Part A

Data 6 Fall 2025 Final Project: Part A

Project Introduction

In this final project, you will explore the interdisciplinary domain of **computational social science**. You will study how large language models can support qualitative coding, a social science research method that involves assigning categorical labels to open-ended text data.

In this two-part project, you will perform the following tasks:

- Analyze a complex text dataset;
- Perform a text classification task with two strategies: hand-coding (Part A) and AI-assisted (Part B). The latter will be completed by prompt engineering with Gemini.
- Analyze the performance of these two strategies by computing Cohen's Kappa scores and confusion matrices;
- Analyze and create CSV files; and
- Write a (two-part) project report; and
- Write supporting code for your report in (two) Jupyter notebooks.

This project builds on the tasks described in the following study:

> Caleb Ziems, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, Diyi Yang; Can Large Language Models Transform Computational Social Science?. Computational Linguistics 2024; 50 (1): 237–291. doi: https://doi.org/10.1162/coli_a_00502

Project Logistics, Timeline, and Deliverables

Project Part A is due **Thursday, December 4 at 8pm.** This is a **partnered project**. Project Part B is released separately and will have similar logistics.

Note for three-person groups: Additional guidelines for three-person groups are listed with each question part.

You should refer to this Google Doc for all project tasks.

- For most tasks, you will write responses in this Google Doc.
- For other tasks, you will write code in the project Jupyter notebook on DataHub.
- For some tasks, you will write code in the project Jupyter notebook AND share/paste code snippets, visualizations, or code output into this Google Doc.

You will submit two deliverables for Project Part A:

1. A written report PDF (a filled-out copy of this Google Doc)
2. A ZIP file of code and data files

Submission will be on Gradescope; instructions are at the end of this document.

Setup instructions

Because this project is partnered, we recommend the following workflows:

- **Written report:** Make a copy of this Google document (File → Make a copy). Share with each other, and work using Google Docs's collaborative editing features.
- **Code/data files:**
    - (recommended) Work together for coding at all times. "Pair program" on the same machine and same DataHub instance.
        - One person is the "driver," writing code into their notebook.
        - One person is the "navigator," telling the driver how and what to write.
        - As you complete each part/finish each coding session, the navigator copies the code cell into their own notebook.
    - If you absolutely must work remotely:
        - (preferred) Send the Jupyter notebook back and forth to one another, and work on DataHub. The best way is the most straightforward way:
            - File --> Download your notebook. Send the `.ipynb` file to your partner.
            - Your partner then uploads the `.ipynb` to DataHub. Go to datahub.berkeley.edu. Navigate to the final project folder in `materials-fa25/project_final`. Upload the `.ipynb` file into this directory, overwriting the original.
        - You can work on a notebook via Google Colab (which has some collaborative features). This requires additional setup, including installing additional packages to support autograding. Stop by office hours and consult a TA if you want to pursue this option.

Part 1: Pick Your Dataset

Qualitative coding involves assigning categorical labels to open-ended data to answer a specific research question. In this project, we look at how to assign categorical labels to small snippets of text associated with a single user or speaker, which is what Ziems et al. calls an *utterance-level* classification task.

The document lists a subset of the tasks in Ziems et al. From this list, choose the text classification task that your group wants to study for the remainder of this project. **Three-person groups**: Not all tasks are available to you; read this document for the "Additional Notes" section to see whether you can do this task.

We **strongly recommend** you explore these data files before committing to your task. You will find dataset CSV files for each file on DataHub. Each file is all prefixed with the dataset `shortname`, e.g., `emotions.csv`. Explore files using the DataHub File Explorer.

How to navigate files in DataHub: File Explorer in JupyterLab view

You will find it useful to use the JupyterLab view on DataHub to view datasets. This is a more complex view than the simple Jupyter Notebook view you have been using this semester and allows you to access all of your files on DataHub. You can open CSVs in this view, download files, and upload files.

