# Large Language Models for Natural Language Processing

Abigail Haddad

July 26, 2023

# If You Remember Three Things

Large Language Models can provide fast-to-build and accurate-enough solution for tasks like named entity recognition, summarization, and classification
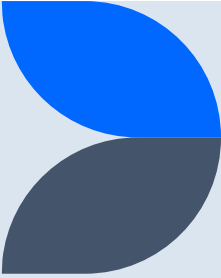
You can (and should!) systematically assess your output

Think carefully about risk – what you're putting in and what you're doing with the results

# "I Used To Be At Army"

-Public policy PhD from RAND -> data science
-Around Army/DoD after that
-Currently in a lead data science role for a DHS client

# What is a Large Language Model?

A model designed to process and generate human-like text to perform tasks like answering questions, summarizing data, or translating languages. They're "large" in that they were trained on a lot of data and have a range of capabilities.
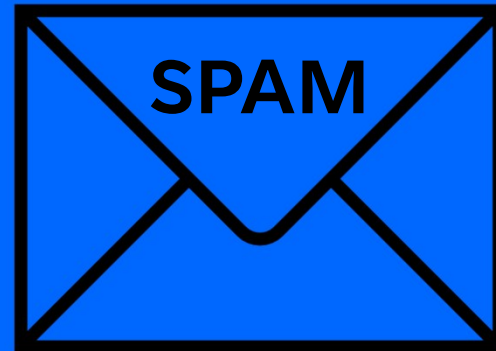
**Examples:**

- **GPT-3.5 and GPT-4 (OpenAI)**

- **Claude and Claude 2 (Anthropic)**

- **Bard (Google)**

- **LLaMA and LLaMA 2 (Facebook/Open Source/many descendants)**

# What is Natural Language Processing (NLP)?
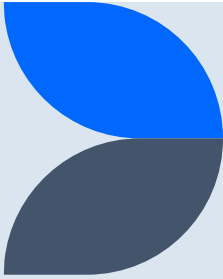
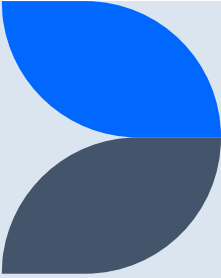A set of techniques that let computers analyze human language

# Classification



HAM

SPAM

# Sentiment Analysis

# Language Translation
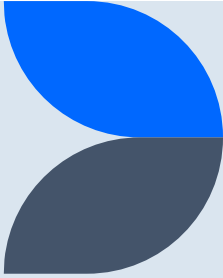
# Summarization

BERT (Bidirectional Encoder Representations from Transformers) is a powerful method for natural language processing (NLP) tasks developed by Google. It's considered a breakthrough because it handles the context of words in a sentence in a way that hadn't been done before. Before BERT, models read text input from left to right or from right to left. But BERT, as the "bidirectional" part of its name suggests, reads the entire sequence of words at once in both directions, which is beneficial because the meaning of a word can depend on what comes before and after it .BERT is pre-trained on a large corpus of text (like the entirety of Wikipedia) and then fine-tuned for specific tasks, like sentiment analysis, question answering, or named entity recognition. The pre-training phase helps BERT understand the general context of language, like how words are usually used together, and the fine-tuning phase adapts this knowledge to specific tasks. An additional benefit of BERT is that you can use it right out of the box for your NLP task. You just take the pre-trained BERT model and fine-tune it on your specific task. This "transfer learning" approach saves you from having to train a model from scratch, which can be very time-consuming and requires a lot of data.

BERT (Bidirectional Encoder Representations from Transformers) is a powerful method for natural language processing (NLP) tasks developed by Google. Before BERT, models read text input from left to right or from right to left. But BERT, as the "bidirectional" part of its name suggests, reads the entire sequence of words at once in both directions, which is beneficial because the meaning of a word can depend on what comes before and after it. You just take the pre-trained BERT model and fine-tune it on your specific task. This "transfer learning" approach saves you from having to train a model from scratch, which can be very time-consuming and requires a lot of data.

