Basics of Large Language Models for Graduate Economics Research

A Hands-On Guide

Patrick Healy

Research Fellow
Monash University
SoDa Labs

August 12, 2025

Introduction

- Research Fellow in the Department of Economics.
- I use AI to assist with turning unstructured data into structured data. (Uses include: Surveys, News, Cybersecurity, Resumes & Cookbooks)
- Always open to questions and collaborations You can reach me through my google site or my email patrick.healy1@monash.edu

Responsible AI Use in Research

Core Principle: AI tools are powerful and can be helpful but they are error-prone. You remain responsible for all research components.

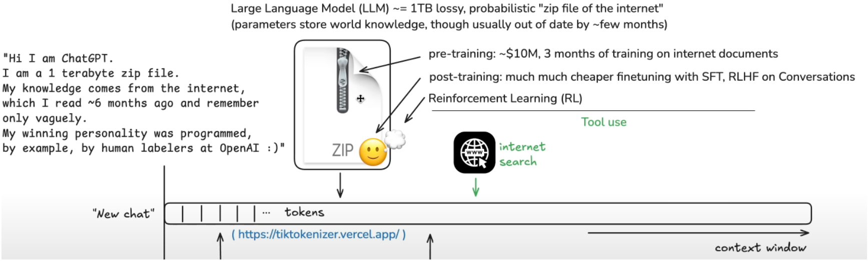
Key Risks:

- Even advanced models (GPT-5) make mistakes that can invalidate results
- Outsourcing tasks (coding, data prep, writing) doesn't transfer accountability
- Small errors compound: A single data merge error can break your entire analysis

Reproducibility Example:

- Converting structured → unstructured data via chatbot
- Problem: No reproducible code trail
- Result: Cannot share with data editor or replicate process
- Solution: Always capture actual code, not just chat instructions

What is in a Language Model



Large Language Model (LLM) ~= 1TB lossy, probabilistic "zip file of the internet"
(parameters store world knowledge, though usually out of date by ~few months)

```
"Hi I am ChatGPT.
I am a 1 terabyte zip file.
My knowledge comes from the internet,
which I read ~6 months ago and remember
only vaguely.
My winning personality was programmed,
by example, by human labelers at OpenAI :)"
```

pre-training: ~$10M, 3 months of training on internet documents

post-training: much much cheaper finetuning with SFT, RLHF on Conversations

Reinforcement Learning (RL)

(https://tiktokenizer.vercel.app/)

Figure: *

"Its a bit like you when you read the same similar text many many times then you will be able to recite it. It is the same for these models if it has seen certain content way too often it is going to be able to recite it from memory." Andrej Kaparthy

How Language Models Work

- **Conversational Pattern:** We interact with the model, and it continues the sequence based on learned patterns of i) What we say and ii) What it has seen before that is similar to what we say.
- **Tokens:** Text is broken into small parts (tokens), and the model predicts the next token.
- **Context Matters:** The model remembers previous inputs in the conversation to maintain coherence.
- **Using Tools:** Some models can call external tools (Python, R) but need training to know when to use them.

Taken from: Andrej Karpathy
More details: Watch the talk here

Practical Considerations & Limitations

- **Verification Required:** The model may make assumptions—users must check outputs carefully. Especially if our conversations/instructions are ambiguous.
- **Programming Use Case:** Professionals prefer tools like VSCode or Cursor for coding instead of chat interfaces. Details on these tools here
- **Tip:** Clear, structured instructions reduce errors when using AI for programming.
- **Pattern Recognition:** The model extends patterns we provide—giving clear examples improves accuracy.