How to navigate to the file explorer on DataHub:

1. Save your changes in your notebook.
2. In the top-right, on the menu bar, click on the dropdown "Open with…" and select "NBClassic" or "JupyterLab" This should open your notebook in a new tab.
3. If you clicked "NBClassic", in the new tab of your notebook, click the Jupyter logo in the top left of your screen. This will navigate you to "JupyterLab." If a pop-up appears saying you have unsaved changes, continue to leave the page (this is because you should still have your original tab with your notebook in it.)
4. Go to the sidebar and click the folder icon. This is the **file explorer**.
5. In the file explorer, navigate to the appropriate folder with this assignment. It should be something similar to `materials-fa25/project/project_final`.

To navigate between folders on DataHub, click the corresponding folder in the file path.

Download files from DataHub:

- Single file: Right click on the file → "Download"
- Multiple files: Shift-click to select your files, Right-click on a file → "Download"
- Folder: Right click on the folder → "Download as an Archive"

Upload files onto DataHub:

- Click and drag a file from your computer onto the File Explorer. The file will then be uploaded to whatever folder is currently available in the File Explorer.

Question 1: Task Shortname (2 points)

Below, indicate which task you will explore for the remainder of this project. Replace the text below with the corresponding `shortname` in Data 6 Fall 2025 CSS Tasks.

Write your response here, replacing this text.

Part 2: Exploratory Data Analysis (EDA)

Question 2: Contextual Analysis (8 points)

This section will help set the stage for understanding your dataset's potential limitations and biases. Read the source paper, read the data README, etc., and explore the **context** of your dataset.

Here are a few guiding questions:

1. How was the dataset created?
2. What biases or perspectives might be present in the data, and how could they influence your subsequent analysis?
3. Who was responsible for applying the labels (i.e., "codes"), and how reliable are these labels?

Below, share your contextual analysis as a 1-3 paragraph response.

Write your response here, replacing this text.

Question 3: Explore the CSV (3 points)

In the next part of this project, you will hand-code part of this data, i.e., assign categorical labels associated with your task. **Labels**, also referred to as **codes**, are the categorical labels assigned to the open-ended text responses of each record in your data.

Depending on which task you chose, you may find that the original paper specified a multitude of labels, stages of labels, or tasks for your dataset. We provide a clean version of the data in `<shortname>.csv`, where `<shortname>` is the shortname of your CSS task.

- Some datasets have a README file. Read this README if it exists; it will help describe the data provided to you.
- **Do not look** at `<shortname>_train.csv` and `<shortname>_val_uncoded.csv`; these are used for the next part.

Question 3a: CSV Shape (1 point)

Report the number of records (number of rows) and number of columns in your provided CSV. You can write code as needed, or analyze using Google Spreadsheets or another CSV viewer. Share these two statistics below.

Write your response here, replacing this text

Question 3b: CSV Columns (2 points)

In the accompanying Jupyter Notebook project on DataHub, explore the subset of data we have provided you in `<shortname>.csv`.
Below, write a short description of each column of your selected dataset. We recommend you look at how datasets have been described to you throughout this course (e.g., on homework assignments or exams).

Formatting: Format your response as a table listing the variable name, a short description, and (for categorical variables) the set of possible categories. Example:

Variable	**Description**	**Category**
ID	A unique integer describing…	
…		
Label	…some description…	1 (Funny) 0 (Not funny)
Secondary Label	…some description…	Speaker Listener Observer

Write your response here, formatted as a table similar to the above

Question 4: Distribution of "Human Gold Labels" (5 points, coding)

The **human gold labels** are the labels accompanying your dataset, generated via the original authors' research. These human gold labels will be considered the "gold standard" of the two classification strategies you will explore in this project (hand-coding and AI-assisted).

Question 4a: Visualization (3 points)

Write code that visualizes the distribution of labels in your dataset. Depending on how many sets of labels you have, you may need to produce multiple visualizations (especially for three-person groups).

