L5: Augmenting Data

Professor Daniel Yue
Information Technology Management Area
Georgia Tech, Scheller College of Business
MGMT 7609 – Observational Studies of Information Systems

This Week

~~L1 Planning Research~~

~~L2 Collecting Data (Scrapers)~~

~~L3 Building Data Pipelines (Databases)~~

~~L4 Exploring Data (Viz)~~

L5 Augmenting Data (LLMs)

L6 Merging Data (Crosswalks)

L7 Student Presentations

Motivation

The Big Picture: LLMs in Research

- L2-L4 focused on using LLMs/AI tools as an assistant – boosting productivity in writing code, implementing our data pipelines and exploring the data.
- This is distinct from today's (and next week’s) focus, using LLMs as a *tool within a research design*.

Two Ways LLMs Function as Research Tools

LLMs as Data Processors (Prompt-to-Response)

- The LLM directly provides the output we want.
- We give it raw text and a prompt.
- It gives us a label, a summary, or structured data.

*Focus for **today** – structure extraction of information*

LLMs as Feature Generators (Embeddings)

- E.g. BERT, Sentence Transformers
- Turns text into dense vectors (numbers).
- These vectors become features used in downstream tasks (e.g., matching, regression).

*Focus for **next week** – application to building crosswalks*

Goals for this lecture:

Introduction to using LLMs for text-based research tasks

Processing Text with LLMs

LLMs as tools in research designs

"Text as Data" (Gentzkow, Kelly, Taddy 2019)

"LLMs to Annotate Data" (Carlson and Burbano 2025)

Extracting Structured Data with LLMs

Practical Application

Extracting data from HF Model Cards

Processing Data with LLMs

**Not discussed today but important:*
Topic Modeling (discovering latent categories)

Data Augmentation

We are not generating new raw text; we are ***augmenting*** an existing dataset

- by *applying judgment to that data* (classification)
- by *adding structure to unstructured data* (structured extraction)

Example 1: Classification
(known categories)

- News Article → Republican or Democrat
- Model Card → Did this model use a proprietary dataset?

Example 2: Structured Extraction
(unknown categories)

- News Article → List of all people mentioned in the article
- Model Card → Which evaluation metrics are reported?

Text as Data (Gentzkow, Kelly, & Taddy 2019)

Raw Text Docs $\mathcal{D}$ $\xrightarrow{\text{“representation”}}$ **Text Features** $\mathbf{C}$ $\xrightarrow[\text{(e.g. SVM, Random Forest)}]{\text{“Prediction”}}$ **Predictions / Signals** $\hat{\mathbf{V}}$

Raw Text Docs $\mathcal{D}$

- Emails
- Transcripts
- Documentation
- Reviews
- Commit Messages
- Model Cards

Text Features $\mathbf{C}$

- Bag of Words (Word Counts)
- TF-IDF Scores
- Topic Model Representations
- Embeddings

Predictions / Signals $\hat{\mathbf{V}}$

- Spam
- Sustainability-Related
- Sentiment
- Racial Animus

$\hat{\mathbf{V}} \xrightarrow{\text{“estimation”}}$ **Parameters** $\hat{\beta}$

Example Application: [Miric, Jia, and Huang 2023]

Text as Data (Gentzkow, Kelly, & Taddy 2019) with LLMs!

Raw Text Docs $\mathcal{D}$ $\xrightarrow{\text{“prompting”}}$ **Predictions / Signals** $\hat{\mathbf{V}}$

- Emails
- Transcripts
- Documentation
- Reviews
- Commit Messages
- Model Cards

Using LLMs dramatically simplifies the process of extracting signals

There are latent “embeddings” (intermediate representations) being created that we are ignoring for now.

- Spam
- Sustainability-Related
- Sentiment
- Racial Animus

“estimation”

Parameters $\hat{\beta}$

Example Application: [Carlson and Burbano 2025]

Helpful Framework: Carlson and Burbano 2025

1. **Method Selection**. What signal do you need? (Research-design driven)
2. **Model Selection**. Which model to use? Several considerations, and consider using multiple!
3. **Prompt Engineering.** Draft your prompt and define your target JSON output structure.
   - Manually label a small, random sample (n=100) of texts (this is your "gold standard").
   - Run your LLM script over the *same* 100 texts.
   - Compare LLM output to human labels. Calculate accuracy. Analyze errors. Refine your prompt/schema.
4. **Cost and Scale Considerations**
   - Once validation is high (e.g., >95% agreement), run on your full dataset.
5. **Analytical Validation and Robustness.**
   - Check robustness of your parameters to prompting choices

Research stage	Principle(s)	Tradeoffs	Best practices
1. Method selection and integration (*Do we use LLMs or another method?*)	Consider LLMs as part of a broader methodological toolkit; match method to research context and data characteristics	• Human Coding: High accuracy and nuance but expensive/slow/limited scale • Keywords/Dictionary: Transparent and simple but rigid/context-blind • Supervised ML: Scalable and consistent but requires training data/ domain expertise • LLMs: Flexible and fast but potentially unstable/ opaque	• Begin with clear task specification and success criteria • Use human coding for validation and complex edge cases • Consider hybrid approaches • Document relative performance across methods

Research stage	Principle(s)	Tradeoffs	Best practices
2. Model selection and stability (*Which LLM should we be using?*)	Prioritize reproducibility and documentation over cutting-edge performance	• Newer models: Better performance but less tested/stable • Older models: More stable but potentially lower performance • Open vs. closed models: Access/cost vs. performance • Reasoning models: Optimized for complex reasoning vs. uncertain effectiveness for basic annotation tasks	• Document model version and access method • Test multiple models as part of sensitivity analysis • Archive model weights (if open-source) or maintain access to older versions if possible • Plan for replication needs • Consider data security if annotating sensitive data (check policies around usage and storage)

Key Technical Skill #1: Calling LLMs through an API

1.

