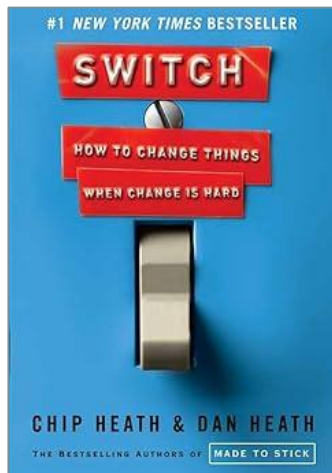


# LLM Engineering I

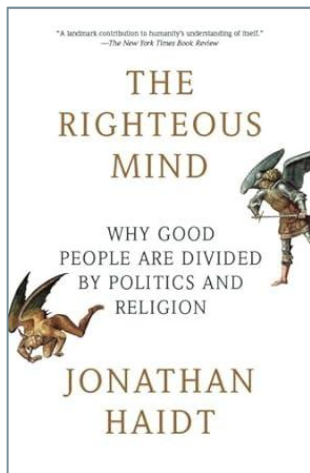
- Social Media Problem
- Data (YouTube comments)
- LLMs → Open Source, local

# **Moderation, Social media data, and LLMs**

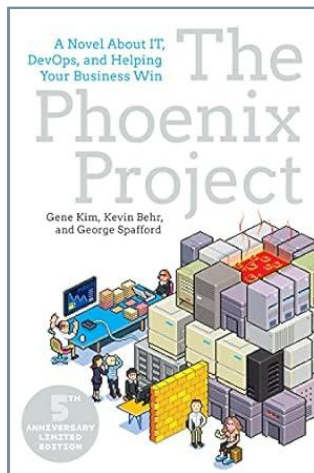
# From last class - a few books on change



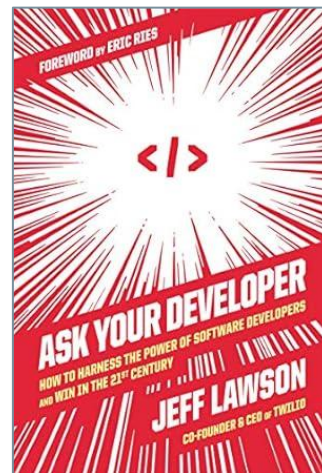
Switch: How to Change Things When Change Is Hard by Chip Heath



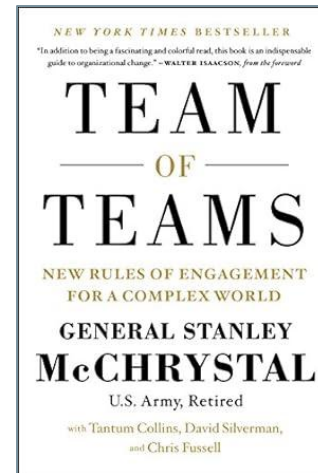
The Righteous Mind: Why Good People Are Divided by Politics and Religion by Jonathan Haidt



The Phoenix Project by Gene Kim



Ask Your Developer: How to Harness the Power of Software Developers by Jeff Lawson



Team of Teams: New Rules of Engagement for a Complex World by General Stanley McChrystal

