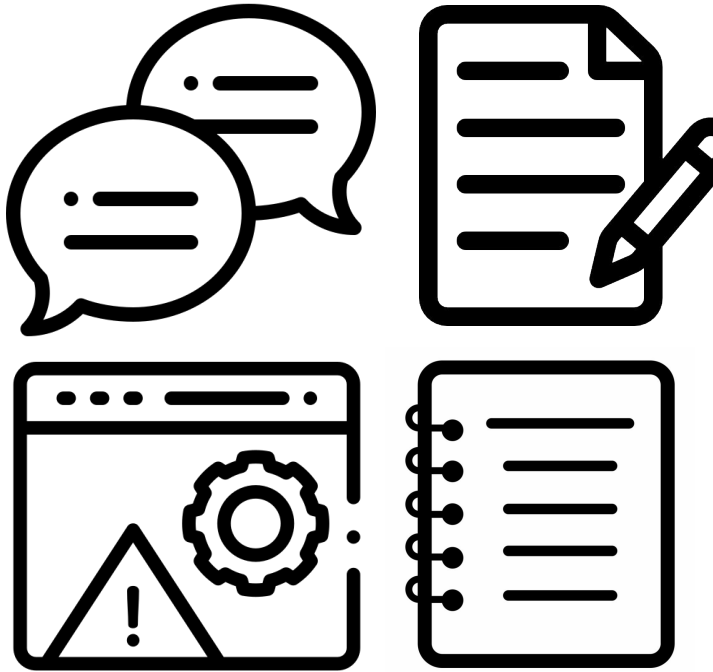
Improving writing instruction Through LLM Scripting

- Starting points
- Creating self-grading assignments
- Organizing assignment results
- Checking grammatical accuracy



Starting points

- Think critically about AI (i.e., applied statistics)
- Large Language Models are revolutionary
  - human languages
  - programming languages

Starting points

- As a result, educators should:
  - make their own programs (i.e., python scripts)
  - streamline their workload
  - focus on direct interaction with students

Starting points

- Ipsative assessment:
  - assessment based on improvement
  - rather than criteria or peer comparison
- Recommendations:
  - extensive assistance from LLMs
  - careful integration with achievement criteria

Zero shot,
Chain of thought

- Zero shot:
  - single prompt
  - No background information
- Chain of thought:
  - series of prompts
  - receiving/developing background information

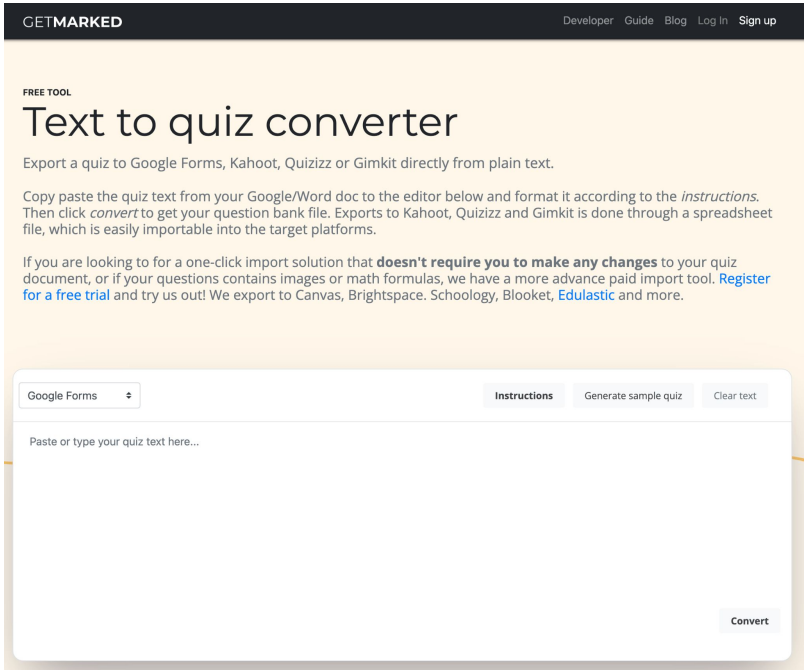
Creating
self-grading assignments

- Why Google Forms?
  - can insert questions from PDFs or ChatGPT
  - self-grading for certain data types
  - variety of possible data types
    - text, multiple-choice, audio, video
  - data output to spreadsheets
  - data output in real-time

