Graduate NLP Assignment_01: Regex and Edit Distance

Objective

This assignment aims to deepen your understanding of regex and edit distance by exploring their mathematical foundations and implementing them in code.

Part 1: Advanced Regular Expressions (Regex)

Task 1: Complex Pattern Extraction

Write a function to perform the following tasks:

1. Extract all URLs from the following text: `"Visit us at https://www.example.com or http://blog.example.org/index.html"`.
2. Extract all balanced parentheses from the string: `"a(b)c(d)e(f(g)h)i"`.
3. Extract all valid phone numbers in the format `(XXX) XXX-XXXX`.

Task 2: Regex Optimization

Two regex patterns for validating phone numbers in the format `(XXX) XXX-XXXX` are given below:

- Pattern 1: `^\(\d{3}\) \d{3}-\d{4}`
- Pattern 2: `^[(){1}]\d{3}[)]{1} \d{3}-\d{4}`

Compare the two patterns in terms of computational efficiency. Write code to test both patterns with a large dataset of phone numbers and report your findings.

Part 2: Edit Distance – Weighted and Practical Applications

Task 1: Weighted Edit Distance

Modify the standard edit distance algorithm to assign weights to operations:

- Insertion = 1
- Deletion = 2
- Substitution = 3

Write a Python function to compute the weighted edit distance and test it on the following pair:

- `"data"` → `"date"`

Task 2: Applications of Edit Distance

Given a dataset of words, write a script to find the closest match to a misspelled word using edit distance. Use the following dataset: `["natural", "language", "processing", "data", "science"]` Example input: `"natrual"`.

Task 3: Dynamic Programming Table Visualization

Implement the Levenshtein distance algorithm and print the dynamic programming table for the following pair:

- `"kitten"` → `"sitting"`

Submission Requirements

- You will work on g_starter_code.py and modify the starter script.
- Submit your regex patterns, explanations, and test outputs.
- Include your Python code for edit distance (weighted and unweighted) with test outputs.
- Provide an analysis of the computational efficiency of your regex patterns.