Gebze Technical University
Department of Computer Engineering
CSE 654 / 484
Fall 2024

Homework 1
Due date: Oct 31th 2024

Statistical Language Models Using N-Grams for Syllables and Characters

In this homework, you will develop two statistical language models for Turkish: one using N-grams of syllables and one using N-grams of characters. You will explore their performance by calculating N-gram probabilities, evaluating perplexity, and generating random sentences. The final report should include detailed analysis, tables, and your conclusions based on the results of both models.

Part 1: Data Preparation (20 points)

1. Download the Turkish Wikipedia Dump:

- Download the dataset from Kaggle and prepare it for processing. (5 points)

2. Text Preprocessing:

- Syllable-based Model: Segment the words into syllables. You can use existing tools or write your own.

- Character-based Model: Normalize the text by converting all letters to lowercase and optionally converting Turkish characters (e.g., ş -> s, ğ -> g). (10 points)

3. Data Splitting:

- Use 95% of the dataset for training and 5% for testing in both models. (5 points)

Part 2: N-Gram Calculation (30 points)

1. Build N-Gram Tables:

- For both syllable-based and character-based models, calculate 1-Gram, 2-Gram, and 3-Gram tables using 95% of the dataset.

- Implement an efficient storage method (e.g., dictionaries, hash tables) since the N-gram tables will be sparse. (15 points)

2. Smoothing:

- Apply Good-Turing (GT) smoothing to handle unseen N-grams in both models. (10 points)

3. Perplexity Calculation:

- Compute the perplexity for 1-Gram, 2-Gram, and 3-Gram models in both the syllable-based and character-based models using the remaining 5% of the dataset. (5 points)

Part 3: Random Sentence Generation (20 points)

1. Generate Random Sentences:

- For each model (syllable-based and character-based), generate random sentences using 1-Gram, 2-Gram, and 3-Gram models. At each step, choose one of the top 5 N-grams randomly. (10 points)

2. Discussion of Generated Sentences:

- In your report, include and analyze the sentences generated by each model. Discuss how the quality and fluency vary based on the N-gram level (1-Gram, 2-Gram, 3-Gram) and between the syllable-based and character-based models. (10 points)

Part 4: Report (30 points)

Your report is crucial for earning points. Failure to submit a meaningful report will result in zero points for this assignment.

1. Introduction:

- Briefly explain the concept of N-gram language models and the differences between syllable-based and character-based models. Outline the objectives of the homework. (5 points)

2. Design and Implementation:

- Provide a detailed explanation of your code design, including how you split the dataset, calculated N-grams, applied smoothing, and generated random sentences.

- Describe each function used in your implementation, specifying their inputs, outputs, and functionality. (10 points)

3. Results and Tables:

- Present your results in a table format. Include the perplexity values for 1-Gram, 2-Gram, and 3-Gram models for both syllable-based and character-based models.

- Include another table with sample random sentences generated by both models (at least two sentences for each N-gram size and model). (5 points)

4. Analysis and Conclusion:

- Analyze the performance of the two models based on the perplexity values. Discuss any trade-offs between using syllables versus characters for N-grams.

- Comment on the quality of the generated sentences and compare the performance across different N-gram levels and models.

- Conclude which model and N-gram size is more suitable for modeling Turkish, based on your findings. (10 points)

5. Table on LLM Usage:

- Include a table that indicates which parts of your submission were generated or assisted by ChatGPT (or any other large language model) and which parts were original. (You should have this part regardless!)

Notes:

1. Use logarithms when multiplying probabilities in the chain rule to avoid numerical underflow.

2. Ensure all letters are converted to lowercase, and optionally convert Turkish characters to their English equivalents (e.g., ş -> s, ğ -> g).

3. Include punctuation marks (sentence-end symbols and spaces) in your N-gram models as needed.

Submission and Rules:

Prepare your report in PDF format and submit it along with your code (either as a ZIP file or a notebook) via the submission platform.

You may use any programming language for the implementation. If you use libraries for N-gram calculation, state which ones you used and explain how.

We encourage the use of tools like ChatGPT for assistance, but your homework should reflect more original work than automated parts.

Remember: No report, no points! Similarly, no code, no points!