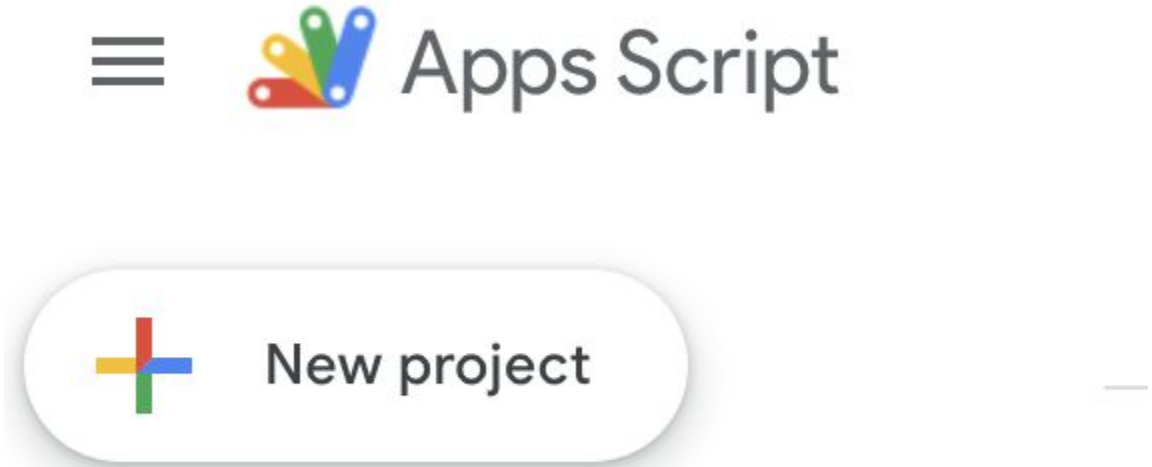
Creating
self-grading assignments

- Please follow the instructions in this link to create a Google form using GetMarked: https://forms.gle/Sb7TcJDp1tkJ7qqc9