# Today - Social Media Data

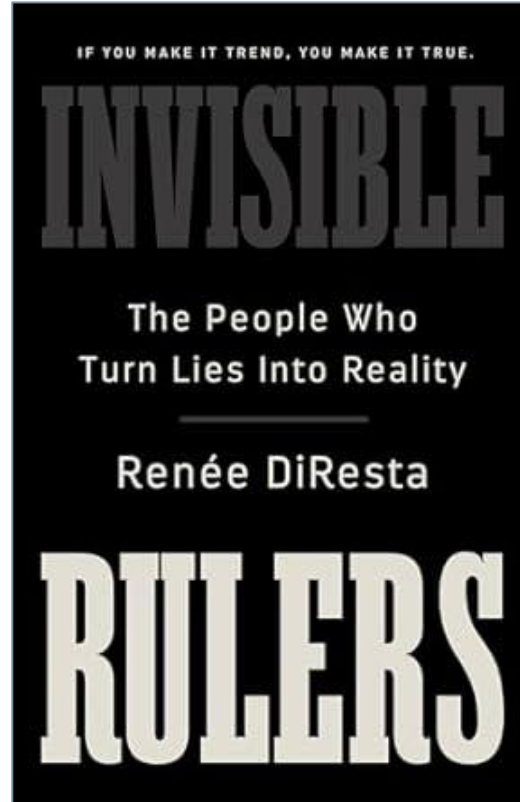


# Book on digital influence

*Invisible Rulers:*

The People Who Turn Lies into Reality

<https://www.amazon.com/dp/1541703375>



The earth  
is **NOT**  
a globe!

Official NASA photo of earth



Why Do Flat Earth Believers Still Exist?  
<https://youtu.be/mYB1JP-gfLE?t=212>

Search **You Tube**

**"A Strangers Guide  
to Flat Earth  
21 Questions"**

 **FE2017.com**

 **TheFlatEarthPodcast** 

# You can't change people's mind with reason

\* A dog's tail wags to communicate. You can't make a dog happy by forcibly wagging its tail. And you can't change people's minds by utterly refuting their arguments. Hume diagnosed the problem long ago:

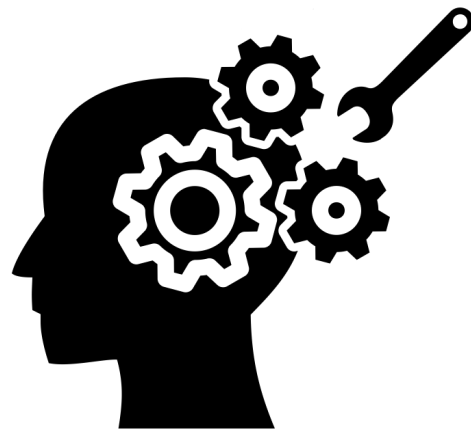
*And as reasoning is not the source, whence either disputant derives his tenets; it is in vain to expect, that any logic, which speaks not to the affections, will ever engage him to embrace sounder principles.*

\* The Righteous Mind: Why Good People Are Divided by Politics and Religion by Jonathan Haidt

# Why Facts Don't Change Our Minds

The human capacity for reason may have more to do with winning arguments than with thinking straight.

Mercier and Sperber argue that reason is an evolved trait that developed to help humans resolve problems in collaborative groups. They argue that ***reason's main purpose is to help humans justify their beliefs and actions to others, and to evaluate the justifications and arguments that others present.***



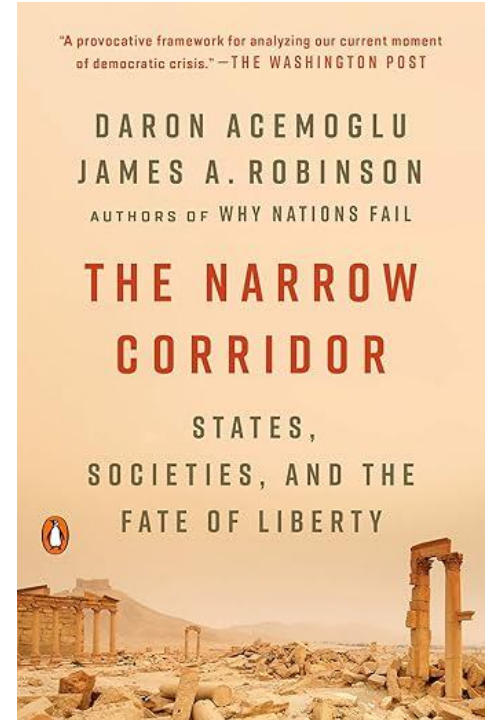
The Enigma of Reason  
by Hugo Mercier, Dan Sperber

Kolbert, Elizabeth. "Why facts don't change our minds." The New Yorker 27.2017 (2017): 47.