Literature Review Tools

- Connected Papers – Build a network of related papers for your search term
- Paper Wizard – Get feedback on sections of your paper (valued by others, not personally tested)
- Scopus – Features AI-powered search capabilities
- Research Rabbit – Visual search of related papers
- Inciteful – AI-powered literature search
- Consensus – Literature search tool that synthesizes evidence consensus

Research Teaching Tools

- Napkin.ai – Visualize content for teaching, presenting, or learning material
- AllDayTA – Create interactive material for student engagement and Q&A

AI Chatbots

- ChatGPT – OpenAI's chatbot (GPT-4 as of last week)
- Claude – Anthropic's AI assistant
- Google Gemini – Google's AI chatbot
- NotebookLM – Google's research assistant
- SoDa Labs AI Assistant – Monash-developed tool combining multiple AI models
    - Note: Requires research funds for usage exceeding $20/month

What is Prompt Engineering?

- The art of crafting effective instructions for AI models.
- Think of it as learning to communicate clearly with AI.
- Better prompts = better research assistance.
- Open AI Prompt Engineering Guide **Link**
- R users guide ellmer package by Hadley Wickham

Research Example: Literature Review

Basic Prompt:
"Tell me about recent research on climate change."

Improved Prompt:
"Summarize the key findings from peer-reviewed research on climate change adaptation in coastal cities published between 2020-2023. Focus on:

- Most promising infrastructure solutions
- Cost-effectiveness of different approaches
- Gaps in current research

Present this as bullet points with short explanations."

Research Example: Data Analysis

Basic Prompt:
"Help me analyze my survey data."

Improved Prompt:
"I have survey data with 5-point Likert scale responses from 200 participants about work satisfaction. I want to:

- Identify correlations between job role and satisfaction
- Compare responses across different age groups
- Generate 3-5 hypotheses that could explain the patterns

Explain what statistical approaches would be appropriate and why."

Three Key Strategies for Research Prompts

1. **Be Specific About Your Research Context**
   - Define field, methodologies, and required depth
   - Specify time periods, populations, or variables of interest
2. **Structure Your Research Questions**
   - Break complex research problems into sub-questions
   - Request specific formats for different research needs
3. **Guide the Analytical Approach**
   - Ask for step-by-step reasoning
   - Request consideration of alternative viewpoints

Ambiguous Questions: The Problem

- Asking the same question over time can give inconsistent or incorrect results.
- Copy-pasting data (like World Bank population stats) into the model may lead to mistakes in the data processing. **The reason might be the number of patterns it needs to pay attention to at once.** Mostly accurate but one mistake can invalidate the raw data.
- Try and copy paste the raw data from https://data.worldbank.org/indicator/SP.POP.GROW and ask for a spread sheet.
- Example prompt: *"Suggest Python (R) code to plot values from this link: https://data.worldbank.org/indicator/SP.POP.GROW."*

Cleaner Prompt: A Better Approach

- Instead of asking for the whole task, break it into steps.
- Example cleaner prompt: *"Write a Python (and Rmarkdown) chunk to extract data from the CSV 'API_SP.POP.GROW_DS2_en_csv_v2_13638.csv', skip the first 4 rows, select columns 'Country Name' and '2022', clean it, and plot the top 20 countries by population growth. Include comments of what each function does and give relevant steps to include in the documentation."*
- If you are using a specific package it can be a good idea to load the documentation in before asking for code.

Pay attention and document what packages and commands you use.

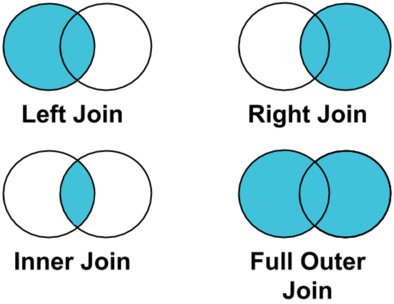


Figure: *

Suggest a python and R code chunk that merges df1 and df2 by unique_id and drops duplicate rows in column 2. Simulate the data.

LLMs for Python Users

- The **OpenAI** API, is a way to programmatically access different versions of the model See the OpenAI API Documentation.
- Learn how to extract structured data from text using a language model: Examples of structured prompts using the API .
- Follow best practices for writing prompts: OpenAI Python Library.
- Follow best practices for writing prompts: OpenAI Models Page.