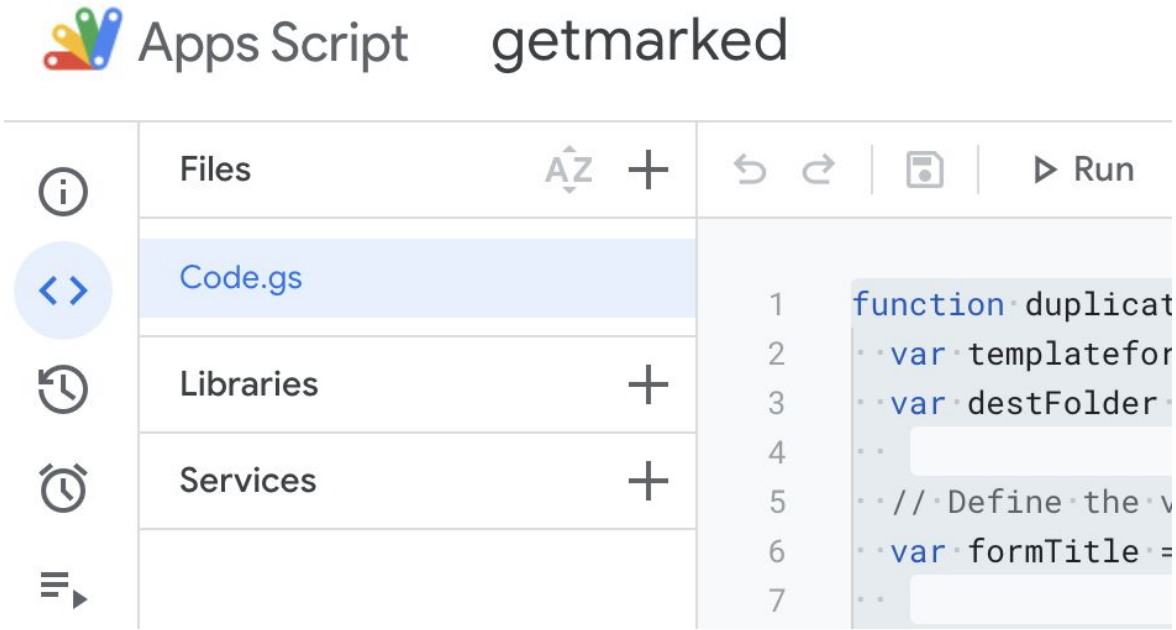
Creating
self-grading assignments



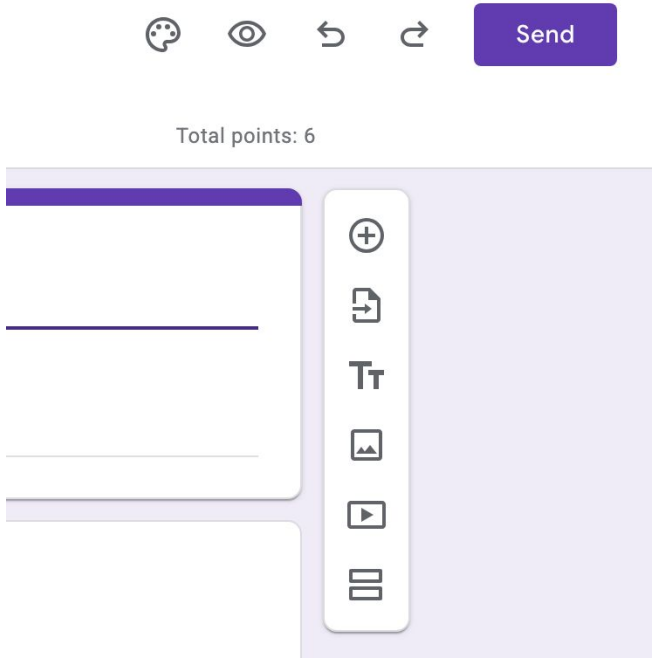
Creating
self-grading assignments



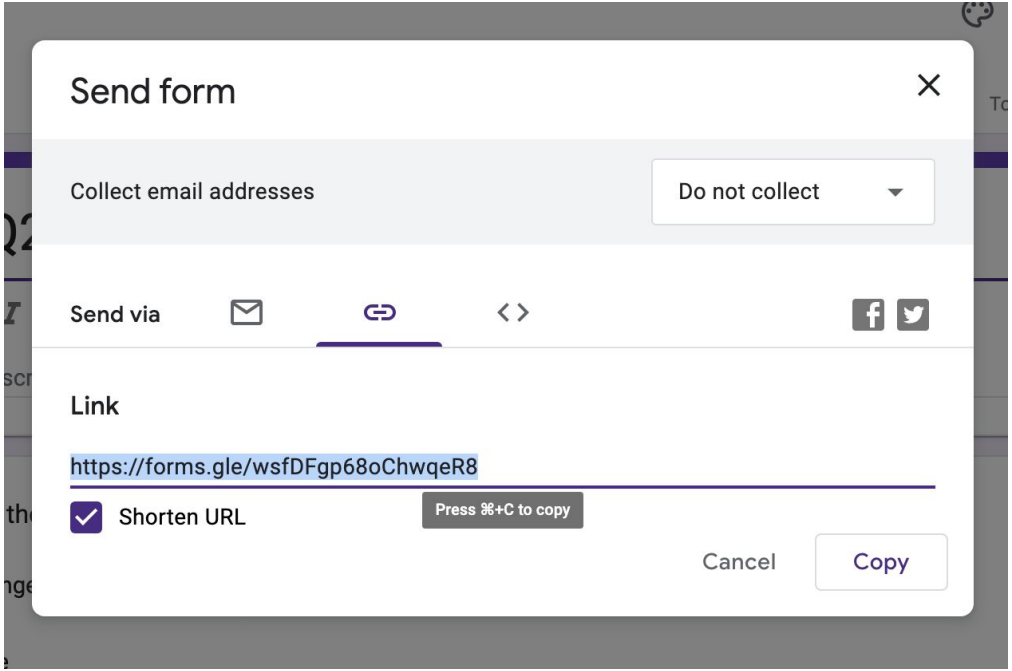
Creating
self-grading assignments



Creating
self-grading assignments



Creating
self-grading assignments



Creating
self-grading assignments

- Please follow the instructions in this link to create a Google form using GetMarked: https://forms.gle/Sb7TcJDp1tkJ7qqc9

Organizing assignment results

- Why Google Sheets?
  - receive data in real-time
  - sophisticated data processing
    - attendance, assignments, grades
    - plagiarism detection, content analysis
