Week 1 Assignment [4 Points]

Generative AI

Saarland University – Winter Semester 2024/25

Submission due: 21 October 2024 by 6pm CET

genai-w24-tutors
genai-w24-tutors@mpi-sws.org

1 Reading Assignment

This week's reading assignment comprises of the following material:

(R.1) Chapters 3.1, 3.4, and 3.5 from the book by Jurafsky and Martin [1]. It is available freely online at the following link: `https://web.stanford.edu/~jurafsky/slp3/ed3bookaug20_2024.pdf`.

2 Exercise Assignment

This week's exercise assignment is based on the reading material provided above and comprises of the following questions. These questions should be answered in the PDF submission file – see instructions in Section 4.

(E.1) Express an upper bound on the number of parameters required for an n-gram model as a function of the order n of the model and the vocabulary size V.

(E.2) For a vocabulary size of $V = 50,000$, what is an upper bound on the number of parameters required for $n = 1$ and $n = 3$ gram models?

3 Implementation Assignment

For this week's implementation assignment, you will get to see how we can generate sentences using n-gram language models trained on a specific corpus. We have provided the following files in `week1_assignment.zip`:

- `lm_ngram.py`: An implementation of an n-gram model that trains the model from a given corpus and then generates sentences.
- `processed_berkeley_restaurant.txt` and `processed_shakespeare.txt`: Two different corpora that can be used as input in `lm_ngram.py` to train n-gram models. The text in these files is already processed using the basic rules mentioned in the reading material.

As part of this assignment, you will simply execute the provided code by modifying the input parameters and don't have to implement any new code. More concretely, in the `__main__` function of `lm_ngram.py`, you will vary two parameters about the value of n and `corpus_file_path`. When the provided code is executed, it first trains the language model and then generates five sentences from the trained language model.

(I.1) Generate five sentences for $n = 1$ and corpus `processed_berkeley_restaurant.txt`. You can put the generated sentences in a .txt file with name `week1_I1.txt`. This `.txt` file should be included as part of the ZIP submission file – see instructions in Section 4.

(I.2) Generate five sentences for $n = 3$ and corpus `processed_berkeley_restaurant.txt`. You can put the generated sentences in a .txt file with name `week1_I2.txt`. This `.txt` file should be included as part of the ZIP submission file – see instructions in Section 4.

(I.3) Generate five sentences for $n = 1$ and corpus `processed_shakespeare.txt`. You can put the generated sentences in a .txt file with name `week1_I3.txt`. This `.txt` file should be included as part of the ZIP submission file – see instructions in Section 4.

(I.4) Generate five sentences for $n = 3$ and corpus `processed_shakespeare.txt`. You can put the generated sentences in a .txt file with name `week1_I4.txt`. This `.txt` file should be included as part of the ZIP submission file – see instructions in Section 4.

(I.5) Provide a qualitative comparison of how the generated sentences vary in terms of quality and characteristics for different values of n and input corpus. You can answer this question in a few sentences and no detailed analysis is required. This should be answered in the PDF submission file – see instructions in Section 4.

4 Submission Instructions

Please submit the following files using your personal upload link:

- **<Lastname>_<Matriculation>_week1.pdf**. This PDF file will report answers to the following questions: E.1, E.2, I.5. The PDF file should be generated in LaTeX using NeurIPS 2024 style files; see further formatting instructions in Section 5.1. The PDF file size should **not exceed** 1**mb**. Please use the following naming convention: Replace <Lastname> and <Matriculation> with your respective Lastname and Matriculation number (e.g., Singla_1234567_week1.pdf).
- **<Lastname>_<Matriculation>_week1.zip**. This ZIP file will report answers to the following questions: I1, I2, I3, I4. Importantly, the ZIP file should unzip to a folder named **<Lastname>_<Matriculation>_week1**. Inside this folder, include four `.txt` files containing the generated sentences. The ZIP file size should **not exceed** 1**mb**. Replace <Lastname> and <Matriculation> with your respective Lastname and Matriculation number (e.g., Singla_1234567_week1.zip).