Research stage	Principle(s)	Tradeoffs	Best practices
3. Prompt engineering (*How do we design and validate our prompts?*)	Structure prompts systematically and transparently; document thoroughly	• Simple prompts: More stable but potentially lower accuracy • Complex prompts: Better performance but harder to replicate • Generic vs. task-specific prompting: Generalizability vs. accuracy	• Use established frameworks (Chain of Thought, few-shot learning) • Generate systematic prompt variations for sensitivity testing • Document all prompt components • Test prompts on diverse examples

Key Issues: Correctness, Improvement, and Reliability

1. How do we know the LLM output is correct?
2. How do we make the LLM output better?
3. How do we interpret / constrain the LLM output format?
   - Imposing JSON format
   - Imposing specific categories

Correctness – Validating the Output

How would you do this?

- Labelled Examples! Need to manually create, can be time consuming.
    - For classification, the labels
    - For structured extraction, the true values
    - ~100 example is usually sufficient (rule of thumb)
- Best practice: create “training” set and “holdout” set to avoid overfitting
    - As if you were training a supervised ML model

Improvement – Making It Better

How would you do this?

1. A few critical tricks:
   - System Prompt — provide and context
   - Few-Shot Prompting (examples) – ***error analysis*** can help here
     - I cannot stress enough how important it is to look at the outputs.
   - Chain-of-thought / suggested reasoning path
   - (Potentially) Tools – enable LLM to seek additional context
2. Better Models!
   - Experiment with a few different models
   - Often, fine-tuned smaller models can outperform larger models (but requires sufficient labels)

Reliability – Constraining the Output

How would you do this?

1. More Prompt Engineering Tricks!
   - Clear instructions
   - Few-Shot Prompting (examples) is extremely powerful
2. Structured Output
   - Modern LLM APIs (like OpenRouter, OpenAI) can force the output to conform to a specific JSON schema.
   - How it works (briefly): The API uses a contextual grammar to guide token generation, ensuring the output is valid JSON.

Key Technical Skill #2: Structuring the Output

1.

Research stage	Principle(s)	Tradeoffs	Best practices
4. Cost and scale considerations (*How do we optimize resources while maintaining rigor?*)	Balance resource usage with statistical power and robustness needs	• Sample size vs. token usage • Prompt complexity vs. processing cost • Depth vs. breadth of sensitivity testing • Model size vs. performance gains	• Plan sensitivity analysis within resource constraints • Document cost structure and optimization decisions • Consider computational constraints for replication

Key Technical Skill #3: Estimating the Cost

1.

Research stage	Principle(s)	Tradeoffs	Best practices
5. Validation and robustness assessment (*How reliable and stable are our findings?*)	Conduct systematic evaluation of result validity and stability	• Depth vs. breadth of validation testing • Cost of comprehensive testing vs. confidence in results • Simple vs. complex sensitivity metrics	• If feasible, maintain gold-standard expert-coded subset • Generate systematic prompt variations • Test impact on downstream analyses • Provide bounded estimates for key findings • Document null results and validation protocols

This part is the wild west in our field right now. Lots of ideas, but unclear what referees may ask for.
I would not over-engineer on this aspect yet.

Review of Carlson and Burbano 2025

1. **Method Selection**. What signal do you need? (Research-design driven)
2. **Model Selection**. Which model to use? Several considerations, and consider using multiple!
3. **Prompt Engineering.** Draft your prompt and define your target JSON output structure.
    - Manually label a small, random sample (n=100) of texts (this is your "gold standard").
    - Run your LLM script over the *same* 100 texts.
    - Compare LLM output to human labels. Calculate accuracy. Analyze errors. Refine your prompt/schema.
4. **Cost and Scale Considerations**
    - Once validation is high, run on your full dataset.
5. **Analytical Validation and Robustness.**
    - Check robustness of your parameters to prompting choices

15 minute break!

Extracting Structured Data with LLMs

Extract data from the Model Cards on Hugging Face to characterize the nature of fine-tuning at scale.

`librarian-bots/model_cards_with_metadata`

Assignment

Add one or more LLM-generated variables to your dataset. Document your validation process and create visualizations comparing automated labels to manual validation samples.

Next Week

~~L1 Planning Research~~

~~L2 Collecting Data (Scrapers)~~

~~L3 Building Data Pipelines (Databases)~~

~~L4 Exploring Data (Viz)~~

L5 Augmenting Data (LLMs)

L6 Merging Data (Crosswalks)

L7 Student Presentations

Class Prep

- Revealing the revealed preferences of public firm CEOs and top executives: A new database from credit card spending — Raffiee et al 2022
  - Section 2.2 and 2.3
- Deep Learning for Economists — Dell 2024
  - Section VIII.1-4 (Embedding Models)
- MTEB: Massive Text Embedding Benchmark — Muennighoff et al 2023
  - Short, influential benchmark — worth reading!