  - easy output to third-party resources
  - easy input from third-party resources

Organizing assignment results

- First columns:
  - each student's name, ID number, e-mail address, & class section
- Later columns
  - attendance, assignments, scores, text of assignments, text of ChatGPT feedback
  - participation grades, statistics (e.g. average number of grammar errors)

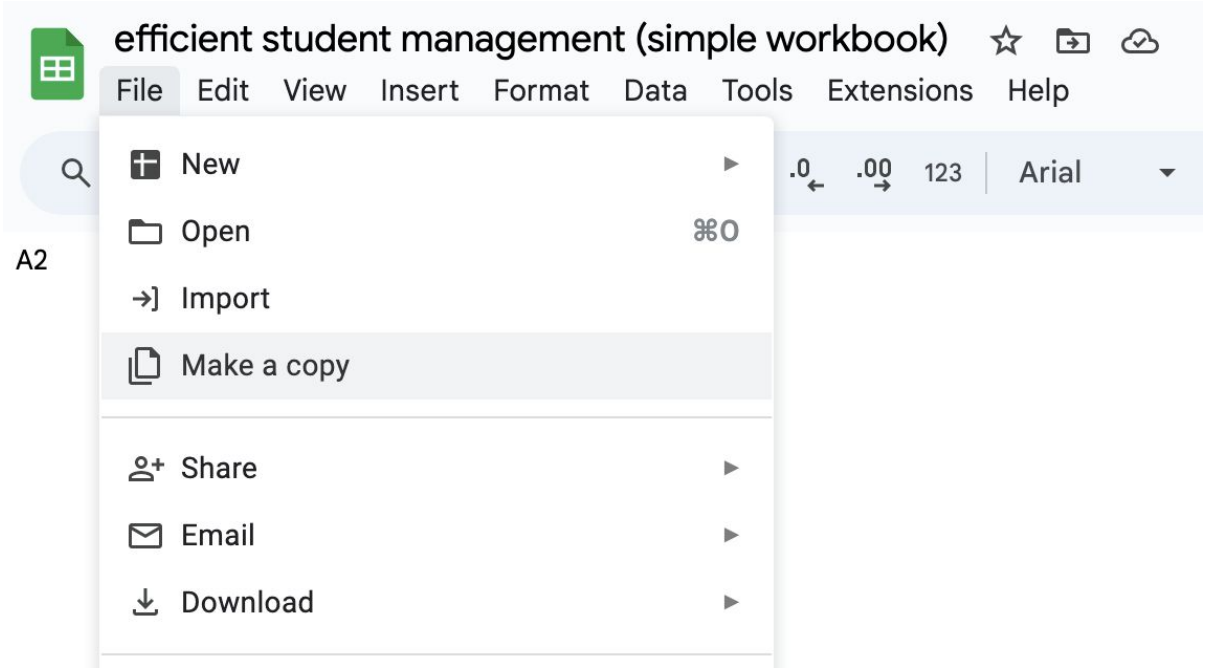
Organizing assignment results

- Later sheets:
  - Google forms output for each assignment
  - datasets for ChatGPT (& third-party resources)
- Third party resources:
  - TSV output (File>Download>Tab Separated Values)
  - run python script
  - paste output in appropriate place