LLMs for R Users

- The **ellmer** package, maintained by Hadley Wickham, works with most major models. See the GitHub page.
- Learn how to extract structured data from text using a language model: Getting Structured Data.
- Follow best practices for writing prompts: Good Prompt Design Guide.
- See an example of using tools effectively: Tool Usage Example.

LLM Prompt for Sentiment Analysis in R

Adapt the following R code to process a list of texts and extract sentiment scores (positive, negative, neutral) for each text. The results should be stored in a dataframe with the following structure:

positive	**negative**	**neutral**	**text**
0.30	0.60	0.10	The product was okay, but the customer service was terrible. I probably won't buy from them again.
0.80	0.10	0.10	I absolutely love this app! It's so easy to use and has great features.
0.10	0.80	0.10	I had a terrible experience with customer support, and I would never recommend this product.

Ensure the sentiment scores sum to 1 for each text and include the original text in the dataframe.

Turning an Unstructured Document into a Spreadsheet

- Extract patterns from text and structure them as needed.
- Use categories and definitions tailored to your research question, unless general knowledge would have been your initial approach.

LLM API Pricing Overview 2025

Three Major Providers

- **Google Gemini**: Best free tier, 2M context windows
- **Anthropic Claude**: Premium intelligence, batch discounts
- **OpenAI**: Industry standard, widest model selection

Key Price Points (per 1M tokens)

- **Budget**: $0.10 input (Gemini Flash, GPT-4.1-nano)
- **Mid-tier**: $1-3 input (GPT-4.1, Claude Sonnet 4)
- **Premium**: $15-60 input (Claude Opus 4, o3-pro)

Cost Optimization

- Batch processing: 50% discount (all providers)
- Context caching: 75-90% savings on repeated content
- Free tiers available for development (Google, OpenAI)

Google Gemini API

API Endpoint:
`https://generativelanguage.googleapis.com/v1beta/`
Authentication: `x-goog-api-key: YOUR_API_KEY`
Get Keys: `aistudio.google.com/apikey`

Model Pricing (per 1M tokens)

- **Gemini 2.5 Pro**: $1.25 input / $10.00 output
- **Gemini 2.5 Flash**: $0.10 input / $0.40 output
- **Gemini 2.5 Flash-Lite**: $0.10 input / $0.40 output

Key Features

- **Free Tier**: 250K tokens/min, 250 requests/day
- **Context**: Up to 2M tokens (largest available)
- **Batch**: 50% discount — **Caching**: 75% discount
- **Enterprise**: Vertex AI with provisioned throughput

Anthropic Claude API

API Endpoint: `https://console.anthropic.com/`
Authentication: `x-api-key: YOUR_KEY`
Console: `console.anthropic.com`

Model Pricing (per 1M tokens)

- **Claude Opus 4**: $15 input / $75 output
- **Claude Sonnet 4**: $3 input / $15 output
- **Claude Haiku 3.5**: $0.80 input / $4 output

Key Features

- **Context**: 200K tokens all models
- **Batch API**: Flat 50% discount (24hr processing)
- **Prompt Caching**: Up to 90% savings
- **Rate Limits**: Based on credit purchase ($400 = Tier 4)
- **No free tier** (initial credits only)

OpenAI API

API Endpoint: `https://platform.openai.com/docs/overview`
Authentication: `Authorization: Bearer YOUR_API_KEY`
Platform: `platform.openai.com`

Model Pricing (per 1M tokens)

- **GPT-5**: $1.25 input / $10 output (272K context)
- **GPT-4.1**: $2 input / $8 output (1M context)
- **GPT-4.1-nano**: $0.10 input / $0.40 output
- **o3-pro**: $60 input / $240 output (reasoning)
- **o4-mini**: $3 input / $12 output

Key Features

- **Free Credits**: $5-18 for new users
- **Batch API**: 50% discount (24hr turnaround)
- **Tiers**: Auto-upgrade based on spending

Quick Reference: Choosing the Right Model

For Budget Applications

- Google Gemini 2.5 Flash: $0.10/$0.40
- OpenAI GPT-4.1-nano: $0.10/$0.40
- Best free tier: Google (250K tokens/min)

