

Project Progress Report

By: Jeremy Flagg

Spring Semester 2025

1. Study Overview

This study focuses on:

- Training Merjek AI models on a GPU cluster.

2. Early Steps & Prompt Generation

The initial phase involved testing different LLM models for prompt generation and analyzing their outputs after database insertion.

3. Models Tested

Several models were tested for effectiveness and performance:

- Open-source models (e.g., LLama 3.1 8B, DeepSeek R1 1.5B, Mistral 7B v0.3)

4. GPU Cluster Specifications

Cluster Quota specifications:

- Max Jobs: 6
- Max Nodes: 3
- Max GPUs per Job: 4
- Max Runtime per Job: 48 hours

Training Progress:(1/24)

- Initial meeting

Training Progress: (2/7)

- Installation of Ollama and different open-source LLM models.
- Prompt generation and insertion into MySQL Workbench.

Training Progress:(2/14-2/28)

- Initial training/test practice, locally and in GPU Cluster,with Human Trafficking and Campus csv files.
- Migration to MongoDB Atlas/Compass

Training Progress: (3/7/25)

- Dataset: 2,000 documents (subset of 10K)
- Split: 80% train, 20% test
- Tested on 2 GPUs (1 node)
- Estimated training time: ~52 minutes for 1 epoch

Training Progress: (3/14/25)

- Created Merjek Github
- Prompt generation 5 hours per 1000 documents
- Meeting at library helping Md with MongoDB setup and prompt generation
- Continue generating ~8K prompts for the entire dataset of ~10K documents.
- Mistral 7B v0.3 is the model used for prompt generation. (LM Studio on my Windows setup)
- After generation, iterated through MongoDB collection to add prompts into arrays.
- Edit Slurm training script before executing within GPU cluster.
- Scaled training from 2,000 docs at 1 epoch to 10,000 docs at 3 epochs.

 **Loaded 305835 valid prompts from the first 10,000 documents.**

Training samples: 244668

Validation samples: 61167

 **Using device: cuda, Batch size: 16**

GPU #: 4

Estimated train time for 1 epoch: 4 hours 41 minutes

Estimated train time for 3 epochs: 14 hours 4 minutes

View inside cluster after 1 epoch for 10K docs:

```
PS C:\WINDOWS\system32> ssh jmlagg@itiger.memphis.edu
jmlagg@itiger.memphis.edu's password:
Last login: Tue Mar 11 20:17:26 2025 from 10.228.110.243
[jmlagg@itiger ~]$ cd /project/jmlagg/merjek-study/
[jmlagg@itiger merjek-study]$ squeue -u $USER
        JOBID PARTITION     NAME     USER ST       TIME  NODES MODELLIST(REASON)
        5421    bigTiger merjekai    jmlagg  R       2:59:00      1 itiger04
[jmlagg@itiger merjek-study]$ tail merjekai-training-output.txt
{'eval_loss': 8.805888175964355, 'eval_runtime': 43.2928, 'eval_samples_per_second': 1412.866, 'eval_steps_per_second': 22.082, 'epoch': 0.62}
{'loss': 8.8116, 'grad_norm': 208776.609375, 'learning_rate': 7.496730316505363e-06, 'epoch': 0.63}
{'eval_loss': 8.804204940795898, 'eval_runtime': 42.7269, 'eval_samples_per_second': 1431.582, 'eval_steps_per_second': 22.375, 'epoch': 0.63}
{'loss': 8.8209, 'grad_norm': 193645.765625, 'learning_rate': 7.444415380591159e-06, 'epoch': 0.63}
{'eval_loss': 8.803586959838867, 'eval_runtime': 42.5937, 'eval_samples_per_second': 1436.057, 'eval_steps_per_second': 22.445, 'epoch': 0.63}
{'loss': 8.8142, 'grad_norm': 205195.5625, 'learning_rate': 7.392100444676957e-06, 'epoch': 0.63}
{'eval_loss': 8.803701400756836, 'eval_runtime': 43.2752, 'eval_samples_per_second': 1413.442, 'eval_steps_per_second': 22.091, 'epoch': 0.63}
{'loss': 8.8328, 'grad_norm': 210443.34375, 'learning_rate': 7.339785508762752e-06, 'epoch': 0.63}
{'eval_loss': 8.803175926208496, 'eval_runtime': 43.3248, 'eval_samples_per_second': 1411.825, 'eval_steps_per_second': 22.066, 'epoch': 0.63}
{'loss': 8.7729, 'grad_norm': 209517.65625, 'learning_rate': 7.287470572848549e-06, 'epoch': 0.64}
[jmlagg@itiger merjek-study]$ exit
logout
Connection to itiger.memphis.edu closed.
PS C:\WINDOWS\system32> ssh jmlagg@itiger.memphis.edu
jmlagg@itiger.memphis.edu's password:
Last login: Tue Mar 11 21:17:54 2025 from 10.228.110.238
[jmlagg@itiger ~]$ cd /project/jmlagg/merjek-study/
[jmlagg@itiger merjek-study]$ tail merjekai-training-output.txt
{'loss': 8.7498, 'grad_norm': 210646.0, 'learning_rate': 6.800941668846455e-08, 'epoch': 1.0}
{'eval_loss': 8.748345375061035, 'eval_runtime': 42.6307, 'eval_samples_per_second': 1434.811, 'eval_steps_per_second': 22.425, 'epoch': 1.0}
{'loss': 8.7193, 'grad_norm': 204133.09375, 'learning_rate': 1.5694480774261054e-08, 'epoch': 1.0}
{'eval_loss': 8.74834156036377, 'eval_runtime': 42.8912, 'eval_samples_per_second': 1426.097, 'eval_steps_per_second': 22.289, 'epoch': 1.0}
{'train_runtime': 16880.9389, 'train_samples_per_second': 14.494, 'train_steps_per_second': 0.226, 'train_loss': 8.908902582622092, 'epoch': 1.0}
Evaluating model...
Evaluation results: {'eval_loss': 8.74834156036377, 'eval_runtime': 42.6337, 'eval_samples_per_second': 1434.71, 'eval_steps_per_second': 22.424, 'epoch': 1.0}
Saving model to ./fine-tuned-model-merjekai3
[✓] Model and tokenizer saved successfully.
[✓] Training job completed.
```

View inside cluster after 3 epochs for 10K:

```
[jmlagg@itiger ~]$ cd /project/jmlagg/merjek-study/
[jmlagg@itiger merjek-study]$ head merjekai-training-output.txt
[✓] Starting merjekai.py...
[✓] Starting merjekai.py...
[✓] Connected to MongoDB Atlas successfully.
[✓] Loaded 305835 valid prompts from the first 10,000 documents.
Training samples: 244668
Validation samples: 61167
[✓] Using device: cuda, Batch size: 16
Starting training...
{'loss': 9.2462, 'grad_norm': 176753.75, 'learning_rate': 1.99825616880286e-05, 'epoch': 0.0}
{'eval_loss': 9.237972259521484, 'eval_runtime': 43.31, 'eval_samples_per_second': 1412.307, 'eval_steps_per_second': 22.073, 'epoch': 0.0}
[jmlagg@itiger merjek-study]$ tail merjekai-training-output.txt
{'loss': 8.2178, 'grad_norm': 242138.140625, 'learning_rate': 3.3132792745662224e-08, 'epoch': 3.0}
{'eval_loss': 8.314220428466797, 'eval_runtime': 43.37, 'eval_samples_per_second': 1410.351, 'eval_steps_per_second': 22.043, 'epoch': 3.0}
{'loss': 8.2782, 'grad_norm': 253271.640625, 'learning_rate': 1.5694480774261054e-08, 'epoch': 3.0}
{'eval_loss': 8.314230918884277, 'eval_runtime': 43.1083, 'eval_samples_per_second': 1418.914, 'eval_steps_per_second': 22.177, 'epoch': 3.0}
{'train_runtime': 50719.8742, 'train_samples_per_second': 14.472, 'train_steps_per_second': 0.226, 'train_loss': 8.52074397049928, 'epoch': 3.0}
Evaluating model...
Evaluation results: {'eval_loss': 8.31423282623291, 'eval_runtime': 42.679, 'eval_samples_per_second': 1433.188, 'eval_steps_per_second': 22.4, 'epoch': 3.0}
Saving model to ./fine-tuned-model-merjekai3
[✓] Model and tokenizer saved successfully.
[✓] Training job completed.
[jmlagg@itiger merjek-study]$
```

View inside MongoDB Compass:

Documents10.1K

Aggregations

Schema


Indexes1


Validation


{Label:10000}

Generat

+ ADD DATA ▾

 EXPORT DATA ▾

 UPDATE

 DELETE

```
_id: ObjectId('67be90d2e152ac3375cc4939')
Label : 10000
Url : "https://www.memphis.edu/gradschool/resources/graduate_faculty/cas/swrk..."
Title : "Social Work Graduate Faculty Resources -
        Graduate School
        - The Un..."
Text : "

        Social Work Graduate Faculty Resources -
        Graduate School
        ..."
  Client : Object
  Prompts : Array (30)
    0: "university memphis campus"
    1: " university memphis academic calendar"
    2: " university memphis admissions process"
    3: " university of memphis faculty members"
    4: " university memphis degrees"
    5: " university memphis school of social work"
    6: " university of memphis application deadline"
    7: " university of memphis faculty positions"
    8: " university memphis doctoral programs"
    9: " university of memphis external graduate faculty"
    10: " university of memphis financial aid"
```

Training Progress: (3/21/25 - 3/28/25)