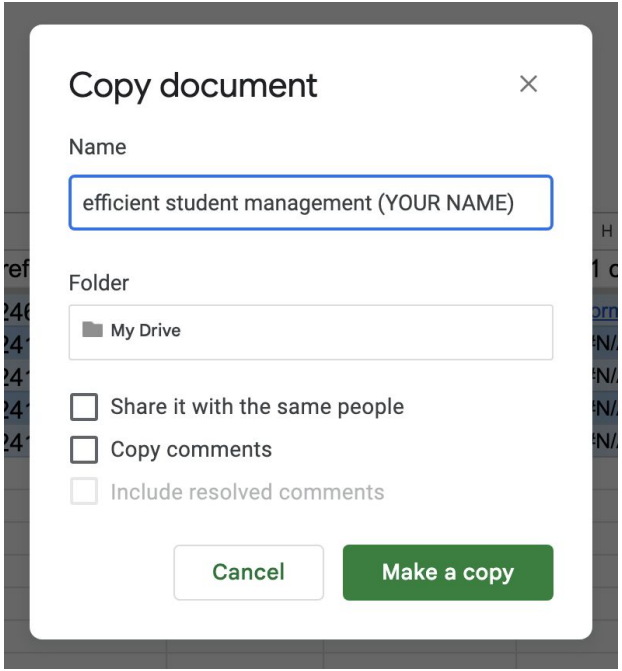
Organizing assignment results

- Please follow the instructions in this link to create a Google Sheet: https://forms.gle/z5pwfzABdT5X7sns9

