11-411/11-611 Homework Assignment 4: QA Agent

NLP Teaching Staff

Due: December 4, 2025

1 Introduction

Our friend Lexi loves the idea of traveling the world. But between classes, meetings, and naps (essential for llamas), she hasn't had much chance to visit places beyond the Cranberry Melon University campuses. Instead, she explores Google Maps to travel "in theory" — and eventually becomes obsessed with GeoGuesser.

One day, Lexi dramatically slams her laptop shut. "I always lose at GeoGuesser." she complains. "It's unfair! I know every bit of knowledge I can acquire online, from Wikipedia to Reddit posts, yet this game is full of random hints and lucky guesses!" She goes on to rant how she relies on language, context, and reasoning, while the game rewards spotting random details in blurry photos.

"If only there were a version based on text, I'd be the GOAT!"

Although slightly skeptical, you realize that you can help. Inspired by what you have learned in the NLP class, you decide to gather the data for a text-based GeoGuesser and create an agent that can take on Lexi.

In this homework, you will be (1) gathering data to curate the GeoGuesser dataset together, (2) building LLM agents using LangChain, (3) incorporating advanced techniques to boost QA performance, and (4) analyzing your approach in depth. There are three types of questions, each with different characteristics, and you will have the chance to stretch your creative muscles for the type that excites you the most. You will also get to join the friendly competition on the Gradescope leaderboard.

🏆 marks tasks with potential bonus points. The exact plan will depend on the final situation since we want to be as generous as possible and make sure the bonus points are shared fairly. This final homework is more like an open-ended project. Have fun!

2 Learning Objectives

- Understand the importance of data by contributing to a shared dataset and collecting high-quality examples.
- Apply PyTorch and the Hugging Face Transformers library to fine-tune, perform inference, and evaluate pretrained models.
- Practice prompt engineering with LLMs, including zero-shot and ICL scenarios.
- Gain hands-on experience with LangChain to build LLM agents and integrate advanced components such as RAG, tool calls, and web search.
- Clearly report experimental methodology and results, and conduct detailed analysis of model performance and behavior.

3 Step 1: Collecting Text-based GeoGuesser Dataset (10%) (🏆5%)

There are three types of questions:

🗺️ **Global Trekker:** Given a short descriptive paragraph about a cultural event, natural landmark, or historical site, can you guess which country it is from? Moreover, can you identify the city if implied?
All paragraphs are in **English**. The scoring weights for (country, city) are (0.8, 0.2).

🍽️ **Culinary Detective:** Given a list of ingredients and descriptions of a dish, can you identify which country and region (*north, south, east, west, or central*) it is from?
All ingredients and descriptions are in **English** with transliterated terms (e.g., *sushi*).
The scoring weights for (country, region) are (0.6, 0.4).

👄 **Lingua Locale:** Given a sentence from an official website (e.g., a government page), can you determine which country's website it comes from?
This task is **multilingual**. You earn 1 points for correct country; 0.4 points for overlapped official language.

3.1 Contributing to the Dataset

One of the best ways to understand the dataset is to join the curation process! You will **contribute at least one entry per question type**. In general, submit the text, labels and additional information that may help you improve agents' performance. Detailed formats and guidelines are provided in the Google Forms. The instructors will review and also contribute entries.

We aim for a **diverse and high-quality dataset**, since poor data can directly affect your experiments and analysis and make your life harder! To encourage participation, **bonus points** will be awarded to top contributors for quality and diversity. (Note: The more you contribute, the more influence you have on the dataset!)

For the **required submission**, please use this Google Form. For **extra entries** afterwards, please use this second Google Form, which is more flexible and helps us track who has made the required submission. We encourage you to finish your initial submissions **as early as possible** so that the instructors can perform thorough quality checks.

3.2 Timeline of Data Release

There are two data splits: public and private. **Do not manually label the private set**!

Public set: This is the **validation set** where you can perform error analysis and gain insight to improve your system design. In addition to the text and label, it also includes reference links and hints for analysis and discussion purposes.

Private set: This is the **test set**, provided with only the text and empty label fields. It will be used for the Gradescope leaderboard. **Hints and reference information are not included in this set**, as they should not be used during inference.