# A new digital world - The Narrow Corridor

Liberty is hardly the "natural" order of things. In most places and at most times, the strong have dominated the weak and human freedom has been quashed by force or by customs and norms. Either states have been too weak to protect individuals from these threats, or states have been too strong for people to protect themselves from despotism. Liberty emerges only when a delicate and precarious balance is struck between state and society.



# The key recommendations

1. **Content Moderation:** It emphasizes the importance of consistent and transparent content moderation. Platforms should follow international human rights laws and focus on balancing free speech with harm prevention. Algorithms need refining to prioritize civility and reduce polarization without completely stifling speech.
2. **Policy and Regulation:** The text suggests that government policies should focus on oversight and transparency, rather than direct involvement in content moderation. Enhancing transparency around government takedown requests and allowing researchers access to social media data can help maintain accountability.
3. **User Autonomy and Control:** There is a call for increasing user control over their content experiences, such as allowing users to customize their feeds and moderation preferences. This can include middleware solutions where users choose third-party curation, enhancing personal relevance and reducing unwanted exposure.
4. **Design and Friction:** It recommends implementing strategic friction, like slowing down the spread of potentially harmful content to allow for fact-checking. Design changes, such as creating mechanisms to minimize impulsive sharing, can help curb the spread of misinformation and reduce polarized interactions.
5. **Decentralization:** The document concludes by exploring the future of social media as potentially decentralized, where communities have more control, and moderation occurs at a more localized level, similar to how platforms like Reddit operate with community-based moderation.

# Decentralized moderation

**Community-Based Moderation:** In a decentralized system, moderation could occur at the community level, where individual groups or smaller platforms set their own rules. For example, Reddit's model allows each subreddit to have unique guidelines and moderation practices that align with the values of its specific community. This means that while the platform enforces broad policies on major issues like hate speech, the day-to-day management is left to local moderators.

**Federated Networks:** Platforms could adopt a structure similar to federated networks, where independent communities or servers interact but maintain their own rules. A well-known example is Mastodon, where multiple servers (instances) have their own community guidelines and moderation policies but can still communicate with one another across the network. This allows users to choose communities that align with their values.

**Increased User Choice:** Decentralization would give users the flexibility to choose from various platforms or communities based on their moderation preferences and content norms. Users might gravitate towards spaces where they feel most comfortable, without needing to conform to a one-size-fits-all approach.

**Innovation in Moderation Models:** By allowing different communities to experiment with various moderation techniques, platforms can foster innovation. This localized experimentation can reveal what works best for specific types of communities and could lead to a broader set of best practices that others can adopt or adapt.

**Resilience Against Control:** Decentralized systems can also be more resilient against censorship and control by a single entity. If one community or server enforces overly restrictive rules, users can move to or create another that better aligns with their preferences, promoting diversity in content and viewpoints.

The image is a screenshot of a web browser displaying the Node.js GitHub repository page. The browser's address bar shows the URL `github.com/nodejs/node`. The repository name `nodejs / node` is visible, along with a `Public` badge. Navigation buttons for `Notifications`, `Fork` (with 29.3k forks), and `Star` (with 107k stars) are present. The main content area is divided into two columns. The left column features the `Contributor Covenant Code of Conduct` section, which includes a sub-section titled `Our Pledge`. The text under `Our Pledge` states: "We as members, contributors, and leaders pledge to make participation in our community a harassment-free experience for everyone, regardless of age, body size, visible or invisible disability, ethnicity, sex characteristics, gender identity and expression, level of". The right column features the `Moderation policy` section. It begins with the text: "If you are not a member of the Node.js GitHub Organizations and wish to submit a moderation request, please see [Requesting Moderation](#)". Below this, there is a bulleted list of links: [Applicability](#), [Terms](#), [Grounds for Moderation](#), [Requesting Moderation](#), [Consideration of Intent](#), and [Guidelines and Requirements](#).

github.com/nodejs/node

nodejs / node Public

Notifications Fork 29.3k Star 107k

## Contributor Covenant Code of Conduct

### Our Pledge

We as members, contributors, and leaders pledge to make participation in our community a harassment-free experience for everyone, regardless of age, body size, visible or invisible disability, ethnicity, sex characteristics, gender identity and expression, level of