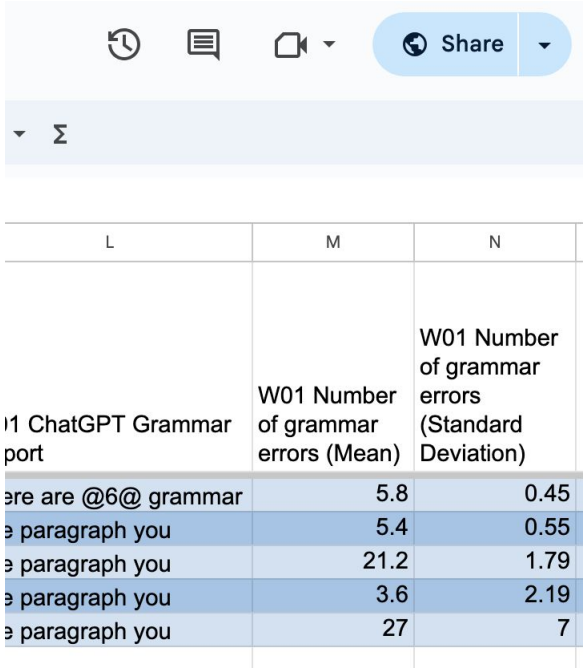
Organizing assignment results



Organizing assignment results

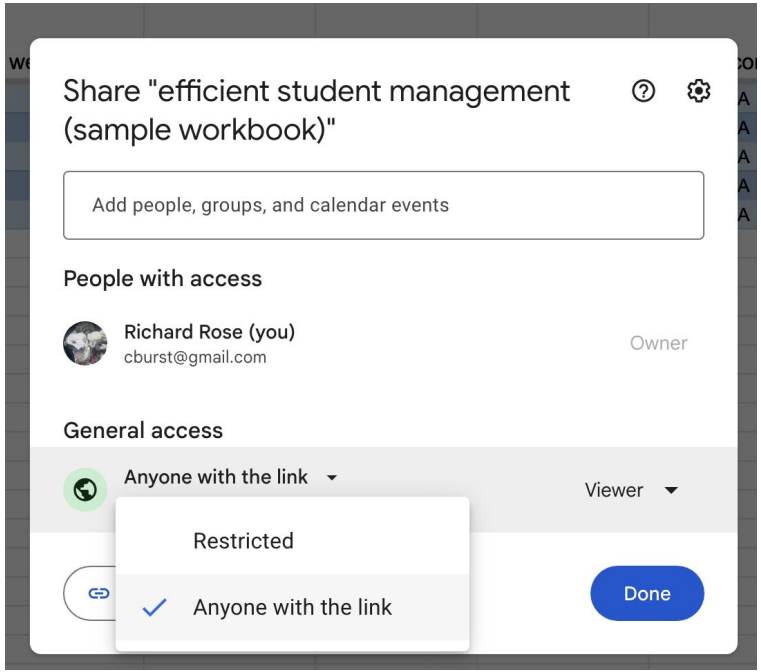


Organizing assignment results

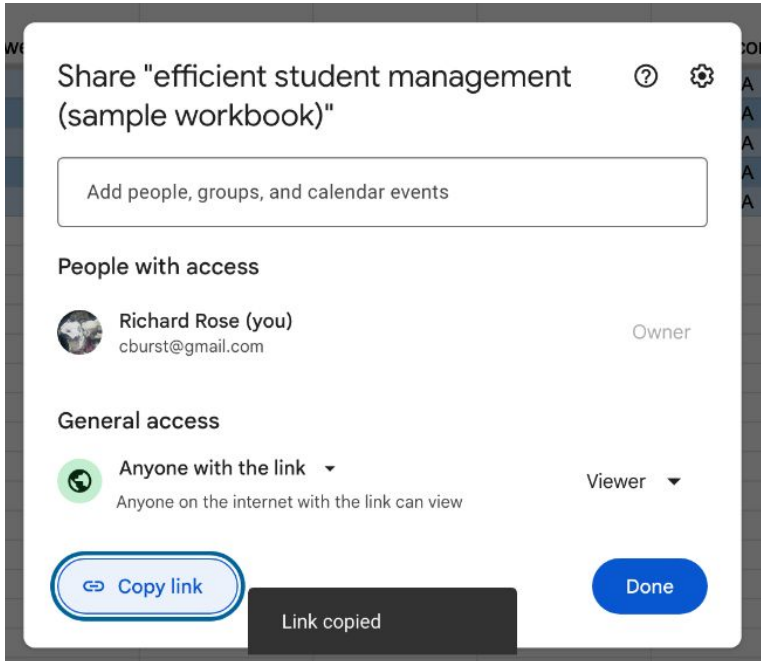


L	M	N
1 ChatGPT Grammar port	W01 Number of grammar errors (Mean)	W01 Number of grammar errors (Standard Deviation)
ere are @6@ grammar	5.8	0.45
e paragraph you	5.4	0.55
e paragraph you	21.2	1.79
e paragraph you	3.6	2.19
e paragraph you	27	7

Organizing assignment results



Organizing assignment results



Organizing assignment results

- Please follow the instructions in this link to create a Google Sheet: https://forms.gle/z5pwfzABdT5X7sns9

Checking grammatical accuracy

- Download and use scripts shared HERE
  - Python (install version 3.11)
  - OpenAI API Key (place in env variable)
  - GPT-CLI & dependencies
- Follow along at your own pace

Checking grammatical accuracy

- Why Python?
  - easy to write (with help from ChatGPT)
  - easy to read (mostly plain English)
- Why the command line interface?
  - conduct 1000s of ChatGPT queries in just a few seconds

Checking grammatical accuracy

- get OpenAI API Key & copy to clipboard
  (https://platform.openai.com/api-keys)
  Create a Key, Setup Billing, Give them $5
- Windows/Mac instructions:
  Go to LLMscripting Releases
- Tutorials:
  Windows Tutorial - English
  Windows Tutorial - Korean
  Mac Tutorial - English
  Mac Tutorial - Korean

