Advanced Deep Learning
Assignment 3

May 8, 2024

1 Instructions

1.1 General Instructions

Important rules and notes:

- All assignments in the course are either group or individual.
- You are not allowed to collaborate with anyone on the assignment, and you are not allowed to communicate your solutions to other students.
- You must not ask for help from anyone except the teachers and TAs on the course.
- On the other hand, we encourage you to use the exercise classes and the Absalon forum to get help. The exercise sessions exist to help you with the assignments, and you are welcome to ask any questions related to the teaching material and the assignments on the forum.
- If your solution contains material from other sources than the assignment text, you must cite the source of the material and any changes you have made. This also applies to material from textbooks, Absalon, etc.
- If your solution uses methods or notation that are not used in the course material, you must specify where you found the method or notation.
- If you are in doubt about plagiarism or citation rules, please ask the teachers or TAs.
- Please be very observant of these rules. We do not want any plagiarism cases, both for your and our sake.

1.2 Assignment 3 Instructions

- This is a group assignment, and the deadline for this assignment is May 28, 2024, 22:00 You must submit your solution electronically via the Absalon home page.

- This assignment has three parts: a multiple-choice test, RNNs, and zero-shot prompting.
- For the second part, you need to include a maximum 2-page **research** report in PDF format. Please use the provided LaTeX template. Do not zip the PDF file. For this assignment, do not submit your code.
- For the third part, you need to include a maximum 1-page report in PDF format. Do not zip the PDF file, **and** submit your final results to a Google Form (see Part C).

2 Assignment 3

2.1 Part A. Multiple Choice Test

Fill out the Google Form multiple-choice test at: ADL24 Assignment 3 Part 1

2.2 Part B. Recurrent Neural Networks

Write a **research report** of **maximum 2 pages** following the next instructions.

You will study the capacity of (Elman/Vanilla RNNs and LSTMs by training them on fractions of a formal language and evaluating how well it generalizes. The formal language is:

$$a^n b^n c^n \tag{1}$$

- Train one RNN and one LSTM on members and non-members of these languages of length 20 or less, sampled from the space of strings in $a^n b^n c^n$. The samples could include:

  <**aaabc,0**> <**aaabbbccc,1**> <**aaaaaaabbbc,0**>

  for example, where 0 means the string is not a member of the language, 1 means that it is a member. You decide the dataset size, but remember to mention this in your text. You should also report the average length of the samples. Note that if you find it hard to learn from this very skewed distribution, you are more than welcome to balance the sampling somewhat, sampling more balanced subsets of random members of each class.

- It is important that you are as fair to the two architectures as possible. You should do some (not complete) hyper-parameter search - in some way - by splitting your training data - in some way. Your report should describe what you did in both cases, e.g., how you searched for what hyper-parameters, and how you split the data (and why).

- Evaluate the selected RNN and LSTM models on members of the language of length 21 or longer. Report the average length of the test sequences. When evaluating the models you should produce a classification accuracy or F1 score and justify your choice. The main result, however, should be a graph, plotting performance (accuracy, or F1) over sentence length. This should take the form of a coordinate system with lengths 21 and

upwards (say to 100) along the x-axis and performance values (between 0 and 1) along the y-axis.

- Write a summary of the results and a comparative analysis of the models' performances, then draw some conclusions from your analysis and findings.

2.3 Part C. Zero-shot Prompting

Write a maximum **one-page report** following the next instructions.

You will test the zero-shot performance of LLMs on predicting whether a tweet belongs to a parody account or a real politician account. The tweets are:

> *It's the #GimmeFive challenge, presidential style.*

shared by the real politician account @BarackObama

> *It's up to you, America, do you want a repeat of the last four years, or four years staggeringly worse than the last four years?*

shared by the parody account @SecretMitRomney

1. **Prompts**. Create 2 prompts for each tweet by designing two different templates: Template A should be formulated as a question, and Template B should be formulated as an instruction. Remember to make use of delimiters, for instance, to indicate the start and end of the tweet (context). You can test a sample of initial prompts at https://chat.openai.com/ or https://www.llama2.ai/. The ultimate goal is that the model response is written in a way that you can extract the assigned label, i.e., *real* or *parody*. This can be done by using a specific character or token to delimit the label, or by adding an instruction so that the model **only** responds with the label. If we wanted to label a large number of instances, this would facilitate the evaluation rather than having to find the answer within the reasoning steps that models are often trained to provide. Tip: look at common templates for models, such as T0. The report should include a list of initial prompts (maximum 5 per template plus the selected two templates).

2. **Evaluation**. Go to Chatbot Arena and test each prompt by applying each template to each tweet. You can either choose the models randomly - Arena (battle) or manually choose the two models from the list of options - Arena (side-by-side). Include your choice in your report.

3. **Results**. Report your results, including the model names and characteristics of the models, such as the number of parameters and whether they are open-source or not, any observations you can make from these results, a screenshot for each prompt (4 in total, i.e., two per tweet showing the results of both models), and the winner of the battle (or tie). Tip: You can save a screenshot by clicking on *Save*.

4. **Google Form**. Finally, include your results in this form: ADL24 Assignment 3 Part 3.