays and Matrices: Row and Column Major Representations, Sparse Matrices.

Stacks: Array Representation, Linked Representation, Applications (Recursive Calls, Infix to Postfix, Postfix Evaluation).

Queues: Array Representation, Linked Representation. **Skip Lists and Hashing:** Skip Lists Representation, Hash Table Representation, Application- Text Compression.

UNIT- II

Trees: Definitions and Properties, Representation of Binary Trees, Operations, Binary TreeTraversal.
Binary Search Trees: Definitions, Operations on Binary Search Trees.

Balanced Search Trees: AVL Trees, and B-Trees.

UNIT –III

Graphs: Definitions and Properties, Representation, Graph Search Methods (Depth First Search and Breadth First Search)
Application of Graphs: Shortest Path Algorithm (Dijkstra), Minimum Spanning Tree (Prim's and Kruskal's Algorithms).

UNIT –IV

Searching : Linear Search and Binary Search Techniques and their complexity analysis.
Sorting and Complexity Analysis: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, and Heap Sort.Algorithm Design Techniques: Greedy algorithm, divide-and-conquer, dynamic programming.

Suggested Reading:

1. Michael T. Goodrich, Roberto Tamassia, David M. Mount, *Data Structures and Algorithms Python* John Wiley & Sons, 2013.
2. Problem Solving with algorithams and Data Structures Using Python by Miller and David L. Ranum
3. Algorithmic Problem Solving with Python by John B. Schneider