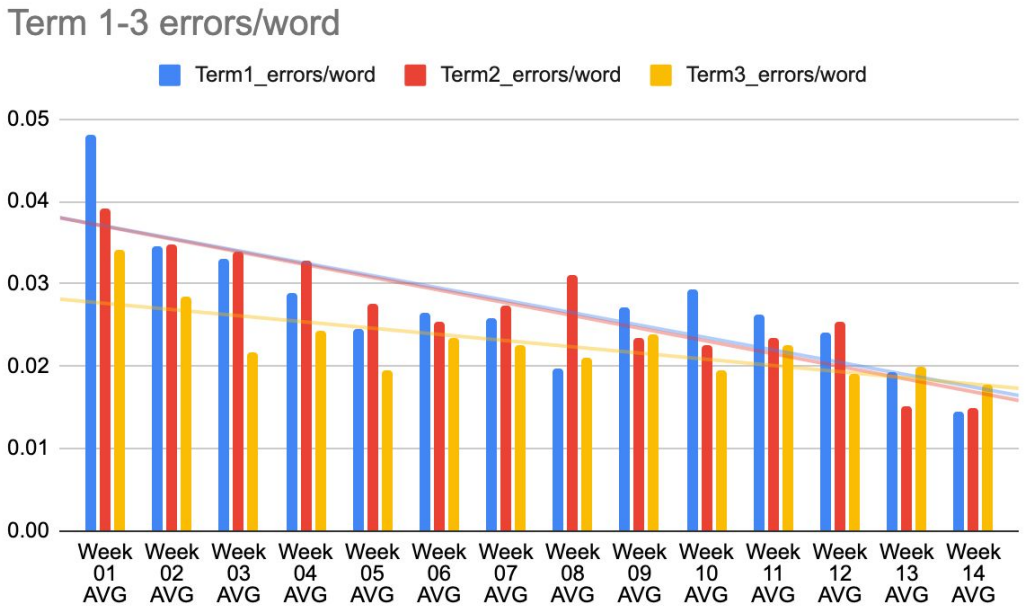
Checking grammatical accuracy

- download speeches-gen-prompts.tsv
- place TSV file into input-files folder
- go to ‘LLMscripting-main’
  in Windows Powershell / Mac Terminal
- On Windows type: python GPT.py speeches-gen-
- On Mac type: python3 GPT.py speeches-gen-
- monitor script, use ChatGPT for errors
- copy processed output in output-files
- into speeches-gen-GPTresults B39

Chain of thought
grammatical accuracy

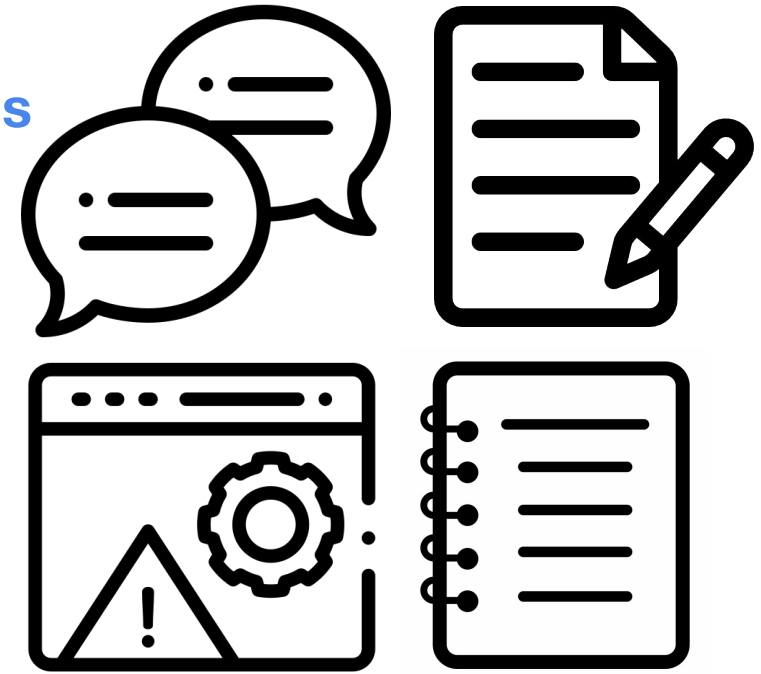
- download speeches-raw.tsv
- place TSV file into input-files folder
- go to ‘LLMscripting-main’
  in Windows Powershell / Mac Terminal
- On Windows type: python GPT.py speeches-gen-
- On Mac type: python3 GPT.py speeches-gen-
- monitor script, use ChatGPT for errors
- copy processed output in output-files
- into speeches-gen-GPTresults B39