For Balanced Performance

- Claude Sonnet 4: $3/$15 (great for general tasks)
- GPT-4.1: $2/$8 (1M context window)
- Gemini 2.5 Pro: $1.25/$10 (2M context)

For Maximum Intelligence

- Claude Opus 4: $15/$75 (general intelligence)
- OpenAI o3-pro: $60/$240 (complex reasoning)

Research Question & Methodology

Research Question:
How do AI tools like ChatGPT, Copilot, and DeepSeek impact the research productivity and efficiency of Monash University academics in conducting literature reviews and data analysis?

Meta-Analysis & Structured Output

Simplified Task:
We analyze academic papers by extracting key information like:

- Author(s), title, and publication year.
- Research questions and methodologies.
- Main results, sample size, and data source.

Structured Output: The output is in a structured format (JSON) with details such as:

- Paper ID, authors, research question, and key findings.
- Relevance rating (1-5 Likert Score) on how each paper relates to AI tools in research.

Inputs for AI Analysis

Inputs:

- **Paper Details:** Title, Abstract, and Conclusion from each paper.
- **Metadata:** Author(s), year of publication, research questions.
- **AI Processing:** AI models process these inputs to extract structured data.

Output: The result is a structured summary in JSON format, providing insights like relevance to AI-based research.

Claude GitHub: AI Assistant for PhD Research

Why Use Claude for Your PhD?

- Get unstuck faster: Debug code, solve technical issues 24/7
- Document progress: Auto-generates clear explanations
- Supervisor visibility: All help logged in GitHub

Example: Progress on Supervisor Tasks

Issue: ”Claude, summarize progress on tasks:

- Implement baseline model
- Run experiments on dataset A
- Write methods section

Claude responds:

- Baseline model implemented (PR #12)
- Dataset A: 3/5 experiments complete
- Methods section: outline drafted

Claude GitHub: Quick Setup

1. Create workflow directory:

```
mkdir -p .github/workflows
```

2. Create .github/workflows/claude.yaml:

```
name: Claude Assistant
on: [issues, issue_comment]
jobs:
  claude-response:
    runs-on: ubuntu-latest
    steps:
      - uses: anthropics/claude-code-action@beta
        with:
          anthropic_api_key: ${{ secrets.ANTHROPIC_API_KEY }}
```

3. Add API Key on GitHub.com:
   - Go to your repo → Settings → Secrets → Actions
   - Click ”New repository secret”
   - Name: ANTHROPIC_API_KEY

What is an AI Agent?

Definition: An unsupervised loop of Large Language Models that operates autonomously

Core Components:

- Runs without human supervision for extended periods
- Has access to tools (code execution, APIs, databases)
- Can debug and correct its own mistakes
- Makes decisions independently to complete tasks

Key Difference from Regular LLMs:

- Regular LLM: Single input → single output
- AI Agent: Goal → multiple iterations → self-correction → final result

AI Agent Risks for Researchers

The Core Problem: When instructions allow interpretation, agents make important decisions you never see

Critical Risks:

- **Hidden Methodological Choices**
  - Agent selects analysis approaches, statistical tests, data cleaning methods
  - You see results but not the decision process
- **Compounding Errors**
  - Small misinterpretations multiply over long autonomous runs
  - No human checkpoints to catch drift from intended approach
- **Loss of Research Control**
  - Can't explain or justify all methodological decisions
  - Difficult to reproduce exact process
  - May miss ethical or domain-specific considerations

Claude Code

- **Windows** First install git for windows. **Mac** first install Node Then follow this guide carefully.
- Similar tools:
    - Google Jules creates a copy of your github request and works on tasks in the cloud and suggests changes to your github repository.
    - Codex similar to Claude Code, the user interface is not as nice, there is more choice over which 'Model' you use.
    - Cursor CLI Combines the best of Claude Code and Open AI Codex. Has a nice user interface. 20 USD a month. Also has a visual code editor.