Goal: Find a way to utilize the GPU cluster for prompt generation

1. Initial Attempt with vLLM

- **Tried using vLLM for running LLaMA 3.1 8B.**
 - **Faced challenges and decided to move on to other methods.**
-

2. Transformers Library Approach

- **Attempted to use the Transformers library and download LLaMA 3.1 8B from HuggingFace.**
 - **Performance was poor, comparable to 1B models.**
 - **Assumed Ollama optimizes models behind the scenes for better performance.**
-

3. Dockerized Ollama Installation

- **Installed a Dockerized version of Ollama on the cluster using Podman (compatible with Docker).**
 - **Downloaded and tested the LLaMA 3.2B model.**
 - **The model worked but encountered two major issues:**
 - **No External Access: Unable to access Ollama from outside the container, even though ports were open and listening.**
 - **CPU-Only Inference: Without external access, couldn't create a Slurm script for GPU usage, resulting in CPU-only inference.**
-

4. Native Ollama Installation

- **Installed Ollama natively by downloading and extracting the binary to a directory.**

- Still faced CPU-only inference since GPUs on the cluster are only accessible through Slurm.
-

5. Understanding Cluster Architecture

- Gained better insight into the cluster's architecture:
 - Head Node: Where users log in, but no GPUs are available.
 - Worker Nodes: GPUs are only available on these nodes through Slurm jobs.
 - Confirmed that Ollama needs to run on worker nodes for GPU access.
-

6. Issues with Docker GPU Pass-Through

- Directly using Docker containers for Ollama with GPU pass-through was unsuccessful.
 - The cluster's GPU access is restricted, and Docker doesn't support direct GPU usage in this environment.
-


7. Solution: Using Apptainer

- Identified Apptainer (formerly Singularity) as the only functional solution for GPU access on the cluster.
- Apptainer is installed on the cluster and supports GPU pass-through.
- This method allows Ollama to run with GPU acceleration via Slurm jobs.

Prompt Generation Analysis

 Database: **crawled_cs_pages2**

 Total Documents: ~548 docs

 Total Time: 410 seconds | 6m 50s

Model Tested: Llama 3.2:3b

```
'grants; University of Memphis cybersecurity degree; University of '
'Memphis online courses; University of Memphis academic misconduct '
'policy; University of Memphis student conduct code; University of '
'Memphis data science program; University of Memphis computer '
'engineering; University of Memphis artificial intelligence; '
'University of Memphis IT department contact; University of '
'Memphis student organization; University of Memphis club sports; '
'University of Memphis campus map; University of Memphis parking '
'rules; University of Memphis library hours; ',
'Url': 'https://www.memphis.edu/cs/courses/syllabi/7900.pdf',
'_id': ObjectId('677ff6dcae52426f563c996'))}
-----

Processing document with _id: 677ff6dcae52426f563c998...

Generated LLama Prompts:
{'Label': 536,
 'Processing Time (s)': 0.68,
 'Prompts': 'University of Memphis computer science; University of Memphis AI '
            'research; University of Memphis admission requirements; '
            'University of Memphis data science program; Kan Yang course '
            'website; COMP 7/8998 university; University of Memphis cloud '
            'security course; MEMPHIS CS department; COMP 7998 grading policy; '
            'University of Memphis plagiarism policy; University of Memphis '
            'disability services; Cloud computing security; Internet of Things '
            'security; Attribute-based access control; Efficient search over '
            'encrypted data; Fog computing; Crowdsourcing authentication; '
            'Blockchain introduction; University of Memphis computer science '
            'courses; MEMPHIS graduate programs; University of Memphis online '
            'courses; COMP 7/8998 course description; University of Memphis '
            'research centers; Computer science department at Memphis.',
 'Url': 'https://www.memphis.edu/cs/courses/syllabi/7998.pdf',
 '_id': ObjectId('677ff6dcae52426f563c998'))}
-----

✅ Document processing job completed.
• Job completed in 410 seconds.
[jmflagg@itiger ollama]$
```


Ollama Setup and GPU Usage on Cluster

 Check for Apptainer Installation

which apptainer

/usr/bin/apptainer

 Download and Convert Ollama Docker Image to SIF

apptainer pull docker://ollama/ollama:latest

- **Converts Docker image to SIF format for Apptainer compatibility.**

 Verify SIF File

ls -lh *.sif

Example Output:

-rwxr-xr-x 1 jmflagg users 1.7G Mar 26 18:44 ollama_latest.sif



Cluster Management



Check Node Status

sinfo

- **Provides general info about available nodes.**



Detailed Node Information

sinfo -N -o "%N %P %C %G %T %M %E"

- Displays node-specific details like CPU, GPU availability, state, and errors.
-

Running Jobs on Specific Nodes

SSH into Cluster

```
ssh jmflagg@itiger.memphis.edu
```

Navigate to Project Directory

```
cd /project/jmflagg/ollama
```

Submit a Job to Specific Node

```
srun --partition=bigTiger --nodelist=itiger03 --gres=gpu:1 --mem=64G  
--time=1:00:00 --pty bash
```

- Requests a GPU node for one hour with 64GB memory.

Check Running Jobs

```
squeue -u $USER
```

Monitor GPU Status

```
nvidia-smi
```

- Displays real-time GPU usage and memory allocation.

Submit Job with SBATCH to specific node

```
sbatch --nodelist=itiger03 app-job.sh
```

Install Ollama Natively

Download Ollama Binary

```
curl -L https://ollama.com/download/ollama-linux-amd64.tgz -o  
ollama-linux-amd64.tgz
```

Extract and Install

```
mkdir -p ~/ollama
```

```
tar -xzf ollama-linux-amd64.tgz -C ~/ollama
```

Add Ollama to Path

```
export PATH=$HOME/ollama/bin:$PATH
```

Confirm installation

```
ollama --version
```

Python file for Prompt Generation

```
import os

import ollama

import pprint

import time

from pymongo import MongoClient


# MongoDB connection details

MONGO_URI =
"mongodb+srv://jeremy-flagg:ewK4Bo7sR8bhDMn5@merjekcluster1.mwxms.mongodb.net/?re
tryWrites=true&w=majority&appName=MerjekCluster1"

DATABASE_NAME = "merjekaidb"

COLLECTION_NAME = "crawled_cs_pages2"

OLLAMA_HOST = "http://127.0.0.1:11434" # Use localhost for Ollama

OLLAMA_MODEL_NAME = "llama3.2:3b"


# Set the OLLAMA_HOST environment variable before calling Ollama

os.environ["OLLAMA_HOST"] = OLLAMA_HOST


def generate_prompts_with_ollama(text):

    try:

        # Construct the prompt for Ollama API

        prompt = f"""

        You are extracting **up to 40 highly relevant search queries** from the
```

```
given text. **Do not summarize, analyze, or paraphrase.**
```

```
### **Instructions:**
```

- Each query must be ****1-8 words long**** and relate to the ****"University of Memphis."****
- Queries should be ****keywords, short questions, or phrases**** (not full sentences).
- ****Strictly separate each query with a semicolon (;)**** on a single line.
- ****Do not add newlines, explanations, numbers, or extra formatting.****

```
### **Text to Process:**
```

```
{text}
```

```
### **Output Format (Example):**
```

```
University of Memphis computer science; University of Memphis AI research;  
University of Memphis admission requirements; University of Memphis data science  
program
```

```
### **Your Output:**
```

```
(Ensure queries are in one single line, semicolon-separated, no newlines)
```

```
"""
```

```
# Now call Ollama without specifying the host directly
```

```
output = ollama.generate(model=OLLAMA_MODEL_NAME, prompt=prompt)
```

```

response = output.get('response', "").strip()

if not response:

    raise ValueError("No response received from Ollama API.")

# Ensure response is formatted correctly

formatted_response = response.replace("\n", " ").strip()

# Validate format: Must contain semicolon-separated queries

if ";" not in formatted_response:

    raise ValueError(f"Unexpected response format from LLM:
{formatted_response}")

return formatted_response

except Exception as e:

    print(f"Error generating prompts: {e}")

    return ""

def connect_to_mongodb():

    try:

        # Connect to MongoDB

        db_client = MongoClient(MONGO_URI)

```

```

        print("Connected to MongoDB.")

        return db_client[DATABASE_NAME]

    except Exception as e:

        print(f"Error connecting to MongoDB: {e}")

        return None

def process_documents(collection, start_label=1):

    for document in collection.find({"Label": {"$gte": start_label}}).limit(10):

        text = document.get("Text", "")

        if not text or len(text) < 15:

            print(f"Skipping document with _id: {document['_id']} due to missing
or short 'Text' field.")

            continue

        print(f"\nProcessing document with _id: {document['_id']}...")

        start_time = time.time()

        prompts = generate_prompts_with_ollama(text)

        end_time = time.time()

        processing_time = round(end_time - start_time, 2)

        # Pretty-print the result

        print("\nGenerated LLama Prompts:")

        pprint.pprint({

```

```
        "_id": document["_id"],

        "Label": document["Label"],

        "Url": document.get("Url", "N/A"), # Print URL or "N/A" if it
doesn't exist

        "Prompts": prompts,

        "Processing Time (s)": processing_time

    })

    print("-" * 80)

if __name__ == "__main__":

    db = connect_to_mongodb()

    if db is not None:

        collection = db[COLLECTION_NAME]

        process_documents(collection)

    else:

        print("Unable to connect to database. Exiting.")
```


Slurm Script

```
#!/bin/bash

#SBATCH --job-name=olam-gen

#SBATCH --output=olam-output.txt

#SBATCH --error=olam-error.txt

#SBATCH --partition=bigTiger

#SBATCH --nodes=1

#SBATCH --ntasks=1

#SBATCH --gres=gpu:2

#SBATCH --mem=64G

#SBATCH --time=48:00:00


export OLLAMA_HOST=http://0.0.0.0:11434      # All connections


apptainer run --nv --bind /project/jmflagg/ollama/data:/data --bind
/project/jmflagg/ollama:/ollama ollama_latest.sif serve &


# Give the container a few seconds to start up

sleep 5


# Confirm Ollama is accessible

echo "📝 Checking Ollama connection..."

curl -s $OLLAMA_HOST || { echo "❗ Failed to connect to Ollama. Ensure it is
running on the node. Exiting."; exit 1; }
```

```
# Activate virtual environment

source /project/jmflagg/ollama/ollama_env/bin/activate # Install ollama and
pymongo libraries with pip in your venv


# Run the Python script

echo "🚀 Starting document processing with Ollama..."

python /project/jmflagg/ollama/cluster_gen2.py          # Whatever your python
file name is


# Deactivate virtual environment after the job is done

deactivate

echo "✅ Document processing job completed."
```

Training Progress: (4/4/25)

Goal: Find a way to use multi-gpus for generation


Success/failure: Failure. The 4 GPUs(RTX 6000 on node 06) are recognized, but Ollama fails to generate prompts using all 4.

