Assignment 2: Transformers and Computational Linguistics

CS 577: Natural Language Processing • Fall 2025

Due: 11/21/2025 @ 11:59PM EST

Part A (36 Points, Yunxin): Mini Transformer

Please go to the Brightspace course and locate the link under *Content/CS577 Assignment 2*, or you can access this part of the homework here. It will take you to Vocareum, an online coding and grading environment. Inside Vocareum, go to Part A, and there will be a notebook that you need to fill in. This part of the assignment will let you code a clean Transformer architecture with minimal dependencies. Note that this part is ***auto graded***. You will see your grades immediately after you hit the `Submit` button. You DO NOT need to do anything after you hit the `submit` button. You have unlimited attempts for (re)submission.

Part B (18 Points, Yunxin): In-context Learning

In this part, you will experiment with in-context learning, a practical prompting approach for Large Language Models, and with other aspects of prompt engineering. All handouts for this part are also in Vocareum (Brightspace - Content - Vocareum - Part B, and click Colab on the top left or here). You can also use the free GPU in Google Colab, or get Google Colab Pro free for 1 year. This part is ***manually graded***. You need to include your response in a PDF along with the answers to Parts C and D, and submit it to Gradescope. You also need to click the `submit` button in Vocareum. Your grade in this part will be dependent on the quality of your report and the quality of your code (e.g., whether it is readable, whether it supports your argument in the report). Results that cannot be included in the report may be included in the notebook (e.g., raw results). In other words, you should treat your notebook more as a richer format of the report rather than just the code.

Part C (30 Points, Yunxin): Computational Linguistics

1. Write context-free grammars for the following languages (with $n \geq 0$, $m \geq 0$, $k \geq 0$)
   (a) $L = \{a^n b^m c^k : n = m \text{ or } m \leq k\}$
   (b) $L = \{a^n b^m c^k : n + 2m = k\}$

2. Write context-free grammars for the *parenthesis matching* problem: Parentheses may be nested or appear side-by-side (e.g., "(())()"). In each question, define your terminals and non-terminals, as well as the production rules.
   (a) The string only contains the characters "(" or ")".
   (b) The string may contain the characters "(", ")", "{", "}", "[", "]".
3. Show that the following grammar is ambiguous by giving a string with two distinct parse trees:

$$S \to aSb \mid SS \mid \varepsilon$$

   Give a verbal description of the language generated by the grammar.

Part D (16 Points, Nathaniel): Discussion Questions

1. As discussed in the lecture, certain evaluation metrics can make LLMs appear to undergo sudden, "emergent" jumps in capability as model size increases. For the task of multi-hop reasoning, choose two evaluation metrics that could scale differently with model size, and explain what properties of each metric might make one appear to show a sudden gain while the other scales smoothly.
2. For each task below, assume you fine-tune an LLM with vanilla policy gradient.
   (a) Design a reasonable reward function. Is the reward sparse or dense? Is it discrete or continuous?
   (b) In 1–2 sentences, justify your choice and predict whether or not the variance of the gradient estimator will be low or high under that design.

   Tasks: Four-Digit Addition, Story Summarization, and Sentiment Analysis.

Submission Instructions

Part A is autograded via Vocareum. For Parts B, C, and D, please compile your report into a single PDF and submit it through Gradescope. Refer to each part for additional details.