Project 1 – Natural Language Processing

Course: This is an applied assignment covering material and platforms taught throughout the course.

Submission: Only pairs of students may submit (!). Submissions by groups of more or fewer than 2 will incur a significant grade penalty.

Project Objective

The aim is to equip you with practical tools for implementing NLP tasks. The field of NLP evolves very rapidly and relies on powerful frameworks—e.g., Google, OpenAI, Meta—used across major companies.

Note: Unlike two years ago, when the main challenge was to write code and translate theory into implementation, this year the emphasis in grading will shift. We expect detailed explanations, your thought processes, and a description of experiments (successful or not). A submission with perfect code but no rationale will receive only partial credit.

Project Structure

1. 1. Part A (30% of grade)

- Dataset: COVID-19 text classification from Kaggle (50,000 labeled records for classification).
- Exploratory Data Analysis (EDA):
  - Visualizations and statistical analyses
  - Creative preprocessing (tokenization, normalization, augmentation)
- Documentation: Explain your reasoning, approaches tried (and failed), and why.

2. 2. Part B (70% of grade)

- Modeling: Implement at least two pretrained models (e.g., from Hugging Face) using PyTorch, applying both:
  1. Full-code fine-tuning
  2. Hugging Face Trainer API fine-tuning
- Hyperparameter tuning: Use Optuna for both models.
- Logging & analysis: Provide detailed W&B dashboards, code comments, and explanations.

3. Write-up: Draft a 6-page academic-style paper covering:
   - Introduction and motivation
   - Methodology and rationale
   - Experimental setup and hyperparameters
   - Results, comparisons, and analysis (include graphs, tables)
   - Conclusions and future work

Deliverables and Submission

Code repository (Git) containing Part A EDA scripts and Part B training scripts, model checkpoints, and a README.

Paper (PDF).

Ensure everything runs end-to-end via git clone and clear instructions; code that fails or lacks documentation will not be graded.

Grading Breakdown

Component	Weight
Part A (EDA, code runs without errors)	20%
Part B code quality, architecture, and correctness	20%
Exploratory analysis (visuals, insights)	20%
Experimental rigor, choice justification, depth	15%
Reporting (W&B, documentation, organized experiments)	25%
Writing the paper (academic style, clarity)	10%
Bonus: Extra content beyond requirements	10%

Deadlines

Part A & B (entire project): August 5

Part B detailed deadline: Announced later

Project presentation date: To be scheduled