Learned: Ollama does not natively support multi-GPU inference and needs external tools like PyTorch, TensorFlow, or HuggingF Transformers to enable it. Still need to test these.

10K Prompt Generation Analysis

 **Database: merjekaidb.uofm_pages_crawled**

 **Total Documents: ~10,048 docs**

 **Total Time: ~ 4 hours, 30 mins**

Model Tested: Llama 3.2:3b

Python Code Implementation:

```
import os

import ollama

import pprint

import time

from pymongo import MongoClient


# MongoDB Atlas connection details

MONGO_URI =
"mongodb+srv://jeremyflagg12:QGTrn51bWa2qrXFL@cluster0.t4orq.mongodb.net/?retryWr
ites=true&w=majority&appName=Cluster0"

DATABASE_NAME = "Prompts"

COLLECTION_NAME = "merjekai4"
```

```

OLLAMA_HOST = "http://127.0.0.1:11434" # Use localhost for Ollama

OLLAMA_MODEL_NAME = "llama3.2:3b"


# Set the OLLAMA_HOST environment variable before calling Ollama
os.environ["OLLAMA_HOST"] = OLLAMA_HOST


BATCH_SIZE = 100 # Process in small batches to avoid cursor timeouts


def generate_prompts_with_ollama(text):
    try:
        prompt = f"""

        You are extracting **up to 40 highly relevant search queries** from the
given text...

        """

        output = ollama.generate(model=OLLAMA_MODEL_NAME, prompt=prompt)
        response = output.get('response', "").strip()

        return response.replace("\n", " ").strip() if response else ""
    except Exception as e:
        print(f"Error generating prompts: {e}")
        return ""


def connect_to_mongodb():
    try:
        db_client = MongoClient(MONGO_URI)

        print("Connected to MongoDB Atlas.")

        return db_client[DATABASE_NAME]

```

```

except Exception as e:

    print(f"Error connecting to MongoDB Atlas: {e}")

    return None

def process_documents(collection):

    last_id = None # Track last processed document

    while True:

        query = {"$or": [{"Prompts": {"$size": 0}}, {"Prompts": {"$exists":
False}}]}

        if last_id:

            query["_id"] = {"$gt": last_id} # Pagination using _id

        documents = list(collection.find(query).sort("_id").limit(BATCH_SIZE))

        if not documents:

            break # No more documents left

        for document in documents:

            try:

                label = document.get("Label", "UNKNOWN")

                text = document.get("Text", "")

                if not text or len(text) < 15:

                    collection.update_one({"_id": document["_id"]}, {"$set":
{"Prompts": ["INSUFFICIENT_TEXT"]}})

                    continue

            start_time = time.time()

```

```

        prompts = generate_prompts_with_ollama(text)

        processing_time = round(time.time() - start_time, 2)

        collection.update_one(
            {"_id": document["_id"]},
            {"$set": {"Processing Time (s)": processing_time, "Prompts":
prompts.split("; ")}}
        )

        last_id = document["_id"] # Update last processed ID

        time.sleep(2) # Prevent rate limits

    except Exception as e:

        print(f"Error processing document {document.get('Label',
'UNKNOWN')}: {e}")

        collection.update_one({"_id": document["_id"]}, {"$set":
{"Prompts": ["PROCESSING_ERROR"]}})

        time.sleep(5)

if __name__ == "__main__":

    db = connect_to_mongodb()

    if db:

        collection = db[COLLECTION_NAME]

        process_documents(collection)

    else:

        print("Unable to connect to database. Exiting.")

```

Slurm Script

```
#!/bin/bash

#SBATCH --job-name=ten-gen

#SBATCH --output=ten-output.txt

#SBATCH --error=ten-error.txt

#SBATCH --partition=bigTiger

#SBATCH --nodes=1

#SBATCH --ntasks=1

#SBATCH --gres=gpu:4

#SBATCH --time=48:00:00

#SBATCH --odelist=itiger07


export OLLAMA_HOST=http://0.0.0.0:11434


apptainer run --nv --bind /project/jmflagg/ollama/data:/data --bind /project/jmflagg/ollama:/ollama ollama_latest.sif serve &


# Give the container a few seconds to start up

sleep 5


# Confirm Ollama is accessible

echo "🔍 Checking Ollama connection..."

curl -s $OLLAMA_HOST || { echo "❗ Failed to connect to Ollama. Ensure it is running on the node. Exiting."; exit 1; }


# Activate virtual environment

source /project/jmflagg/ollama/ollama_env/bin/activate


# Run the Python script

echo "🚀 Starting document processing with Ollama..."

python /project/jmflagg/ollama/prompt_gen.py
```

```
# Deactivate virtual environment after the job is done  
deactivate
```

This command is important for checking which node has enough resources to allocate to your specific needs.

Detailed Node Information

```
sinfo -N -o "%N %P %C %G %T %M %E"
```

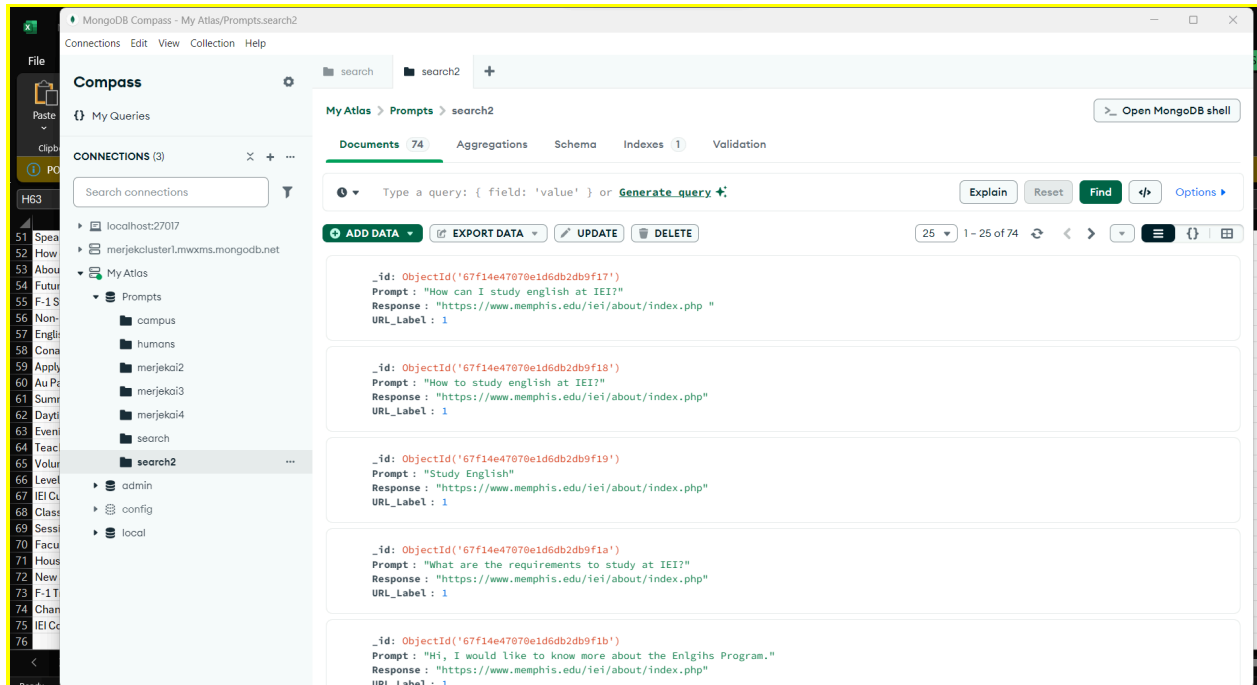
```
[jmflagg@itiger ollama]$ sinfo -N -o "%N %P %C %G %T %M %E"  
  
NODELIST PARTITION CPUS(A/I/O/T) GRES STATE PREEMPT_MODE REASON  
itiger01 bigTiger* 36/28/0/64 gpu:h100_80gb:8(S:0-1) mixed OFF none  
itiger02 bigTiger* 16/48/0/64 gpu:rtx_6000:8(S:0-1) mixed OFF none  
itiger03 bigTiger* 4/60/0/64 gpu:rtx_6000:8(S:0-1) mixed OFF none  
itiger04 bigTiger* 64/0/0/64 gpu:rtx_6000:8(S:0-1) allocated OFF none  
itiger05 bigTiger* 64/0/0/64 gpu:rtx_6000:8(S:0-1) allocated OFF none  
itiger06 bigTiger* 40/24/0/64 gpu:rtx_6000:8(S:0-1) mixed OFF none  
itiger07 bigTiger* 0/64/0/64 gpu:rtx_5000:8(S:0-1) idle OFF none  
itiger08 bigTiger* 0/64/0/64 gpu:rtx_5000:8(S:0-1) idle OFF none  
itiger09 bigTiger* 0/64/0/64 gpu:rtx_5000:8(S:0-1) idle OFF none  
itiger10 bigTiger* 0/64/0/64 gpu:rtx_5000:8(S:0-1) idle OFF none  
itiger11 bigTiger* 0/64/0/64 gpu:rtx_5000:8(S:0-1) idle OFF none
```