# Named Entity Recognition

**John Doe**, an employee of **Google Inc.**, bought **500 shares** of the company's stock on **July 1, 2023**, at the **New York Stock Exchange** for **$1000**.

```
{"PERSON": ["John Doe"],
   "ORGANIZATION": ["Google Inc.", "New York Stock Exchange"],
   "DATE": ["July 1, 2023"],
   "MONEY": ["$1000"],
   "QUANTITY": ["500 shares"]}
```

# New LLMs as "Pretrained Everything" Models

Some of these tasks we previously had out-of-box solutions for
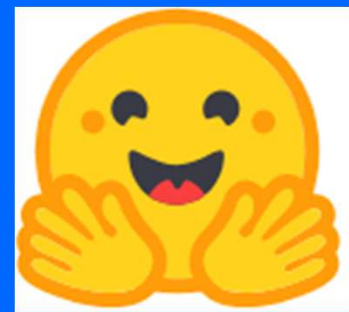
Others required labeling massive amounts of data just to see if it might work

Now, you can build and test faster

# How Fast Are We Talking?

- ~100 lines of code/ an afternoon for OpenAI's API's – you may spend longer pulling/cleaning your data

- Potentially longer if you want to use open-source tools (more setup/dependency management/environment management)

# Prompts I'm Using

## 01

Provide a summarized list of key points from this job posting

## 02

List UP TO 5 occupational names that match this job posting

## 03

please respond YES if this candidate has worked in a role where they have been coding in R or Python to solve data science tasks and NO if they have not or if you are not sure.

# Process

**Read in your text**

Get your text from your .csv file, API, etc.

Write a prompt

**Use With Your Model**
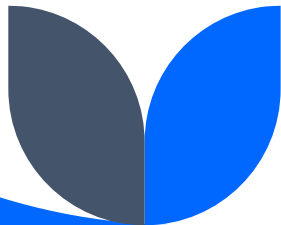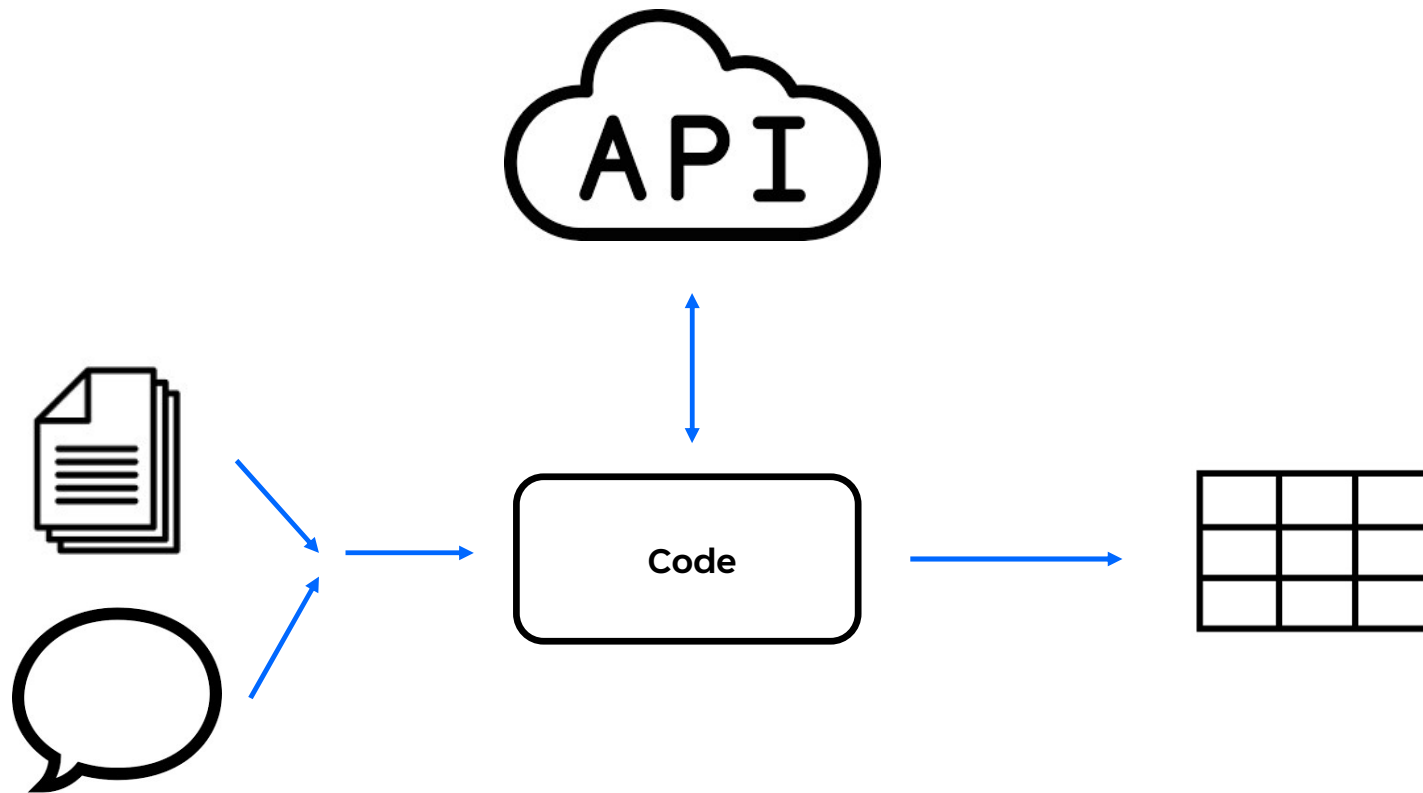
Send a block of text