- [Update 12/1] Note: the definition of "label" may vary depending on what your task is. For example, for the FLUTE dataset you likely want to plot the distribution of simile, metaphor, idiom, sarcasm.8
- Go to the accompanying Jupyter Notebook, scroll to this Question, and write your code that produces this visualization.
    - You should use the `datascience` library and associated visualization methods in this course.
    - Your analysis should be on `<shortname>.csv`, where `<shortname>` is the shortname of your CSS task. **Do not look** at `<shortname>_train.csv` and `<shortname>_val_uncoded.csv`; these are used for the next part.
- Include screenshot(s) of the resulting visualization(s) in this Google Doc. Do not screenshot the code.

Replace this text with screenshot(s) of visualization(s)

Question 4b: Interpretation (2 points)

Based on your visualizations above, comment on the distribution of labels (i.e., codes). Is the distribution skewed, or is there relatively equal representation across labels? Limit your response to 1-2 sentences.

Your response here, replacing this text

Question 5: Distribution of Text (9 points, coding)

Question 5a: Top 20 Most Frequent Words (6 points)

Write code that reports the top 20 most frequent words in your dataset, across all sentences, **ignoring stop words**.

Hints/recommendations:

- This question is intentionally open-ended.
    - [Update 12/1] You should consider what your task is to choose which columns of data to include in your "dataset." (you can peek ahead to the different CSVs in Part 3).
- We've imported the `stopwords` package for you.
- A reasonable solution will use some sort of dictionary.
    - Look at the bag of words model that you constructed in a previous project. Translate that approach to this application.
        - [Update 12/1] Hint: Questions 1 - 3 in Project 2 do what you want, but they start with a list of all words of strings. In this case, you have a column of strings. We strongly recommend you copy some of this code into this part.
        - [Update 12/1] We have given you some code to get going, also adapted from Project 2. See the `split_into_words` demo code.
    - If you are looking for a challenge, we recommend you look at the CountVectorizer library in the sklearn documentation.
- It is alright to use Generative AI for this question, but you will be expected to understand your approach. Depending on your approach, during grading we may ask you follow-up questions about your code.

List the top 20 words in this document (ignoring stop words), sorted by most common first.

Replace this text with your list

Question 5b: Interpretation (3 points)

How might these most common words inform your process for hand-coding labels, if at all? What words seem particularly meaningful for your specific task? Limit your response to 1-2 sentences.

Your response here, replacing this text

Part 3: Hand-coding

In this part, you will try hand-coding labels for your task and analyzing comparing your performance with your partner. **Hand-coding** is the process of manually assigning labels to your text data.

Note that your data is *already* coded with the "gold standard labels," so you may think this task is trivial. However, in practice you will rarely have these labels for your data! By having the gold standard to compare against ***but by avoiding referencing them during the coding process*** (i.e., labeling process), you are effectively simulating a situation where you may not have access to labeled data.

Side note: Labeled data is often used for a class of machine learning tasks called supervised learning. Ask us if you're curious about this field.

Here is the strategy outlined in the following questions:

- With your partner, study 20 examples with human gold labels and agree upon a strategy for hand-coding a subset of the data.
- Individually hand-code (i.e., label/categorize) the same 30 records, following the coding strategy you agreed upon.
- Report Cohen's Kappa, a measure of inter-rater agreement, for your two sets of hand-coded codes.
- Evaluate and analyze how reliable you were as hand-coders.

Question 6: Develop Coding Strategy (8 points)

With your partner, study the 20 examples in `<shortname>_train.csv`, where `<shortname>` is the shortname of your CSS task. These records have been labeled for you using the labels you identified in the previous part. Still with your partner, use these 20 examples to discuss and agree upon a "coding strategy" for hand-coding examples. Document this strategy below.

Do not look at any CSV files other than the 20 examples provided to construct your coding strategy. We are trying to simulate a situation where you do not have access to the rest of the data, so that you can more accurately hand-code in the next section. Peeking at the rest of the data at this point poses a threat to the validity of your hand-coding approach!