Progress 4/11

Goal: Train model on IEI CSV and build search engine

Step 1:

Create labels for each unique URL and import into MongoDB



Step 2: BERT Model

Initial results after training on 3 epochs (Realized some labels were wrong, so I had to remake dataset. Still inaccurate after 10 epochs)

```

1 from datasets import Dataset
2 import pandas as pd
3 from sklearn.model_selection import train_test_split
4 from transformers import (
5     BertTokenizer,
6     DataCollatorWithPadding,
7     BertForSequenceClassification,
8     Trainer,
9     TrainingArguments
10 )
11 import torch
12 from pymongo import MongoClient
13 import numpy as np
14 import evaluate

```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```

Evaluation Results: {'accuracy': 0.26397800183318054, 'f1': 0.11026202252060334}
100% | 273/273 [15:09<00:00, 3.33s/it]
Evaluation Results: {'accuracy': 0.26397800183318054, 'f1': 0.11026202252060334}
Evaluation Results: {'accuracy': 0.26397800183318054, 'f1': 0.11026202252060334}
• Test predictions saved to test_predictions_output.csv
(venv) PS C:\Users\jerem\Desktop\merjek-study> python '.\test_model_local.py'
Device set to use cpu

User input: How can I study english at IET?
Predicted label: 2
Confidence: 0.2256
True label: 1

• Associated URL: https://www.memphis.edu/iei/about/index.php
(venv) PS C:\Users\jerem\Desktop\merjek-study> python '.\test_model_local.py'
Device set to use cpu

User input: How much?
Predicted label: 2
Confidence: 0.2893
True label: 4

• Associated URL: https://www.memphis.edu/iei/about/fees.php
(venv) PS C:\Users\jerem\Desktop\merjek-study> nvcc --version