5 Technical Instructions

5.1 Instructions for Preparing PDFs

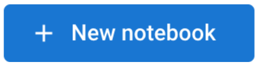
Please use the following formatting instructions for the PDF submission:

- The PDF file should be generated in LaTeX using NeurIPS 2024 style files. We have provided the required style files and a sample LaTeX file in `week1_latex_sample.zip`:
  - `neurips_2024.sty` is a style file downloaded from `https://media.neurips.cc/Conferences/NeurIPS2024/Styles.zip`.
  - `sample.tex` is a sample LaTeX file to prepare the PDF.
  - `references.bib` is a bibliography file.
  - `sample.pdf` is a PDF file generated from the above LaTeX file.
- Overleaf is a popular online LaTeX editor that could be used for preparing the PDFs (link: `https://www.overleaf.com/`).
- Inside the .tex source, use appropriate week number in the title field and use your Lastname / Matriculation number / email in the author's field.
- For ease of reading, we recommend that you create different sections (e.g., one per exercise).

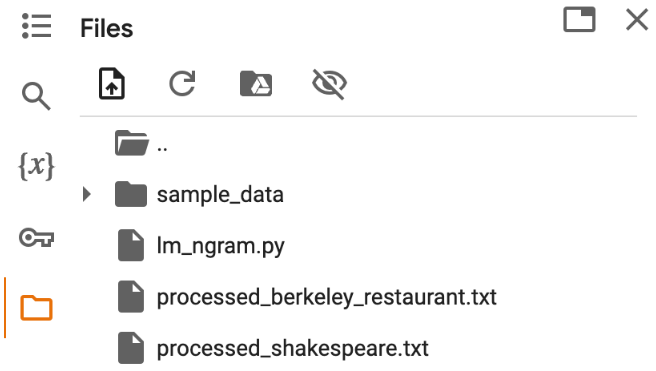
5.2 Instructions for Implementation

Throughout the course, we will have implementation assignments that will require more intensive compute resources, including GPUs. Hence, we will explore using online computing services like Google Colab (link: `https://colab.research.google.com/`). In particular, Google Colab also offers limited free compute resources that would be sufficient for the implementation assignments in this course. Below, we provide instructions on how you can use Google Colab to execute the code as part of this week's implementation assignment.

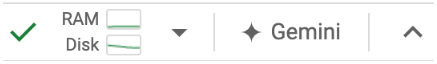
(1) Open Google Colab (https://colab.research.google.com/) in your browser. When you are logged in with google account, you will see a popup to click and create "New notebook".



(2) On the left panel, click on the directory icon and upload three files (`lm_ngram.py`, `processed_berkeley_restaurant.txt`, and `processed_shakespeare.txt`).

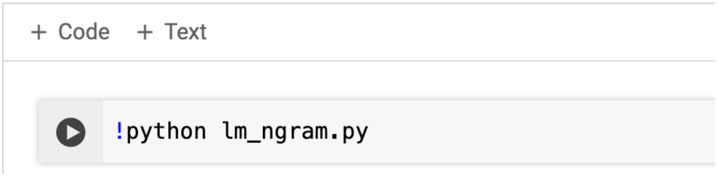


(3) Ensure that you are connected to a runtime session. It would normally happen automatically when you upload files, assuming compute resources are available at that time.



(4) Double-click on the code file `lm_ngram.py` to open the editor and edit the parameters in the `__main__` function.

(5) Enter `!python lm_ngram.py` in the workspace and click run icon.



```
!python lm_ngram.py
```

(6) The output from executed code will appear in the workspace.

(7) In case you make changes to your files within the editor, ensure that these files are saved elsewhere. The uploaded files will be deleted automatically once this runtime session is terminated.

References

[1] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3rd edition, 2024. Online manuscript released August 20, 2024; https://web.stanford.edu/~jurafsky/slp3/.

Based on template from https://media.neurips.cc/Conferences/NeurIPS2024/Styles.zip