Get back a result

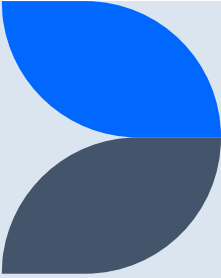Put it in a table/data frame

**Evaluate the Results**

Informally: does it look like you want?

Formally: via comparing with your 'labeled' data
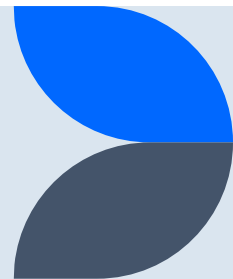
Code

# Assessment is important

# Binary Classification Assessment

- "Good" vs. "bad" data science resumes

- Tested combinations of models (GPT 3.5/GPT 4) and prompts

- Results: confusion matrices

|  | **Actual Values** | |
|---|---|---|
|  | **Positive (1)** | **Negative (0)** |
| **Positive (1)** | TP | FP |
| **Negative (0)** | FN | TN |

Predicted Values

# Tools for Comparing Text to the Text You Wanted (Your Label)

Exact search (yes/no)

Regex/keywords ("does it include 'python'")

Text similarity metrics

Ask the LLM!

# Managing Risk: What You Put In

Follow your organization's policies
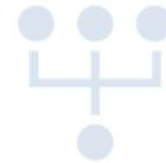
Prototype with fake data to prove out your use case

# Managing Risk: What You Get Out

What level of accuracy do you need?

How do you feel about Type 1 vs. Type 2 error?

Do you need a human in the loop?

# Can I Do This In R?

**OpenAI API Interaction**

Yes.

**Open-source models**

There are resources – check out

https://rpubs.com/eR_ic/transfoRmers

https://blogs.rstudio.com/ai/posts/2023-07-12-hugging-face-integrations/

# For More Information

Slides: https://github.com/abigailhaddad/mors_llm_talk

Blog: https://presentofcoding.substack.com/

Example Project/Code: https://github.com/abigailhaddad/resume_classification