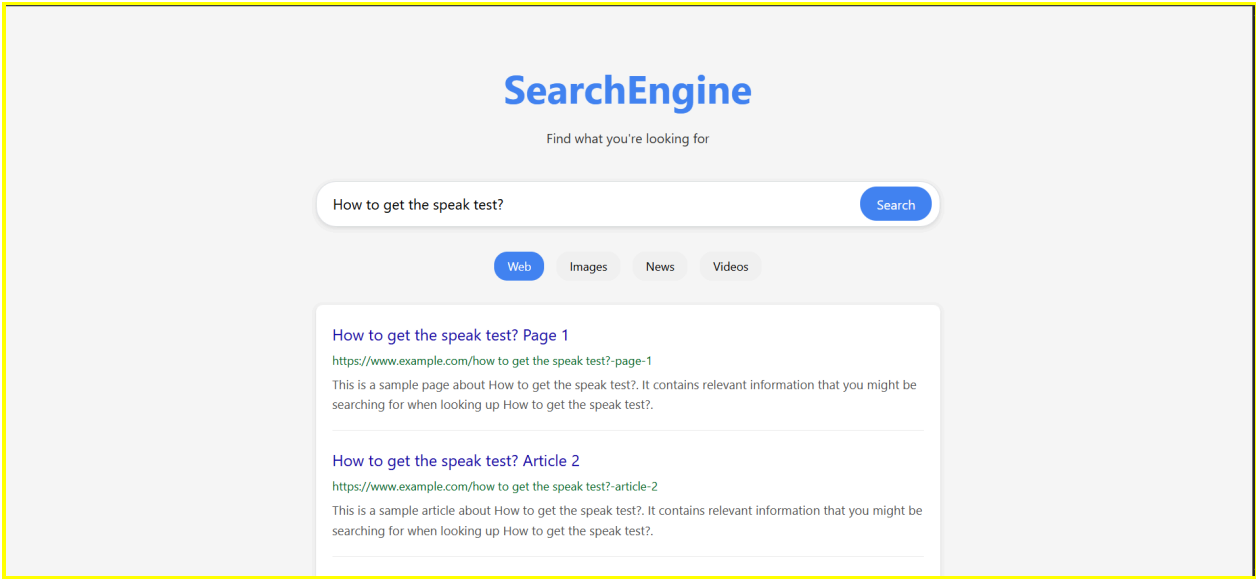
```

Ln 4, Col 27 Spaces: 4 UTF-8 CRLF

Step 3: DistilBERT Model : 15 epoch

DistilBERT is smaller and offers faster training when it comes to small datasets, but it is still struggling with accuracy. Virtually no improvement compared to the bigger BERT model.

Front End Debut:



U of M pages crawled prompt generation sample:

Duration: ??? Ended early when bug was revealed

Model Tested: Llama 3.2: 3b

Bug: Ollama makes generate() API call successfully, but refuses to generate prompt inside DB starting near prompt 4800 and onwards.

ADD DATA EXPORT DATA UPDATE DELETE 100 1 - 100 of 9000

▼ Prompts : Array (4)

- 0: "What is the University of Memphis computer science department's curric..."
- 1: "what types of projects does the University of Memphis AI research grou..."
- 2: "what are the admission requirements for University of Memphis graduate..."
- 3: "what kind of specializations or concentrations does the University of ..."

_id: ObjectId('67f1694cd8570f5e9d2591c2')

LabelNo: 45

Url: "https://www.memphis.edu/student-affairs/index.php"

Title: "Division of Student Affairs -
Student Affairs
- The University of ..."

Text: "

Division of Student Affairs -
Student Affairs
- The Univ..."

► **Client:** Object

Processing Time (s): 1.06

▼ **Prompts:** Array (23)

- 0: "University of Memphis computer science"
- 1: "University of Memphis AI research"
- 2: "University of Memphis admission requirements"
- 3: "University of Memphis data science program"

Progress 4/18

Goal: Find success with search engine from previous week

Using a new Approach: Simple Classification Models

This approach maps user prompts (natural language queries) directly to predefined categories, which in my case are URLs.

Core Model & Pipeline

I'm using:

- **TF-IDF Vectorization:** Transforms text into numerical features based on how important words (or characters) are across the dataset.
- **Character n-grams:** Specifically `char_wb` 3–5 length — which captures substrings, helping with misspellings and short phrases.
- **Linear SVM (Support Vector Machine):** A strong, simple classifier that works well for small-to-medium-sized text classification problems.

Refinements Being Used

1. Text Cleaning

- Lowercasing
- Removing punctuation
- (Optionally: can also remove stopwords or apply lemmatization)

2. Character-Level TF-IDF

- Instead of word n-grams, you're using character n-grams, which is especially helpful for:
 - Spelling variations

- Short queries

- Language noise

3. Rare Label Grouping

- Any URL with fewer than 3 examples is grouped into a generic “other” class
- Prevents overfitting and improves generalization

4. Label Encoding

- Converts labels (URLs or class IDs) into a format suitable for the classifier

5. Cross-validation & Hyperparameter Tuning

- I used GridSearch to find the best C value for the SVM

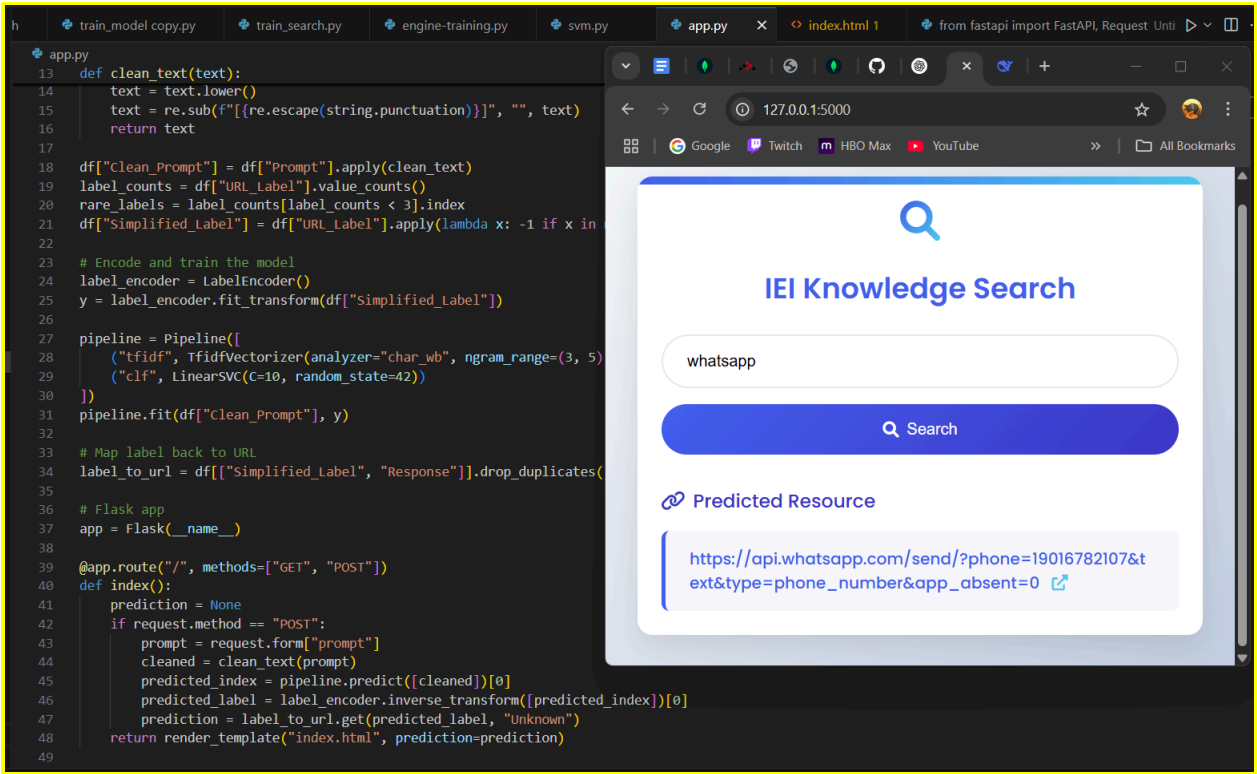
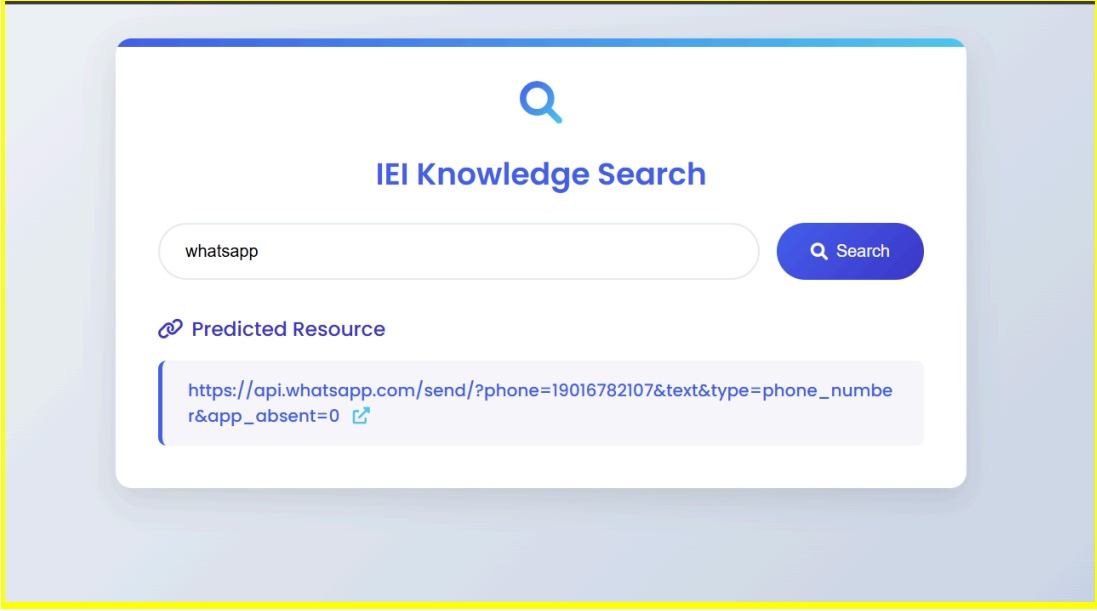
Result

With these refinements:

- Managed to boost accuracy from ~41%(Logistics Regression model) to ~68% (Linear SVM model)
- Created a Flask-based search engine

(Models tested before concluding to use Linear SVM)

Model	Accuracy	Std. Dev
Linear SVM	50.1%	4.4%
Random Forest	44.7%	4.0%
Logistic Regression	41.9%	7.8%
Naive Bayes	40.6%	6.1%
Decision Tree	23.0%	8.2%



Generated extra prompts for each IEI URL using Ollama and stored them in a secondary CSV file.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
51	Speak Test	https://www.memphis.edu/iei/speaktest/ind	11	memphis.edu speak test website; online english placement; iei speak test website; free english placement test; memphis state university english; college english language														
52	How can I get the speak test?	https://www.memphis.edu/iei/speaktest/ind	11	memphis speech and language pathology test; speech evaluation at memphis university; online speech tests for adults; memphis speech therapy programs; free speech a														
53	About IEI	https://www.memphis.edu/iei/about/index.p	1	https://www.memphis.edu/iei/about;memphis institute of engineering;institute for engineering education;engineering education programs;memphis university engineeri														
54	Future Students	https://www.memphis.edu/iei/writing-1.php	12	memphis writing center tips; college essay writing guides; academic writing style; tips for writing thesis; essay format template; writing research papers; writing a cover let														
55	F-1 Student information	https://www.memphis.edu/iei/f1students.ph	8	https://www.memphis.edu/iei/f1studentadvising.php;f1_student_advisor;international_students_memphis;memphis_jenterprise_innovation_center;f1_student_resource														
56	Non-F1 Student information	https://www.memphis.edu/iei/nonf1student:	13	memphis non-f1 student admissions information; non-citizen international student tuition; nonimmigrant international students; international student scholarships memp														
57	English Conditional Admission	https://www.memphis.edu/iei/goingon.php	14	mips instructions for going on; intel edu resources; memphis engineering education; going on mips tutorial; intel edp going on; memphis engineer program; computer arch														
58	Contact us	https://www.memphis.edu/iei/contact/index	15	https://www.memphis.edu/iei/contact/index.php;memphis edu iei contact;contact us memphis edu;memphis institute of engineering;memphis edu iei;engineering progr														
59	Apply Now	https://apply.memphis.edu/register/?id=26d	16	register; memphis university application; apply online memphis; graduate school admissions memphis; register for memphis course; memphis student portal login; mempo														
60	Au Pair	https://www.memphis.edu/iei/contact/index	15	https://www.memphis.edu/iei/contact;contact us at memphis edu;memphis edu iei contact;iei memphis university;memphis university of mississippi;memphis iei office; i														
61	Summer School	https://memphis.studioabroad.com/index.cf	17	program details; study abroad in memphis; university of tennessee programs; memphis studio abroad; international student opportunities; undergraduate courses; langua														
62	Daytime Intensive Program	https://www.memphis.edu/iei/daytime.php	4	memphis edu iei daytime; nighttime sleep tips for students; iei sleep schedule; academic success tips memphis edu; improve memory tips memphis edu; how to study effe														
63	Evening Courses	https://www.memphis.edu/iei/community.pl	18	memphis edu iei community; community engagement memphis; memphis university community service; memphis education initiative; community involvement memphis; r														
64	Teach/Observe	https://www.memphis.edu/iei/teach/index.p	19	memphis university courses online; teaching english in memphis; online language certification programs; college courses for adults; teach english abroad memphis; langu														
65	Volunteer	https://www.memphis.edu/iei/volunteer/indi	20	volunteer opportunities in Memphis;memphis area volunteer work;Volunteer requirements for Memphis;how to get involved volunteering;Memphis non profit organizations														
66	Level and courses	https://www.memphis.edu/iei/about/levels.i	21	https://www.memphis.edu/iei/about; levels and courses; IIE Memphis; graduate programs; master's degrees; online courses; course catalog; degree requirements; IIE tu														
67	IEI Curriculum	https://www.memphis.edu/iei/about/curricu	9	https://www.memphis.edu/iei/about; curriculum; course; curriculum philosophy; online learning; academic; curriculum design; Memphis; iei curriculum; higher educati														
68	Class Schedule	https://www.memphis.edu/iei/about/schedu	7	https://www.memphis.edu/iei/about/schedule; memphis iei schedule; iei course schedule; iei university of memphis; memphis university iei; iei semester schedule; mem														
69	Session Dates	https://www.memphis.edu/iei/about/session	22	https://www.memphis.edu/iei/about/sessiondates;session dates memphis university;memphis ie summer session;memphis edu iei schedule;university of memphis 2023														
70	Faculty and Staff	https://www.memphis.edu/iei/about/faculty	23	faculty members at memphis institute; faculty research areas; memphis institute of engineering; faculty profiles memphis institute; engineers at memphis institute; faculty														
71	Housing	https://www.memphis.edu/iei/about/housin	24	memphis institute for engineering; housing options; graduate student accommodation; university apartments; off-campus living options; student housing near campus; do														
72	New Students (No US Visa Status)	https://www.memphis.edu/iei/f1_j20.php	25	mfa for international students at memphis; scholarship requirements for international students; transfer credits from abroad memphis; graduate school programs memph														
73	F-1 Transfer In process	https://www.memphis.edu/iei/f1_transfer.ph	26	memphis.edu transfer; f1 student; international students; transfer process; transfer requirements; graduate school transfer; transfer to USA; international student loan; fir														
74	Change of Status	https://www.memphis.edu/iei/change_of_st	27	https://www.memphis.edu/iei/change_of_status.php;change of status procedure; Memphis change of status form;how to update your record;Memphis student status cha														
75	IEI Contact information	https://www.memphis.edu/iei/contact/index	15	https://www.memphis.edu/iei/contact/index.php;memphis edu iei contact;memphis edu contact form;iei contact memphis;memphis contact info;memphis edu iei;iei m														
76																		

Observation: Adding extra prompts increases accuracy. For example, adding 10-15 prompts to different categories increased accuracy from 68% to 74%

But more prompts does not necessarily mean an increase in accuracy every time. It can also DECREASE based on prompt quality and class balance.