## Moderation policy

If you are not a member of the Node.js GitHub Organizations and wish to submit a moderation request, please see [Requesting Moderation](#)

- [Applicability](#)
- [Terms](#)
- [Grounds for Moderation](#)
- [Requesting Moderation](#)
- [Consideration of Intent](#)
- [Guidelines and Requirements](#)

`https://github.com/nodejs`

# Papers

- Goldstein, Josh A., et al. "***Generative language models and automated influence operations: Emerging threats and potential mitigations.***" arXiv preprint arXiv:2301.04246 (2023).
- McGuffie, Kris, and Alex Newhouse. "***The radicalization risks of GPT-3 and advanced neural language models.***" arXiv preprint arXiv:2009.06807 (2020).
- Crothers, Evan N., Nathalie Japkowicz, and Herna L. Viktor. "***Machine-generated text: A comprehensive survey of threat models and detection methods.***" IEEE Access 11 (2023): 70977-71002.



# Data



Open a new codespace instance - default blank template





## Exercise: Let's get some data - YouTube comments

- How do we get some without going through the Google YouTube API?

# YouTube comments extractor

## 1) Simple script for downloading Youtube comments without using the Youtube API

<https://github.com/egbertbouman/youtube-comment-downloader>

To install:

```
$ pip install youtube-comment-downloader
```

## 2) Download the comments

```
$ youtube-comment-downloader --url https://www.youtube.com/watch?v=U5KaGM1pXfo --output data.json
```

Or

```
$ youtube-comment-downloader --youtubeid U5KaGM1pXfo --output data.json
```

See documentation for more options:

<https://github.com/egbertbouman/youtube-comment-downloader>

# Exercise: How do you read it?

- Using python

# Read JSON data

```
# -----  
#   Verify that you can read the json file  
# -----  
import json  
  
file = open('data.json')  
data = json.load(file)  
# print(data[0])  
# print(data[0]['cid'])  
# print(data[0]['text'])  
  
for index, item in enumerate(data):  
    print(item['cid'], index)  
    # print(item['text'], index)
```

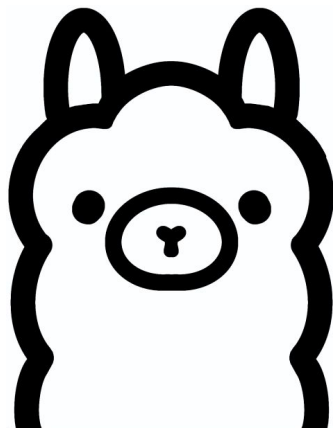
# Sample data - YouTube comments

- Zuckerberg on Lex Fridman
  - <https://compuxo.org/assets/data/zuck.json.zip>
  - 17,169 Comments
- Tucker on Lex Fridman
  - <https://compuxo.org/assets/data/tuck.json.zip>
  - 60,901 Comments

# Large Language Models (LLMs)



# Running your own LLM - Install Ollama

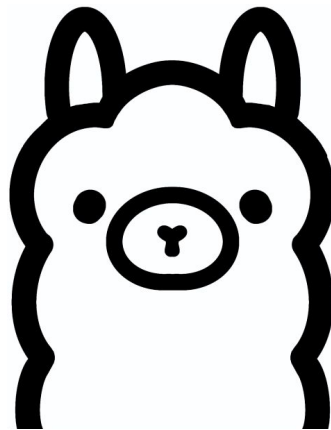


`https://ollama.com`

`*Make sure you install the command line`

# Ollama - run LLMs locally, great for *little* large language models

- Free
- Great for exploration
- User friendly
- Install locally easily
- 100+ models
- Model library
  - <https://ollama.com/library>



<https://ollama.com>



# llama3.2

*Sample model*

Meta's Llama 3.2 goes small with 1B and 3B models.