The timeline is as follows (in ETC timezone, on days with class):

1. Crowdsourcing Data: Nov 13 (Thu) – Nov 18 (Tue)
2. Public set release: Nov 20 (Thu)
3. Private set release: Nov 25 (Tue)

4 Step 2: Baseline Implementation (30%)

In this part, you will learn about the structure of LangChain and how it is used in our scenario. The `HW4_baseline.ipynb` notebook requires only a few lines of code to complete! Many concepts build on our previous prompt engineering exercises (e.g., API setup, answer formatting, and evaluation), so feel free to revisit them as needed. As in previous exercises, this homework requires OpenAI API credits and/or GPUs. You can run it on Colab, AWS, a remote server, or your own machine – just keep an eye on your credit usage.

Each input consists of one or more text fields and the output is a tuple of strings containing the requested information. Your task is to implement the **answer extraction functions** to remove excessive content and **refine the prompt** for better formatting. Since this is a short-form QA with a single answer, we use **soft match** and **exact match** as the evaluation metrics for different answer types. You will receive full credit for a question type if your final score exceeds the baseline score.

5 Step 3: Improve Upon Baseline (🏆5%)

After running the basic pipeline, **choose one question type as your main task**, which will be the focus of your final report.

We encourage you to explore substantial improvements beyond prompt engineering that reflect your own insights. Possible directions include but are not limited to retrieval (RAG with collected data, knowledge-graph–based retrieval, web search), reasoning (iterative refinement, ReAct, chain-of-thought), and tool augmentation (translator, finetuned transformer).

The only restriction is that **you may not use web search agents or RAG with crawled data to improve 👄 Lingua Locale**, as that constitutes direct web retrieval.

Each question type has its own leaderboard, with **bonus points for the top 3 submissions in each**. If you are motivated, you can also improve multiple tasks to compete on the leaderboard!

6 Step 4: Report write-up (60%)

As the main component of this homework, you will write a report to share your observations, hypotheses, experiments, and analyses. Think of it as a **mini research paper**. To facilitate grading, please **start each section on a new page**.

6.1 Observing baseline performance (15%)

🗺️ **Global Trekker:** Identify an example where the challenge arises from the data itself (e.g., ambiguous descriptions or incorrect labels).

🍽️ **Culinary Detective:** Find a case where the RAG system might be misleading, explain why, and evaluate the zero-shot performance without RAG.

👄 **Lingua Locale:** Within the same language, which country or countries does the model usually predict? Can you hypothesize why this occurs?

6.2 Proposed method to improve (15%)

State the question type you chose, your hypothesis about performance bottlenecks, and your experimental design. Support your reasoning with evidence or prior work (citations to papers or technical reports are preferred; informal sources like Reddit or tweets are acceptable). Provide enough detail for someone to re-implement your approach without major effort or lengthy explanations.

6.3 Experimental Results (10%)

Present your results in a table like Table 1 that includes **score and efficiency** metrics. List **at least two approaches beyond the baseline**, such as preliminary trials, ablations, or variants explored. Report public-set performance for all entries and private-set results for those submitted to Gradescope.

Include a figure that visualizes your results (e.g., confusion matrix, prediction distribution) with clear legends and captions. Avoid redundant plots that repeat the same data as in the table.

Approach	Public set			Private set		
	Score	Time	Cost	Score	Time	Cost
Baseline						
Your Main Approach						
Your Other Approach						
...						

Table 1: Example results table.

6.4 In-depth analysis (20%)

Present further analysis, observations, and conclusions. For example, are there trade-offs between accuracy and efficiency? Why does an approach perform better than others? What types of error do the models make and what might the evaluation be missing? If you were building this system in the real world, what concerns would you have?

Include appropriate graphics, plots, or data to support your analysis. There is no single correct answer, as we encourage you to explore your own questions and perspectives. Your submission will be evaluated on the **depth, breadth, and clarity** of both written discussion and visualizations (tables, figures, captions). You may refer to the published papers linked in the course slides for examples of strong analysis and presentation.

6.5 Appendix

Include supplementary materials that clarify the above sections, such as prompt or data processing examples. You may lose points if explanations in the main sections are unclear and necessary details are missing from the appendix. Organize this section clearly and concisely so that your work is easily understood.

7 Gradescope Submission

Prediction for leaderboard: Submit two text file to Gradescope Coding Part: `public.txt` for public set, and `private.txt` for private set. Each line should correspond to an answer; refer to the baseline code for details on how to generate this file.

Report: Submit your report, named `HW4_{andrewid}.pdf`, to Gradescope Written Part. Remember to select the appropriate sections; points will be deducted if you forget to do so.

Code for reference: Submit your notebook, named `HW4_{andrewid}.ipynb`, along with any other required dependencies, to Gradescope Coding Part.