```

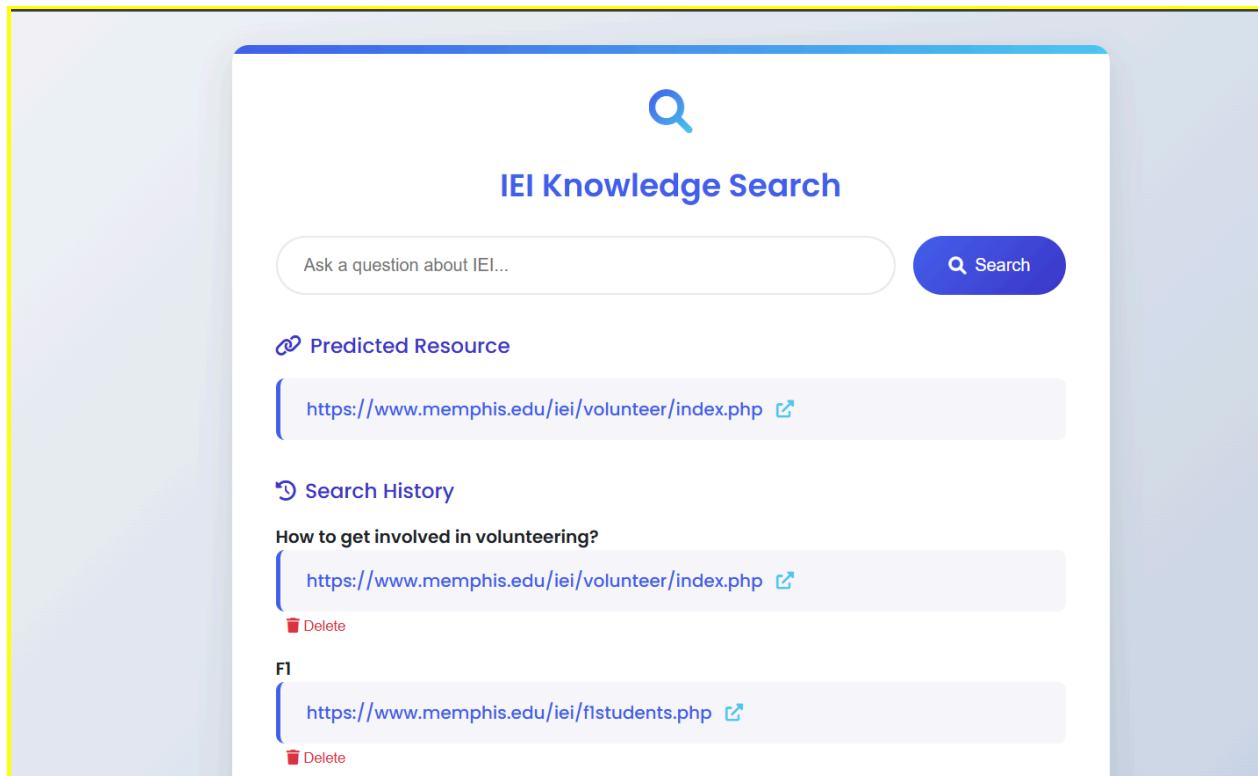
77 # Encode labels
78 label_encoder = LabelEncoder()
79 y = label_encoder.fit_transform(df["Simplified_Label"])
80
81 # Define pipeline

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Press CTRL+C to quit
 * Restarting with stat
 C:\Users\jerem\Desktop\merjek-study\venv\lib\site-packages\sklearn\model_selection_validation.py:100: UserWarning: The number of samples in the validation set is less than n_splits=5.
 warnings.warn(
 Model Accuracy: 0.7405 ± 0.1062

Session-based search history



Technical Summary

This is a supervised text classification pipeline built with **TfidfVectorizer** and **LinearSVC**. Prompts are cleaned and vectorized using character-level n-grams (**char_wb**, 3–5) to help capture word fragments and handle minor variations.

The SVC is trained to predict a label corresponding to a URL. Rare label classes are grouped into a fallback (-1) to reduce overfitting on underrepresented targets. Once the model is trained, it can generalize to unseen prompts as long as they share structural or lexical similarity with the training set.

Strengths

- Learns patterns from labeled examples.
- Handles moderately varied phrasing (e.g., "How do I register for classes?" vs. "Where do I sign up for courses?").

- Fast predictions using a lightweight model.
-

Limitations

- Struggles with semantically very different prompts that share no vocabulary with training data.
- Doesn't measure confidence, so fallback logic isn't yet implemented.
- Currently classification-only — doesn't return top-N options or ranked suggestions.

Overall accuracy level after 137 Database entries:

This is the overall cross-validated accuracy of the model.

For clarification:

- It uses 5-fold cross-validation (**cv=5**).
- The data is split into 5 parts (folds). In each round, 4 parts are used for training and 1 for testing.
- This process runs 5 times so every part is used as a test set once.
- **np.mean(scores)** gives the average accuracy across those 5 folds — the estimated overall accuracy of the model on unseen data.

- `np.std(scores)` shows how much the accuracy varies across those 5 folds (i.e., the standard deviation of accuracy).

0.8169 → Mean accuracy across the 5 folds (i.e., average model performance).

0.0670 → Standard deviation of the accuracies across those folds (i.e., how much the performance varied between folds).

A lower standard deviation is better — it indicates more consistent performance across different parts of the data.

```
17 app.config['SESSION_TYPE'] = 'filesystem'
18 Session(app)
19
20 # Load and prepare the data
21 df = pd.read_csv("iei-prompts2.csv")
22
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Model Accuracy: 0.8169 ± 0.0670
* Debugger is active!
* Debugger PIN: 616-640-226
(venv) PS C:\Users\jerem\Desktop\merjek-study>

Training: 4/25/25

Goal: Redesign app to mimic U of M design.

Transitioned to Bert model on IEI dataset with 132 entries

Observation: After 8 epochs, confidence levels are low, but the correct link was always the first result.

Merjek-AI

Find the best matching resource from your prompt.

Submit

Results for: "I want to register"

Label: 5

Confidence: 34.69%

URL: <https://apply.memphis.edu/register/?id=26d3ab22-2a48-4613-8af9-63091b32d786>

Label: 25

Confidence: 8.83%

URL: https://www.memphis.edu/iei/f1_i20.php

Label: 2

Confidence: 7.36%

URL: https://www.memphis.edu/iei/about/application_process.php

Prompt History

I want to register

Merjek-AI

Find the best matching resource from your prompt.

Submit

Results for: "Tell me about volunteering"

Label: 20

Confidence: 40.20%

URL: <https://www.memphis.edu/iei/volunteer/index.php>

Label: 18

Confidence: 11.95%

URL: <https://www.memphis.edu/iei/community.php>

Label: 24

Confidence: 5.18%

URL: <https://www.memphis.edu/iei/about/housing.php>

Prompt History

Tell me about volunteering

Used data augmentation to increase variety in the IEI dataset, which brought the total amount of entries to 1,001.

Observation after 8 epochs: Confidence levels raised significantly and they are extremely high for each prompt.

Merjek-AI

Find the best matching resource from your prompt.

Submit

Results for: "What are the expenses for IEI?"

Label: 3

Confidence: 98.71%

URL: <https://www.memphis.edu/iei/about/fees.php>

Label: 8

Confidence: 0.17%

URL: <https://www.memphis.edu/iei/f1students.php>

Label: 2

Confidence: 0.14%

URL: https://www.memphis.edu/iei/about/application_process.php

Merjek-AI

Find the best matching resource from your prompt.

Type your prompt here...

Submit

Results for: "Tell me about volunteering"

Label: 20 **Confidence: 98.75%**

URL: <https://www.memphis.edu/iei/volunteer/index.php>

Label: 18 **Confidence: 0.10%**

URL: <https://www.memphis.edu/iei/community.php>

Label: 27 **Confidence: 0.09%**

URL: https://www.memphis.edu/iei/change_of_status.php

Prompt History

Tell me about volunteering

Original redesign for demo



The University of Memphis

INTENSIVE ENGLISH FOR INTERNATIONALS (IEI)

IEI Search Engine

Find the best matching resource from your prompt.

Type your prompt here...

Search

Results for: "I want to know how long the program lasts"

3 results found

7 **Confidence: 96.16%**

URL: <https://www.memphis.edu/iei/about/schedule.php>

4 **Confidence: 2.04%**

URL: <https://www.memphis.edu/iei/daytime.php>

3 **Confidence: 0.22%**

URL: <https://www.memphis.edu/iei/about/fees.php>

Used GPU cluster to train the aggregation of crawled UM data with IEI data

Result: Poor accuracy

```
Some weights of BertForSequenceClassification were not initialized from the model checkpoint (bert_model.ckpt) and are newly initialized from a normal distribution. This warning is thrown if the model was trained on the command line using torch.nn.DataParallel or DataParallel module. You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
/project/jmflagg/comp/bert/lib64/python3.9/site-packages/transformers/optimization.py:429: FutureWarning: torch.optim.AdamW is deprecated in a future version. Use the PyTorch implementation torch.optim.AdamW instead, or set 'no_warnings.warn()' to False.
Epoch 1: 100%|██████████| 31144/31144 [1:17:01<00:00, 6.74it/s, loss=9]
Epoch 2: 100%|██████████| 31144/31144 [1:17:02<00:00, 6.74it/s, loss=8.37]
Epoch 3: 100%|██████████| 31144/31144 [1:16:59<00:00, 6.74it/s, loss=8.7]
Epoch 4: 100%|██████████| 31144/31144 [1:17:00<00:00, 6.74it/s, loss=9.57]
Epoch 5: 100%|██████████| 31144/31144 [1:16:59<00:00, 6.74it/s, loss=8.52]
[jmflagg@itiger comp]$ queue -u jmflagg
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
9752 bigTiger train-be jmflagg R 6:32:06 1 itiger08
```

Much more time efficient script that takes less than 25 minutes to complete, but training results are still poor.

Dataset and/or script needs heavy modification before I can train successfully.