Formatting: You can format your response as a bulleted list, a decision tree diagram, a set of paragraphs, some examples, etc.—whatever is easier for the two of you to reference. Limit your response to at most half a page.

- *Your response here, replacing this text*

Question 7: Hand-code Your Data (5 points)

Individually code (i.e., label/categorize) the 30 records in `<shortname>_val_uncoded.csv`. Each partner should hand-code the same 30 records using the coding strategy you agreed upon earlier.

To avoid threats to validity:

- No peeking at the gold standard labels!!!
- No peeking at each others' work!

Once the two of you are done, save your code labels in a new CSV labeled `<shortname>_val_human.csv`, where `<shortname>` is the shortname of your CSS task.

Formatting: Your `<shortname>_val_human.csv` file should have the 30 records from `<shortname>_val_uncoded.csv` with the following columns:

- All original columns from `<shortname>_val_uncoded.csv`
- Two additional columns `human_A` and `human_B` corresponding to the two sets of hand-coded labels that you and your partner individually generated.

To complete this part, make sure that your CSV file is uploaded in the correct DataHub directory and follows the naming convention provided. The accompanying Jupyter Notebook provides some starter code to save this file; you can also use Google Sheets and upload a file.

There is no written response needed for this question, but you must format your `<shortname>_val_human.csv` file correctly and submit it with your code.

Question 8: Compute Inter-Reliability Agreement (2 points, coding)

Report Cohen's Kappa (κ) as a measure of the agreement between you and your partner's hand-coded labels. Additionally, use the Data 6 Notes to report the agreement level based on your computed κ.

Notes:

- In the accompanying Jupyter Notebook, we have provided some starter code to compute Cohen's Kappa. It uses the `cohen_kappa_score` function from the `sklearn` library. The code also assumes that you have labeled the CSV file in the previous part correctly.
    - [Update 12/2]: The starter code assumed columns `"humanA"` and `"humanB"`, but you should edit these to `"human_A"` and `"human_B"`, respectively.
    - [Update 12/2]: The starter code assumes your CSV is uploaded to the same directory as your notebook. Depending on where you uploaded the file, you may need to edit the CSV location, to, say `datasets/<filename>.csv`.
- It is okay to report low κ/agreement level in this part. This part is graded on completion.
- We are **not** comparing your codes to the human gold standard; we will do so in Part B of the assignment.

Complete this part by writing code to compute Cohen's Kappa in the accompanying Jupyter Notebook, then reporting κ **and** the agreement level below.

Replace this text:

- *Cohen's Kappa, κ=<your response here>*
- *Agreement level: <one of no agreement, poor, fair, moderate, good, near-perfect agreement>*

Question 9: Reflect on Hand-Coding Strategy (8 points)

Reflect on your experience hand-coding. Was it challenging? Did you and your partner disagree on certain labels? on certain examples? Discuss any difficulties and insights gained from the process.

Limit your response to at most 1 page.

Your response here, replacing this text

Done!

Submission instructions

A complete submission for Part A involves submitting to two Gradescope portals.

Both portals require a **group submission**. This means that only one person should submit on Gradescope by ADDING the second person as a group member for each portal's submission. Please see the Gradescope instructions for adding group members to a submission.

1. **Final Project Part A (Written)**: In your copy of this Google Document, export this document as a PDF (File → Download → PDF Document (.pdf)), then submit this document on Gradescope.
2. **Final Project Part A (Coding)**: Export a zip of your work so far.
    - Navigate to the DataHub JupyterLab view (with the file explorer)
    - On DataHub, navigate to `materials-fa25/project/project_final`. Export a zip of this folder by right-clicking on `project_final` and selecting "Download as an Archive."
    - Submit this zip to Gradescope.
    - **Make sure that the files you submit include**:
        - `project_final_partA.ipynb`, your code notebook
        - `<shortname>_val_human.csv`, which is the CSV you created in Part A.3.