Upcoming events

Upcoming events | Information Retrieval (AIMLCZG537/DSECLZG537)(S... | New event

Upcoming events

Assignment - 1

Saturday, 7 June, 8:00 AM » Sunday, 22 June, 11:59 PM

Course event

Information Retrieval (AIMLCZG537/DSECLZG537)(S2-24)

Assignment - 1 Problem Set - 7 is due

Sunday, 22 June, 11:59 PM

Course event

General Instructions

1. Each group is expected to submit a Jupyter notebook (.ipynb) file with output displayed for each computation. Add brief meaningful comments to the modules wherever required in the notebook.
2. No extension of the deadline.
3. Submissions using other Python IDEs will not be considered for grading.
4. Input Dataset of 10 files in different formats (.docx,.pdf, .csv etc.) should be chosen according to the domain in which the question is based upon. Ensure each file has enough content (minimum 100–200 words) to reflect realistic search and correction operations.

Note: You may obtain relevant datasets from Kaggle or other open-source platforms such as UCI Machine Learning Repository, Google Dataset Search, or data.gov, depending on the domain selected (e.g., healthcare, legal, academic, or e-commerce).

5. Display the outputs of each computation / function even if not explicitly mentioned in the question. Marks will be awarded for outputs shown in each step.
6. Inverted index should be displayed for all programs in sorted order.

Question : (10 marks)

Title: *Comparative Analysis of Standard vs. Weighted Edit Distance for Isolated Word Correction in Legal Document Retrieval*

Domain:

Legal Information Retrieval (e.g., Westlaw, LexisNexis)

Objective:

Accurate retrieval in legal search systems depends on correct spellings of complex legal terms. Traditional spell correction techniques like **standard Levenshtein Edit Distance** treat all edit operations (insertion, deletion, substitution) equally. However, in the legal domain—where some spelling errors are more likely than others—a **Weighted Edit Distance** model may provide better correction accuracy by assigning custom costs to each operation. This case study aims to compare the effectiveness of these two models in correcting isolated misspelled legal terms.

Develop a Python-based application that:

1. **Constructs a legal term dictionary** containing at least 100 valid legal terms.(1)
2. **Implements both**:

- Standard Levenshtein Edit Distance, and
- A **Weighted Edit Distance algorithm**

Accepts a **misspelled query word** from the user and returns the closest valid legal term using both algorithms. (4)

3. **Compares and explains the differences** in corrections suggested by both models. (2)
4. Tests the system on **at least 5 real-world misspellings** of legal terms (e.g., "plentiff", "jurispudence", "habeas corpas", etc.), and analyzes:

- Accuracy of corrections
- Number of operations and associated cost
- Situations where weighted edit distance improves correction (3)

Information Retrieval (AIMLCZG537/DSECLZG537)(S2-24)

Add submission

Import or export calendars