```
er.weight']
You should probably TRAIN this model on a down-stream task to be able to use it for
/project/jmflagg/comp/bert/lib64/python3.9/site-packages/transformers/optimization.py
n a future version. Use the PyTorch implementation torch.optim.AdamW instead, or set
warnings.warn(
Epoch 1: 100%|██████████| 8043/8043 [04:13<00:00, 31.67it/s, loss=8.35]
Epoch 2: 100%|██████████| 8043/8043 [04:14<00:00, 31.63it/s, loss=8.32]
Epoch 3: 100%|██████████| 8043/8043 [04:14<00:00, 31.60it/s, loss=8.36]
[jmflagg@itiger comp]$ cat train-output.txt
🚀 Starting training
Total samples after collapsing: 80425
Unique labels after collapsing: 4022
Label distribution:
  Label
4017    36
4020    28
4019    24
4031    24
4018    24
    ..
4039    12
4014    11
4016    10
1876    10
547     10
Name: count, Length: 4022, dtype: int64
Final Accuracy: 0.0002
```


Progress 5/2/25

Achieved 93% accuracy on newly scraped dataset, which is an incremental improvement of over 40%.

Method: Data augmentation on CSV after pulling the data from API

The screenshot displays the website for The University of Memphis Intensive English for Internationals (IEI). The header features the university's logo and name, along with a navigation menu. Below the header, a secondary navigation bar lists various program options. The main content area is titled "IEI Search Engine" and includes a search bar with the placeholder text "Type your prompt here...". To the left of the search bar is a sidebar with a button labeled "Explore IEI" and a list of links: "About IEI", "Curriculum", "Schedule", "Session Dates", "Fees", "Housing", "Faculty and Staff", and "Au Pair". The search results section shows "Results for: 'What is the contact number on Whatsapp?'" and indicates "5 results found". Two results are visible: the first has a confidence of 81.49% and a title "Share on WhatsApp" with a URL pointing to a WhatsApp share link; the second has a confidence of 9.65% and a title "IEI Contact".

The University of Memphis
INTENSIVE ENGLISH FOR INTERNATIONALS (IEI)

Menu

About Future Students English Conditional Admission SPEAK Test Contact Us Apply Now! Au Pair Summer Program

Explore IEI

About IEI
Curriculum
Schedule
Session Dates
Fees
Housing
Faculty and Staff
Au Pair

IEI Search Engine

Find the best matching resource from your prompt.

Type your prompt here... **Search**

Results for: "What is the contact number on Whatsapp?"

5 results found

6 Confidence: 81.49%
Title: Share on WhatsApp
URL: https://api.whatsapp.com/send/?phone=19016782107&text&type=phone_number&app_absent=0

15 Confidence: 9.65%
Title: IEI Contact

Progress: 5/9/25 FINAL WEEK

My method is 3 splits of the 28k dataset. 10K, 10K, and 8K.

I plan on training the first subset alone to see if BERT learns well.

Examples of prompts being generated :

Prompts.half1

STORAGE SIZE: 36.72MB

LOGICAL DATA SIZE: 87.24MB

TOTAL DOCUMENTS: 10000

INDEXES TOTAL SIZE: 372KB

Find

Indexes

Schema Anti-Patterns 1

Aggregation

Search Indexes

Generate queries from natural language in Compass

Filter

Type a query: { field: 'value' }

Processing Time (s) : 0.47

Prompts : Array (1)

0: "University of Memphis Law School admissions"

_id: ObjectId('68163e6e01a38f9cde5b4866')

LabelNo : "20"

Url : "https://www.memphis.edu/law/programs/"

Title : "Academic Programs - \n School of Law \n

Text : "\n\n\n\n \n\n\n\nAcademic Programs - \n

Processing Time (s) : 0.41

Prompts : Array (1)

0: "University of Memphis Law School programs"

- The University of Memphis"

School of Law \n

- The Unive..."

PREVIOUS

21-40 of many results

Prompts.half1

STORAGE SIZE: 36.72MB LOGICAL DATA SIZE: 87.24MB TOTAL DOCUMENTS: 10000 INDEXES TOTAL SIZE: 372KB

Find

Indexes

Schema Anti-Patterns 0

Aggregation

Search Indexes

[Generate queries from natural language in Compass](#)

Filter

Type a query: { field: 'value' }

QUERY RESULTS: 1-20 OF MANY

```
_id: ObjectId('68163e6e01a38f9cde5b4851')
LabelNo : "0"
Url : "https://www.memphis.edu/"
Title : "The University of Memphis - The University of Memphis"
Text : "\n\n\n\n\n\n\nThe University of Memphis - The University of Memphi..."
Processing Time (s) : 16.36
▼ Prompts : Array (1)
  0: "What are the available programs and activities at the University of Me..."
```

[Generate queries from natural language in Compass](#)

Filter

{LabelNo: "200"}

QUERY RESULTS: 1-1 OF 1

```
_id: ObjectId('68163e6e01a38f9cde5b491d')
LabelNo : "200"
Url : "https://www.memphis.edu/campusrec/aquatics/index.php"
Title : "Aquatics - \n Campus Recreation\n - The University of Memphis"
Text : "\n\n\n\n\n\n\nAquatics - \n Campus Recreation\n - The University ..."
Processing Time (s) : 0.55
▼ Prompts : Array (1)
  0: "What is the University of Memphis's Masters Swim Program called?"
```

Prompts.half1

STORAGE SIZE: 36.71MB LOGICAL DATA SIZE: 87.43MB TOTAL DOCUMENTS: 9999 INDEXES TOTAL SIZE: 372KB

Find

Indexes

Schema Anti-Patterns 0

Aggregation

Search Indexes

Generate queries from natural language in Compass

Filter

{LabelNo: "1000"}

QUERY RESULTS: 1-1 OF 1

_id: ObjectId('68163e7101a38f9cde5b4c50')

LabelNo : "1000"

Url : "https://www.memphis.edu/accounting/pdf/acct05.pdf"

Title : NaN

Text : "Request for Meal Reimbursement\nClick here for online help.\nThe Unive..."

Processing Time (s) : 0.14

▼ Prompts : Array (1)

0: "University of Memphis meal reimbursement policy"

Filter

{LabelNo: "3000"}

QUERY RESULTS: 1-1 OF 1

_id: ObjectId('68163e7701a38f9cde5b545e')

LabelNo : "3000"

Url : "https://www.memphis.edu/research/pdf/umrctaskforce20192020.pdf"

Title : NaN

Text : "University of Memphis Research Council\nOctober 2019\nUMRC\nTask Force..."

Processing Time (s) : 0.18

▼ Prompts : Array (1)

0: "University of Memphis research policies task force"

4K sample dataset that I tested on GPU Cluster:

Took 2hr 40m for prompt generation to finish.