Tools

1B

3B

↓ 254.2K Pulls    ⌚ Updated 6 days ago

1b



🏷️ 63 Tags

ollama run llama3.2:1b



Updated 7 days ago

baf6a787fdff · 1.3GB

model	arch <b>llama</b> · parameters <b>1.24B</b> · quantization <b>Q8_0</b>	1.3GB
template	< start_header_id >system< end_header_id > Cutting Knowled...	1.4kB
license	<b>**Llama 3.2**</b> <b>**Acceptable Use Policy**</b> Meta is committed ...	6.0kB
license	LLAMA 3.2 COMMUNITY LICENSE AGREEMENT Llama 3.2 Version Re...	7.7kB

*Note size*  
**1.3 GB**



# Ollama - Hello World!



- Make sure you installed the command line
- Run any model with
  - `% ollama run [model name]`
- Model library
  - <https://ollama.com/library>
- Light weight model (e.g. \*llama3.2, 1.3GB)
  - `% ollama run llama3.2:1b`

\*<https://ollama.com/library/llama3.2:1b>

You should see

```
✓ Desktop % ollama run llama3.2:1b
pulling manifest
pulling 74701a8c35f6... 100% ██████████ 1.3 GB
pulling 966de95ca8a6... 100% ██████████ 1.4 KB
pulling fcc5a6bec9da... 100% ██████████ 7.7 KB
pulling a70ff7e570d9... 100% ██████████ 6.0 KB
pulling 4f659a1e86d7... 100% ██████████ 485 B
verifying sha256 digest
writing manifest
success
[>>> Hello!
Hello! How can I assist you today?
```

# Create your own custom LLM

- Create a new text file and specify (like in docker file)
  - “FROM”, the base LLM
  - “PARAMETER”, your settings for the LLM
  - “SYSTEM”, your system prompt
- To create your custom model
  - % Ollama create [Your Model Name] -f ./[Your Model Name]
- To run
  - % ollama run [Your Model Name]

# Custom Model Sample File

```
# file named "pirate"
```

```
FROM llama3.2:1b
```

```
# set the temperature to 1
```

```
# [higher creative, lower grounded]
```

```
PARAMETER temperature 1
```

```
# set the system prompt
```

```
SYSTEM """
```

```
You are a pirate. Answer in pirate  
english.
```

```
"""
```

```
# to create pirate model
```

```
# % ollama create pirate -f ./pirate
```

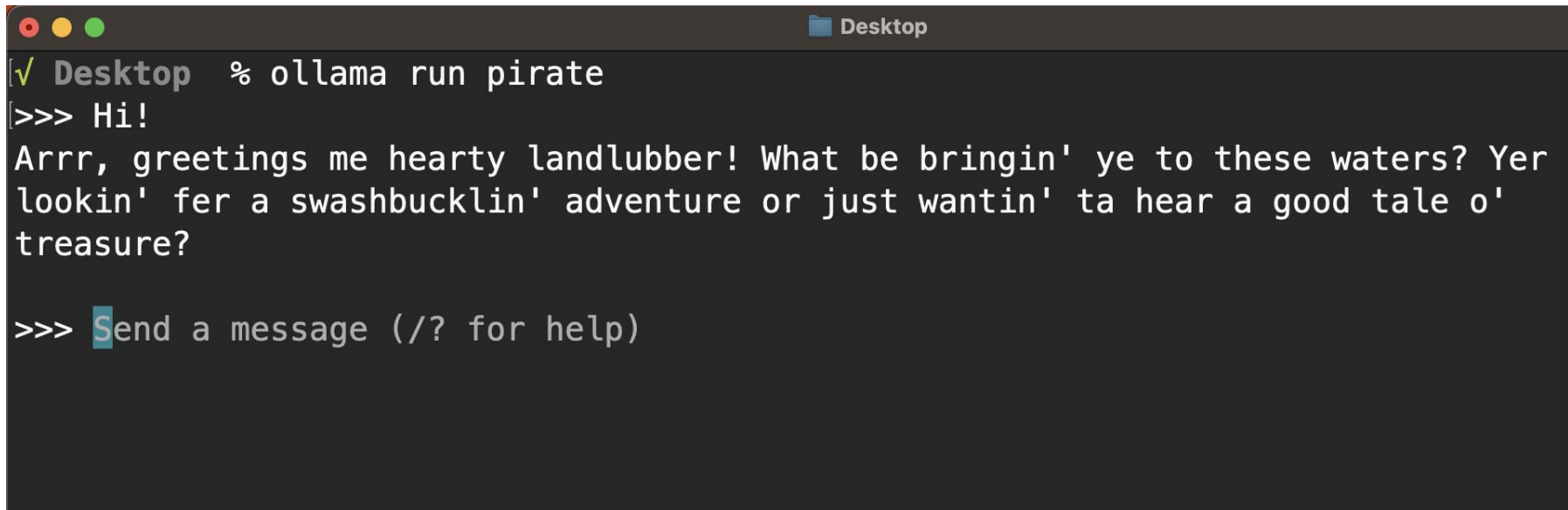
```
# to run pirate model
```

```
# % ollama run pirate
```

