N-grams and Misspelling Correction
ELL884 Assignment 1

Deadline - 6:33 PM, 14th February, 2025

1 Overview

In this assignment you will build a robust n-gram language model and leverage it to implement a probabilistic misspelling error correction system. You will work on several interconnected tasks, including implementing a basic n-gram class, various smoothing mechanisms, and a system to correct misspellings using a noisy-channel model. All hyperparameters will be specified in a configuration file. The assignment also involves extensive evaluation of the language model performance.

2 Tasks

Task 1: Implement a Basic N-Gram Class

Complete the NGramBase class in the `ngram.py` file that builds n-gram frequency counts from a text corpus. Please implement the required functions. You can add more functions but the compulsory ones must be implemented.

Task 2: Implement Smoothing Mechanisms

In the file `smoothing_classes.py`, implement different smoothing techniques including:

- No smoothing (raw MLE estimation).
- Add-k smoothing
- Stupid Backoff
- Good Turing
- Interpolation
- Knesner-Neys

Note: Do not change the provided function signatures so that autograding can operate correctly.

Task 3: Implement Misspelling Error Correction Using N-Grams
Using your n-gram models, implement a probabilistic misspelling error correction system. Your system should:

- Accept input text containing misspellings.
- Generate candidate corrections based on n-gram probabilities.
- Utilize a probabilistic (noisy-channel) model that combines the language model likelihood with an error model to select the best candidate correction.

In-Depth Details:
For the misspelling error correction component, your approach should integrate:

(a) **Error Model:** Estimate the probability that a given word is misspelled and quantify the likelihood of specific typo-to-correction transitions.

(b) **Language Model:** Use the n-gram probabilities (with appropriate smoothing) to assess the fluency and context-fit of candidate corrections.

(c) **Combination:** Merge these probabilities (e.g., via a noisy-channel model) to select the most likely correction in context.

You are encouraged to experiment with different candidate generation and ranking strategies. Evaluate your system using the provided test corpus for misspellings.

3 Config.py

Specify All Hyperparameters in `config.py`
Edit the contents of (`config.py`) that defines all hyperparameters used in your implementation, including:

- N-gram order(s).
- Smoothing parameters (e.g., the value of k in Add-k smoothing, discount values for other methods).
- Parameters for the misspelling error model.

4 Dataset

The assignment provides the following datasets:

- **Training Corpora:** Two separate corpora that you must pre-process (tokenization, normalization, handling rare words, etc.) to build your n-gram models.
- **Test Corpus for Misspellings:** A sample test corpus containing misspellings to evaluate your error correction system. Each line of the dataset contains the data seperated as " $< CORRECT\_TEXT >$ && $< INCORRECT\_TEXT >$ "

5 Submission Details

Your submission should include the following:

1. **All Code Files:** Submit all source files. Do **not** change the function signatures provided in the template; this is required to support the auto-grading system.
2. **Report:** A comprehensive report that includes:
   - Performance analysis of your n-gram models across different values of N and smoothing techniques. Include perplexity measurements and other relevant metrics.
   - Examples of text generated by your model.
   - A detailed explanation of your probabilistic misspelling error correction model, including the error model, candidate generation strategy, and quantitative and qualitative results on the sample test corpus.

6 Evaluation

Your work will be evaluated based on:

- Correctness and efficiency of the implemented algorithms.
- The clarity, structure, and modularity of your code.
- The depth of analysis in your report, including performance metrics and generative examples.
- Testing of your misspelling error correction code on private misspelling dataset.