Assignment 1 (30)

CS F429: Natural Language Processing

Deadline: 8th September 2025, 04:00 AM

1 Problem Statement: Language Modelling for Text (20)

1.1 Task Description

You are required to implement the following 3 language modelling techniques from scratch:

- N-gram
- Glove
- Fasttext

1.2 Datasets

You will train models on datasets of varying sizes and record your **observations**. Save each trained model in order to test them during the demos.

There are two datasets (Hindi and English), each split into three subsets based on size:

- **Small** – 2,500 samples
- **Medium** – 15,000 samples
- **Large** – 30,000 samples

The cleaned and pre-sized datasets can be downloaded here: https://drive.google.com/drive/folders/1u1nhwKEmurgoTP6eFX7T91TxNZFVk6zI?usp=sharing

More about the datasets:

- The Hindi dataset is a news article dataset originally from https://www.kaggle.com/datasets/shivamtaneja2304/inshorts-dataset-hindi
- The English dataset is movie and plot dataset originally from https://www.kaggle.com/datasets/jrobischon/wikipedia-movie-plots

1.3 Evaluation

Use both intrinsic and extrinsic evaluation to measure the performance of your models. Implement "perplexity" to perform intrinsic evaluation. To perform extrinsic evaluation, you will conduct a downstream text classification task using the learnt vectors/ngrams on both the Hindi and English datasets. The Hindi dataset is labeled with News Categories, while the English dataset is labeled with Genres. These labels serve as the targets for classification. You can use the dedicated test file for this task: https://drive.google.com/drive/folders/1u1nhwKEmurgoTP6eFX7T91TxNZFVk6zI?usp=sharing. You will compare performance across the three embedding models using classification metrics such as accuracy, precision, recall, and F1-score.

2 Bonus Problem (Additional 20% more)

GloVe requires a large amount of memory because it builds and stores a dense co-occurrence matrix of words. To handle this, implement **batch processing** where the co-occurrence matrix is divided into smaller chunks that fit into memory. This will prevent requiring the entire matrix in memory at once.

3 Report

You should document your findings in a report. Report should contain the following sections

- **Problem Statement**
- **Dataset Statistics**
- **Experimental Setup**
- **Results**: Report perplexity, precision, recall, and F1 scores for each of the 6 variations in the data set and 3 language models.
- **Discussion**: Explain the results. Include examples where the models perform well, perform poorly, or show neutral results. Compare the models with each other, and explain why one outperforms another on a given dataset while not on others.
- **Time Taken**: Report time taken to train and test the models on each of the datasets. Include your system configuration. Explain the times.

4 What should you submit?

Files to be submitted.

1. A zip file with code

2. A report.pdf written in latex. Report should document all findings from the experiments.

Checklist for zip file.

1. Code files, preferably in Python.
2. **requirements.txt**: You may include extra libraries for implementation. To be able to run your submission, all your dependencies should be written to a 'requirements.txt' file and submitted.

5 Grading Principles

Demos will be conducted for evaluation. Scores will be awarded based on:

1. Whether or not your code runs into error (30% penalty)
2. If your code/report is found to be plagiarized (plagiarism policy)
3. Report structure and content
4. Implementation correctness and viva