Assignment 3 – Grammar Correction and Legal Case Similarity Retrieval

Course: Natural Language Processing
Instructor: Muhammad Owais Idrees
Total Marks: 100
Submission Format: Jupyter Notebook / Python Script + Report (PDF)
Submission Deadline: 16-Nov-2025

Assignment Overview

This assignment is a continuation of **Assignment 2**, where you developed a Neural Language Model and generated both extractive and abstractive summaries of legal judgments.

In **Assignment 3**, you will extend that work in two directions:

1. **Refine the abstractive summaries** from Assignment 2 by applying grammar correction and linguistic improvement techniques.
2. **Develop a Legal Case Similarity Retrieval System** that identifies the most semantically similar cases based on learned embeddings.

The focus of this task is on **language refinement**, **semantic understanding**, and **information retrieval** within legal texts.

Part 1 – Grammar Correction and Summary Refinement (50 Marks)

Objective

Improve the grammatical accuracy and readability of the abstractive summaries created in Assignment 2.

Your refined summaries should be grammatically correct, coherent, and concise, without altering the meaning of the original sentences.

Task Description

1. **Input**
   Use the **abstractive summaries** generated in Assignment 2 as input. Each summary corresponds to a specific case number.
2. **Grammar Correction**
   Apply **automatic grammar correction** techniques using natural language processing tools or rule-based approaches.
   You may choose any of the following strategies:
   - Use grammar correction tools or libraries.
   - Apply rule-based correction using **Part-of-Speech (POS) tagging** or **dependency parsing**.
   - Combine both methods to achieve improved grammatical accuracy.

3. **POS-Based Refinement**
    - Analyze the grammatical structure of sentences.
    - Ensure **subject–verb agreement**, proper **tense usage**, and correct **noun and article placement**.
    - Remove unnecessary repetition or redundant words.
4. **Output Format**
   Present your final results in a comparison table showing the original and corrected summaries.

   Example Structure:

Case No.	Original Abstractive Summary	Corrected Summary
C001	The court dismiss the guilty defendant case.	The court dismissed the guilty defendant's case.
C002	The appeal file against decision.	The appeal was filed against the decision.

5. **Reflection and Analysis**
   Provide a brief written explanation (1–2 paragraphs) describing your grammar correction approach, challenges encountered, and how the corrected summaries improved clarity.

Deliverables for Part 1

- A clearly formatted table showing original and corrected summaries.
- Description of the grammar correction process and reasoning behind chosen methods.
- Short discussion of improvements observed in language quality.

Part 2 – Legal Case Similarity Retrieval (50 Marks)

Objective

Develop a retrieval system that identifies **Top-K most similar legal cases** based on the document embeddings learned in Assignment 2.

This task will demonstrate your understanding of **semantic similarity** and the ability of embeddings to represent the meaning of entire legal documents.

Task Description

1. **Input**
   Each case is represented by:
   - A **unique case number**, and
   - Its **document embedding vector**, obtained from Assignment 2 by averaging all sentence embeddings within the case text.
2. **Query Case**
   You will be provided with one or more **new cases** (identified by case numbers).
   For each new case:
   - Compute its document embedding using the same method as before.
   - Use this embedding as a query to find similar cases.
3. **Similarity Computation**
   - Compute the **cosine similarity** between the query case embedding and all other case embeddings in your dataset.
   - The similarity score should reflect how semantically close the two cases are.
4. **Retrieving Top-K Cases**
   - Rank all cases based on their similarity scores.
   - Select the **Top-K most similar cases** (e.g., K = 3 or 5).
   - Record their **case numbers** and **similarity scores**.
5. **Output Format**
   Present your final results in a structured table.

   Example Structure:

Query Case No.	Similar Case No.	Similarity Score
C010	C002	0.95
C010	C001	0.92
C010	C005	0.89

6. **Interpretation**
   In your report, briefly explain:
   - Why the retrieved cases are likely to be similar (e.g., similar legal themes, verdicts, or terminology).
   - How embeddings helped capture semantic similarity in the context of legal texts.
   - Any observations or limitations of your approach.

Deliverables for Part 2

- A working similarity retrieval system that outputs Top-K similar case numbers.
- A results table with case numbers and similarity scores.
- A short written explanation (1–2 paragraphs) interpreting the retrieval results.

Final Deliverables

Task	Deliverable	Description
1	Grammar Correction & Refinement	Corrected summaries with explanation of the process
2	Case Similarity Retrieval	System that retrieves and displays Top-K similar case numbers
3	Report	PDF summarizing your methods, observations, and results

---

Optional Bonus (+10 Marks)

Students can earn up to 10 additional marks by completing one or more of the following:

- Visualize case embeddings using **t-SNE** or **PCA** to show how similar cases cluster together.
- Evaluate retrieval accuracy using manually annotated similar cases.
- Integrate both parts into a simple command-line or web-based interface for querying cases.

Evaluation Criteria

Criteria	Marks
Grammar Correction Implementation	25
Quality of Corrected Summaries	25
Case Similarity Retrieval Logic	30
Code Quality and Documentation	10
Report and Analysis	10
Total	**100**

Submission Instructions

1. Submit a single **ZIP file** named as:
   Assignment3_<YourName>_<RollNumber>.zip
2. The ZIP file should include:
   - Your Python or Jupyter Notebook file
   - A Json or CSV file containing corrected summaries
   - A results file or table for case similarity retrieval
   - A concise report (PDF) explaining methods and observations
3. Upload your submission to the LMS or send via email before the deadline.

💬 **Important Notes**

- Reuse the embeddings and model components you built in Assignment 2.
- Ensure all outputs are clearly labeled and formatted for readability.
- Focus on interpretability — clearly explain how your model identifies similar cases and improves summary quality.
- Avoid using large pre-trained models unless explicitly approved.