Atal Bihari Vajpayee- Indian Institute of Information Technology and Management Gwalior

NLP Lab : 01

Introduction to **Natural Language Processing (NLP)**, **NLTK Toolkit**, and **PyTorch**

Instructor – Dr. Sunil Kumar
Assistant Professor @ Dept. of Information Technology, ABV-IIITM Gwalior

Teaching Assistants (TAs)

Meher (PhD Scholar)	Sheel Patel & Radhika [M.Tech IT]	Group-01
Shalmon Titre (MTech)	Lokesh Jain (MTech)	Group-02

Objectives

- To understand the concept and scope of **Natural Language Processing (NLP)**
- To study the **NLTK toolkit** and its role in text processing
- To analyze why **NLTK** is used for basic NLP tasks
- To explore **alternative NLP libraries and frameworks**
- To gain an introductory understanding of **PyTorch** for modern NLP applications

What is Natural Language Processing (NLP)?

Natural Language Processing (NLP) is a branch of Artificial Intelligence (AI) that enables computers to understand, interpret, and generate human language in a meaningful way.

Why NLP is Needed ?
Human language is: Ambiguous, Unstructured, Context-dependent
Computers work with numbers, so NLP converts language into a machine-understandable format.

Examples of NLP Applications
1 Spell Check : NLP detects incorrect words and suggests corrections.

- Input: I am hapy
- Output: I am happy

2 Autocomplete / Text Prediction : NLP predicts the next word.

- Input: How are y
- Output: How are you?

3 Chatbots : NLP bots understand questions and generate answers.

- User: What is NLP?
- Bot: NLP stands for Natural Language Processing...

4 Sentiment Analysis : NLP detects emotion from text.

- Sentence: This movie is amazing
- Output: Positive sentiment 😊
- Sentence: I hate this service
- Output: Negative sentiment 😠

5 Machine Translation : NLP converts text from one language to another.

- Input: Hello
- Output: Hola (Spanish)

What is the NLTK Toolkit?

NLTK (Natural Language Toolkit) is a **Python library** used for **teaching, research, and basic NLP tasks**.

It provides easy-to-use tools for working with text data.

Key Features of NLTK

- Tokenization
- Stop-word removal
- Stemming & Lemmatization
- POS tagging
- Named Entity Recognition
- Built-in corpora (datasets)

Why Use NLTK?

Reason	Explanation
Easy to learn	Simple Python APIs
Educational	Best for understanding NLP fundamentals
Pre-built datasets	Comes with text corpora
No deep learning required	Works with rule-based NLP

PyTorch

Aim: To study and implement the **basics of PyTorch**, including tensor creation, basic tensor operations, and a simple linear model

1. Tensor Creation

Description:

PyTorch tensors can be created from Python lists or NumPy arrays and are the basic data structure used for computation.

Example: Tensor from list: [1, 2, 3]

Tensor from NumPy array: [[1, 2], [3, 4]]

2. Basic Tensor Operations

Description: Basic mathematical operations such as addition and multiplication can be performed on PyTorch tensors.

Example:

Addition Result: [2, 4, 6]

Multiplication Result: [1, 4, 9]

PyTorch

Aim: To study and implement the **basics of PyTorch**, including tensor creation, basic tensor operations, and a simple linear model

3. Tensor Properties

Description: Tensor properties such as shape and data type help in understanding the structure of the data.

Example: Tensor Shape: (3,)

Tensor Data Type: torch.int64

4. Matrix multiplication:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \qquad B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

AxB =

Result Matrix =
[19 22
43 50]

5. Dot Products in Pytorch

Vector x = $[1, 2, 3]$

Vector y = $[4, 5, 6]$

Dot Product Output = 32

6. Tensor Indexing

Tensor T = [10 20 30
40 50 60]

Print output of : T[0], T[1], T[0][1], T[1][2]

7. Important functions in Pytorch:

- **List of useful torch command:**
  1. torch.empty()
  2. torch.zeros()
  3. torch.rand()
  4. torch.ones()
  5. torch.eyw()
  6. torch.arange(……)
  7. torch.linspace
  8. torch.add(x,y)
  9. torch.mm()
  10. torch.dot(x,y)
  11. torch.bmm()
  12. torch.abs()
  13. torch.argmax()
  14. torch.argmin()
  15. torch.mean()
  16. torch.sort()
  17. torch.cat()

Execute all the listed commands with suitable arguments

Thank you !