Checking grammatical accuracy

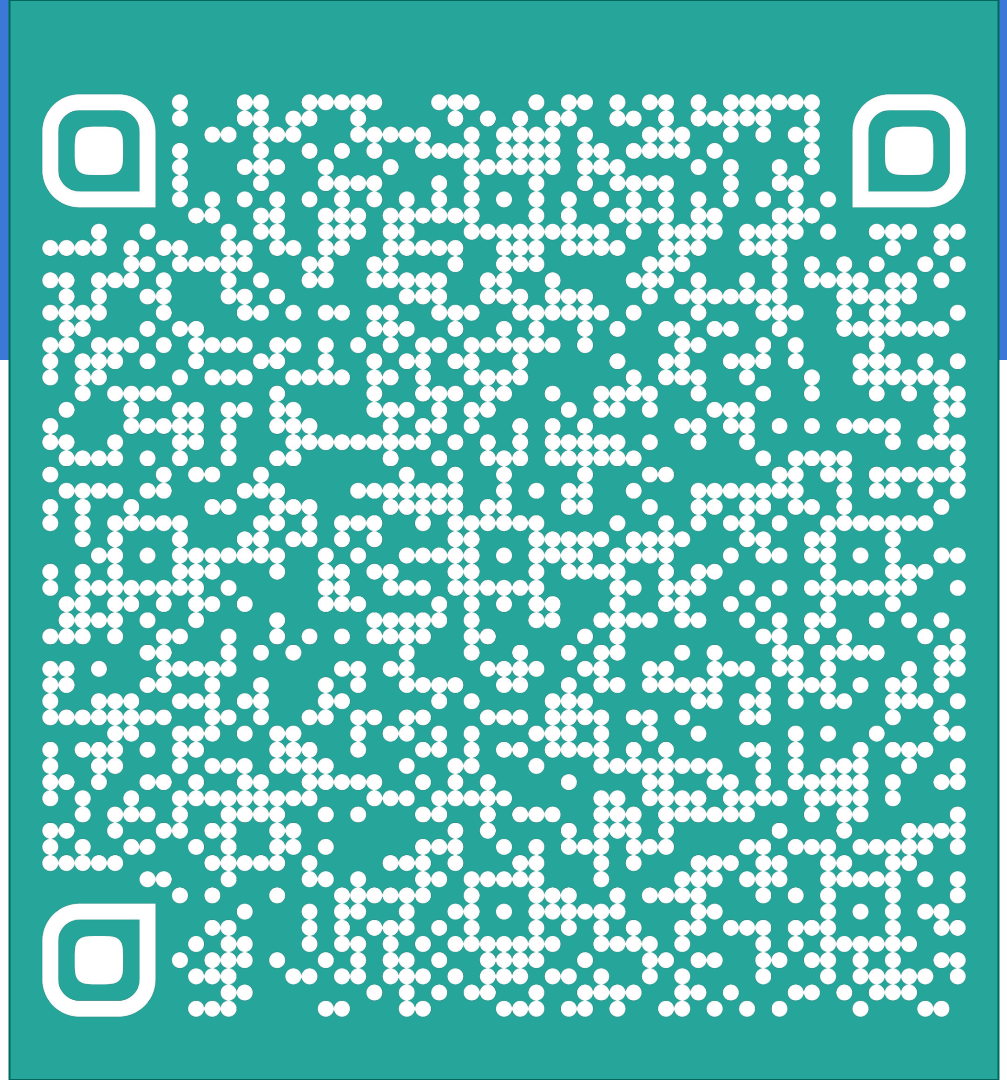


Improving writing instruction Through LLM Scripting

- Thanks for your time and attention
- Let me know if you have any questions
- richard.rose@yonsei.ac.kr



Cite this presentation:

Rose, R. (2024). Improving grammatical accuracy through instructor, peer, and LLM-assisted corrective feedback. Manuscript in preparation.

Rose, R. (2024). Improving syntactic complexity in engineering students' writing through digital portfolios and visual analytics. Manuscript in preparation.