```
GIN] 2025/05/03 - 06:22:35 | 200 | 292.259533ms | 127.0.0.1 | POST | "/api/generate"
GIN] 2025/05/03 - 06:22:37 | 200 | 267.880007ms | 127.0.0.1 | POST | "/api/generate"
GIN] 2025/05/03 - 06:22:39 | 200 | 287.158683ms | 127.0.0.1 | POST | "/api/generate"
GIN] 2025/05/03 - 06:22:42 | 200 | 271.951254ms | 127.0.0.1 | POST | "/api/generate"
GIN] 2025/05/03 - 06:22:44 | 200 | 390.300185ms | 127.0.0.1 | POST | "/api/generate"
GIN] 2025/05/03 - 06:22:46 | 200 | 291.977516ms | 127.0.0.1 | POST | "/api/generate"
GIN] 2025/05/03 - 06:22:49 | 200 | 310.255558ms | 127.0.0.1 | POST | "/api/generate"
GIN] 2025/05/03 - 06:22:51 | 200 | 414.388647ms | 127.0.0.1 | POST | "/api/generate"
GIN] 2025/05/03 - 06:22:54 | 200 | 256.242144ms | 127.0.0.1 | POST | "/api/generate"
GIN] 2025/05/03 - 06:22:56 | 200 | 356.609074ms | 127.0.0.1 | POST | "/api/generate"
GIN] 2025/05/03 - 06:22:58 | 200 | 259.474395ms | 127.0.0.1 | POST | "/api/generate"
GIN] 2025/05/03 - 06:23:01 | 200 | 336.616331ms | 127.0.0.1 | POST | "/api/generate"
GIN] 2025/05/03 - 06:23:03 | 200 | 291.311268ms | 127.0.0.1 | POST | "/api/generate"
GIN] 2025/05/03 - 06:23:05 | 200 | 264.516356ms | 127.0.0.1 | POST | "/api/generate"
GIN] 2025/05/03 - 06:23:08 | 200 | 285.285116ms | 127.0.0.1 | POST | "/api/generate"
GIN] 2025/05/03 - 06:23:10 | 200 | 229.097535ms | 127.0.0.1 | POST | "/api/generate"
GIN] 2025/05/03 - 06:23:12 | 200 | 171.737501ms | 127.0.0.1 | POST | "/api/generate"
GIN] 2025/05/03 - 06:23:14 | 200 | 113.483363ms | 127.0.0.1 | POST | "/api/generate"
GIN] 2025/05/03 - 06:23:17 | 200 | 380.480405ms | 127.0.0.1 | POST | "/api/generate"
GIN] 2025/05/03 - 06:23:19 | 200 | 567.043058ms | 127.0.0.1 | POST | "/api/generate"
GIN] 2025/05/03 - 06:23:22 | 200 | 567.929684ms | 127.0.0.1 | POST | "/api/generate"
GIN] 2025/05/03 - 06:23:25 | 200 | 596.408816ms | 127.0.0.1 | POST | "/api/generate"
GIN] 2025/05/03 - 06:23:27 | 200 | 288.226115ms | 127.0.0.1 | POST | "/api/generate"
GIN] 2025/05/03 - 06:23:30 | 200 | 590.605119ms | 127.0.0.1 | POST | "/api/generate"
GIN] 2025/05/03 - 06:23:32 | 200 | 324.313906ms | 127.0.0.1 | POST | "/api/generate"
GIN] 2025/05/03 - 06:23:34 | 200 | 269.609161ms | 127.0.0.1 | POST | "/api/generate"
GIN] 2025/05/03 - 06:23:36 | 200 | 269.192273ms | 127.0.0.1 | POST | "/api/generate"
GIN] 2025/05/03 - 06:23:39 | 200 | 502.556623ms | 127.0.0.1 | POST | "/api/generate"
Connected to MongoDB Atlas.
Error generating prompt: POST predict: Post "http://127.0.0.1:46075/completion": EOF (status code: 500)
Job finished at: Sat May 3 06:23:42 UTC 2025
Total runtime: 2h 40m 16s
[jmflagg@tiger ollama]$
```

9.1K documents(First Split)(prompt generation

✓ Job finished at: Sat May 3 23:15:21 UTC 2025

🕒 Total runtime: 6h 56m 7s

✓ Initial training of Split 1:

Number of epochs: 3

Extremely subpar accuracy beneath 1%

✓ Second training of Split 1

Scaled documents up to 170k using data augmentation and initiated 2nd training:

9743 unique URLs → 9743 classes

Eval accuracy at 3 epochs: ~56.5%

 Estimated runtime: ~ 33m

✓ Third Training (Split 1) & Evaluation Metrics (9 epochs with Early Stopping)

Evaluation Results

- Evaluation Accuracy: 63.68%
- Evaluation Loss: 1.41
- Evaluation Runtime: 21.69 seconds
- Training Loss: 3.61
- Training Runtime: 5928.87 seconds (≈1 hour 38 minutes)

✓ First Training (Split 2)

Evaluation Results

→ 6619 unique URLs → 6619 classes

of Original Documents: 10940

Scaled to 345K entries using data augmentation. 50 prompts per URL

✓ Split 2 Training Summary:

→ 6619 unique URLs → 6619 classes

of Original Documents: 10940

Scaled to 345K entries using data augmentation. 50 prompts per URL



Evaluation Metrics:

- Evaluation Loss: 1.309
- Accuracy: 59.78%
- Evaluation Runtime: 36.02 seconds
- Epoch: 6



Training Metrics:

- Total Training Runtime: 4605.64 seconds (\approx 1.28 hours) (3 GPUs)
- Final Training Loss: 2.97
- Epochs Completed: 6 (Early Stopping)

Full Dataset Training



Final Training & Evaluation Metrics (Epoch 10)

→ 26881 unique URLs → 26881 classes

Evaluation Results

- Evaluation Accuracy: 56.09%
- Evaluation Loss: 1.86
- Evaluation Runtime: 55.74 seconds
- Samples per Second: 919.97
- Steps per Second: 14.39

Training Results

- Training Loss: 3.38
- Training Runtime: 9,858.64 seconds (≈2 hours 44 minutes)
- Training Samples per Second: 468.11
- Training Steps per Second: 7.31

****Full Dataset with GPT 4 prompts** (Test 1 on old dataset)**

→ 27340 unique URLs → 27340 classes

****Training Metrics:****

- Loss: 5.7659
- Learning Rate: 8.137800081378001e-09
- Epoch: 10.0
- Train Runtime: 20187.8815 seconds (≈5.6 hours)
- Train Samples per Second: 467.467
- Train Steps per Second: 7.304
- Train Loss: 6.81036013685087

****Evaluation Metrics:****

- Eval Loss: 6.149524211883545
- Eval Accuracy: 0.07630319098209007
- Eval Runtime: 121.0097 seconds (≈2 minutes)
- Eval Samples per Second: 866.526
- Eval Steps per Second: 13.544
- Epoch: 10.0

✓ Model, tokenizer & encoder saved to ./bert_search_model3

Latest Dataset

- Size: 1,105,432 examples
- Unique URLs: 28,814

- **GPUs:** 4

Evaluation Metrics (Epoch 3.0)

- **Eval Loss:** 6.61
- **Eval Accuracy:** 0.06
- **Eval Runtime:** 124.00 seconds
- **Eval Samples per Second:** 891.48
- **Eval Steps per Second:** 13.94

Training Metrics (Epoch 3.0)

- **Train Runtime:** 6078.84 seconds (1 hour 41 minutes)
- **Train Samples per Second:** 490.99
- **Train Steps per Second:** 7.67
- **Train Loss:** 7.49

Latest Dataset 2nd Training

- **Size:** 1,105,432 examples
- **Unique URLs:** 28,814
- **GPUs:** 4

Evaluation Metrics (Epoch 10.0)

- **Eval Loss:** 10.281360626220703
- **Eval Accuracy:** 0.00010855405992184108
- **Eval Runtime:** 122.6603 seconds
- **Eval Samples per Second:** 901.221
- **Eval Steps per Second:** 14.088

Training Metrics (Epoch 10.0)

- **Train Runtime:** 19894.9828 seconds (approximately 5.5 hours)
- **Train Samples per Second:** 500.07
- **Train Steps per Second:** 7.814
- **Train Loss:** 10.257603881945869

}

✓ Done. Model + tokenizer + label encoder saved to ./bert_url_classifier

Is this true Apptainer gives GPU support to virtualized containers, but it can't bypass Ollama's limit of only supporting 1 GPU

Yes, that's true.

Apptainer (formerly Singularity) provides GPU support within its containers—meaning, it can give containerized workloads access to system GPUs (using flags like `--nv` for NVIDIA GPUs). So if you're running a model inside an Apptainer container and the host system has available GPUs, your container can use them.

However, **Ollama itself is the limiting factor** here:

- **Ollama currently only supports a single GPU** per inference run or model instance.
- Even if multiple GPUs are visible to the container (via Apptainer or otherwise), Ollama **won't utilize more than one GPU**.

So:

Apptainer = can pass through multiple GPUs
Ollama = only uses one, regardless

How to install Ollama using Apptainer

```
[jmflagg@itiger ollama]$ which apptainer
```

```
/usr/bin/apptainer
```

```
[jmflagg@itiger ollama]$ apptainer pull docker://ollama/ollama:latest
```

Expected Output:

```
INFO: Converting OCI blobs to SIF format
```

```
INFO: Starting build...
```

```
Copying blob dfe8fac24641 done |
```

```
Copying blob d9802f032d67 done |
```

```
Copying blob 161508c220d5 done |
```

```
Copying blob a5fe86995597 done |
```

```
Copying config a67447f853 done |
```

```
Writing manifest to image destination
```

```
2025/03/26 18:44:05 info unpack layer: sha256:d9802f032d6798e2086607424bfe88cb8ec1d6f116e11cd99592dcaf261e9cd2
```

```
2025/03/26 18:44:05 info unpack layer: sha256:161508c220d57e68d292235939d3d0ca2714e963f165892a306c727286d7872a
```

```
2025/03/26 18:44:06 info unpack layer: sha256:a5fe86995597de09f7324de449c4a17aa948d32d9ebe0ccb966f9fe6170084e8
```

```
2025/03/26 18:44:06 info unpack layer: sha256:dfe8fac246417c5c2b27f0724eba520c823ec156af24da5b2a2e63a823a998e9
```

```
INFO: Creating SIF file...
```

Confirm that Ollama SIF file was created:

```
[jmflagg@itiger ollama]$ ls -lh *.sif
```

```
-rwxr-xr-x 1 jmflagg users 1.7G Mar 26 18:44 ollama_latest.sif
```

Command to shell into your Apptainer with Ollama:

```
apptainer shell ollama_latest.sif
```

```
ollama serve & (runs Ollama as background process)
```

Open new terminal and see which models are installed in the apptainer

```
[jmflagg@itiger ~]$ cd /project/jmflagg/ollama/
```

```
[jmflagg@itiger ollama]$ apptainer shell ollama_latest.sif
```

Pull your desired model:

```
ollama pull llama3.1:8b
```

List all models installed

```
Apptainer> ollama list
```

NAME	ID	SIZE	MODIFIED
llama3.1:8b	a80c4f17acd5	4.9 GB	2 days ago

While trying to download models, there may be a time that you receive a home storage quota error:

You will have to delete unnecessary files from your home directory to create space

Command that I use to check my home directory:

```
du -h --max-depth=1 /home/jmflagg
```

Example: Output

```
11K  /home/jmflagg/bin
```

```
512  /home/jmflagg/Downloads
```

```
1.5K  /home/jmflagg/.x2go
```

```
1.0K  /home/jmflagg/.nv
```

349K /home/jmflagg/.config

1.0K /home/jmflagg/.gnupg

3.3G /home/jmflagg/.apptainer

512 /home/jmflagg/Videos

512 /home/jmflagg/Music

1.9G /home/jmflagg/.ollama

3.2G /home/jmflagg/ollama

3.4G /home/jmflagg/.local

512 /home/jmflagg/Public

8.9G /home/jmflagg/.cache

512 /home/jmflagg/Pictures

1.5K /home/jmflagg/.mozilla

512 /home/jmflagg/Documents

512 /home/jmflagg/Templates

512 /home/jmflagg/Desktop

73K /home/jmflagg/.ssh

21G /home/jmflagg

[jmflagg@itiger ollama]\$ rm -rf ~/.cache (gives me 9GB of space)