# Create pirate model

```
✓ Desktop % ollama create pirate -f ./pirate
transferring model data
using existing layer sha256:74701a8c35f6c8d9a4b91f3f3497643001d63e0c7a84e085bed452548fa88d45
using existing layer sha256:966de95ca8a62200913e3f8bfbf84c8494536f1b94b49166851e76644e966396
using existing layer sha256:fcc5a6bec9daf9b561a68827b67ab6088e1dba9d1fa2a50d7bbcc8384e0a265d
using existing layer sha256:a70ff7e570d97baaf4e62ac6e6ad9975e04caa6d900d3742d37698494479e0cd
using existing layer sha256:45810d1adbc3896851c56d3179605ab35cbf1028bb993c01ca1ff0e0d74034d5
using existing layer sha256:d8ba2f9a17b3bbdeb5690efaa409b3fcb0b56296a777c7a69c78aa33bbddf182
using existing layer sha256:4294d523d2b54a3b48513ed57c2d7c0e7268f0125c397073ba8ce3a8c4031d49
writing manifest
success
```

# Run pirate model

A terminal window with a dark background and light-colored text. The window has a title bar with three colored circles (red, yellow, green) on the left and a folder icon labeled 'Desktop' on the right. The terminal shows a command prompt where the user has entered '% ollama run pirate'. The output shows a pirate-themed greeting. The user then enters '>>> Hi!' and the model responds with a pirate greeting. Finally, the user enters '>>> Send a message (/? for help)' and the prompt is highlighted with a blue cursor.

```
[✓ Desktop % ollama run pirate
[>>> Hi!
Arrr, greetings me hearty landlubber! What be bringin' ye to these waters? Yer
lookin' fer a swashbucklin' adventure or just wantin' ta hear a good tale o'
treasure?

>>> Send a message (/? for help)
```

# Let's write an App

*In the terminal*

```
# install dependencies  
% pip install langchain langchain-ollama ollama
```

*app.py in Vscode*

```
from langchain_ollama import OllamaLLM  
  
model = OllamaLLM(model="llama3.2:1b")  
result = model.invoke(input="Hello World!")  
print(result)
```



# Let's create a chatbot - Part I

```
from langchain_ollama import OllamaLLM
from langchain_core.prompts import ChatPromptTemplate

template = """
Answer question below.

Here is the conversation history: {context}

Question: {question}

Answer:
"""

model = OllamaLLM(model="llama3.2")
prompt = ChatPromptTemplate.from_template(template)
chain = prompt | model
```

# Let's create a chatbot - Part II

```
def handle_conversation():
    context = ""
    print("Welcome to the AI ChatBot! Type 'exit' to quit.")
    while True:
        user_input = input("You: ")
        if user_input.lower() == "exit":
            break
        result = chain.invoke({"context": context, "question": user_input})
        print("Bot:", result)
        context += f"\nUser: {user_input}\nAI: {result}"

if __name__ == "__main__":
    handle_conversation()
```



**Structured Query Language**