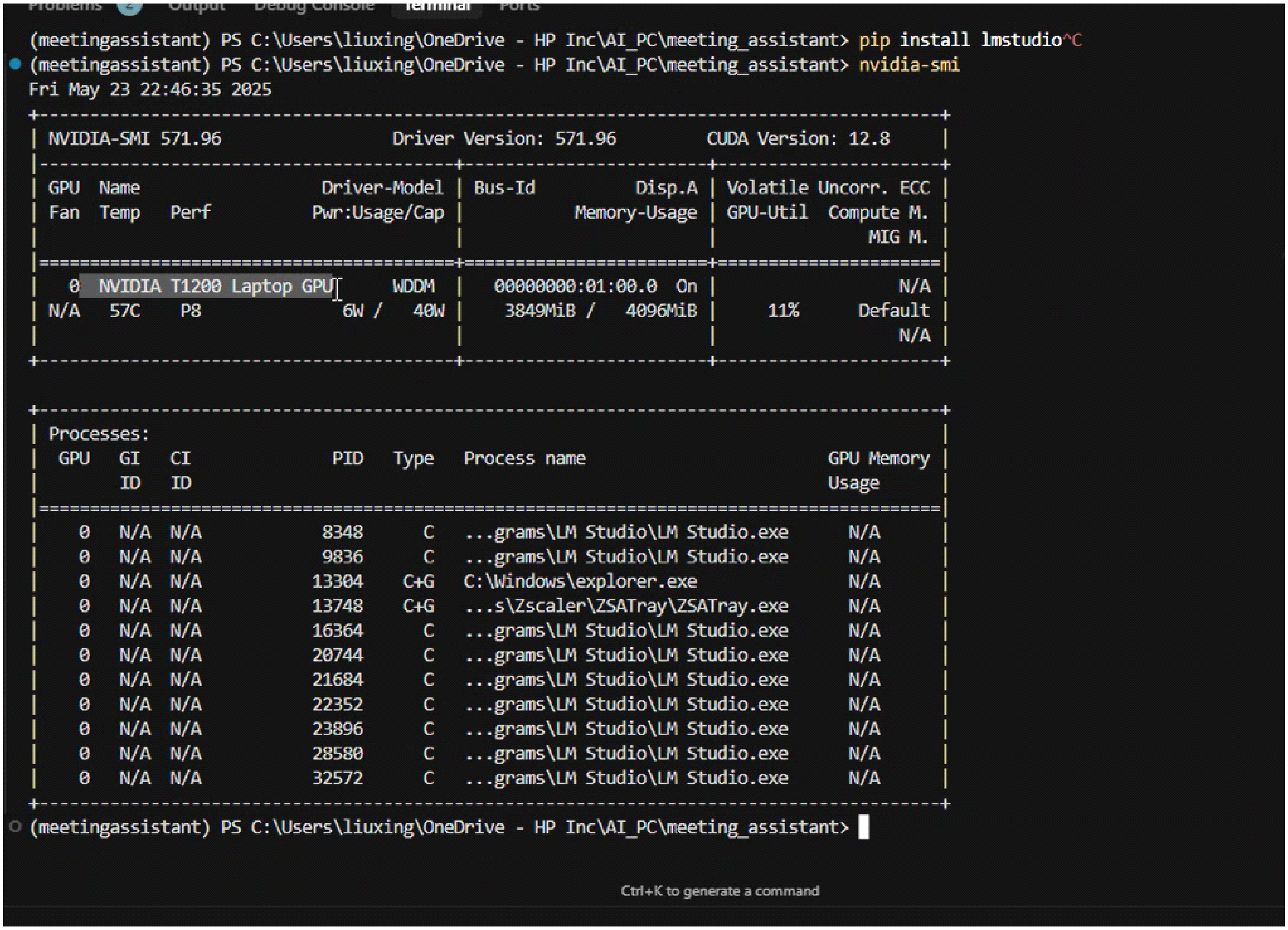
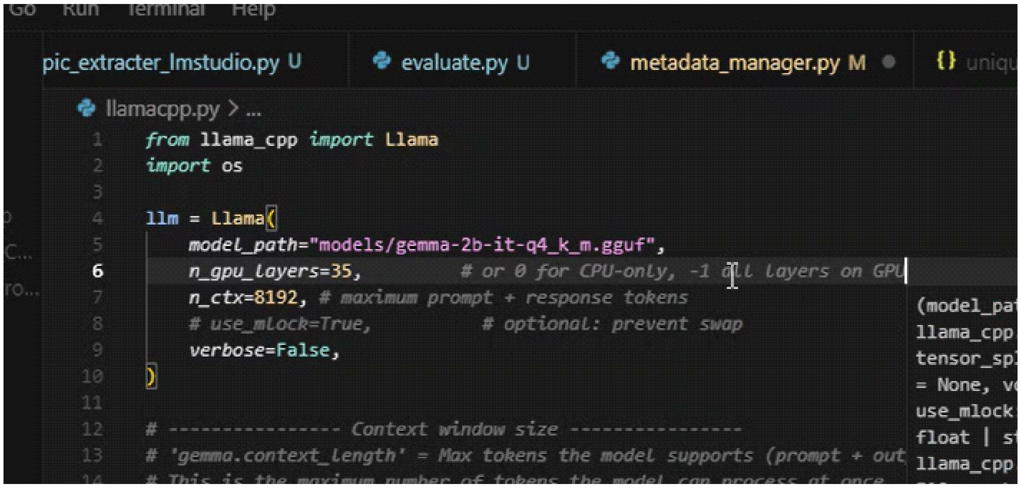
Friday, May 23, 2025   10:47 PM



```
(meetingassistant) PS C:\Users\liuxing\OneDrive - HP Inc\AI_PC\meeting_assistant> pip install lmstudio^C
(meetingassistant) PS C:\Users\liuxing\OneDrive - HP Inc\AI_PC\meeting_assistant> nvidia-smi
Fri May 23 22:46:35 2025
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 571.96                 Driver Version: 571.96         CUDA Version: 12.8     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                  Driver-Model | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  NVIDIA T1200 Laptop GPU      WDDM  |   00000000:01:00.0  On |                  N/A |
| N/A   57C    P8              6W /   40W |    3849MiB /   4096MiB |     11%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI              PID   Type   Process name                        GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|    0   N/A  N/A            8348      C   ...grams\LM Studio\LM Studio.exe      N/A      |
|    0   N/A  N/A            9836      C   ...grams\LM Studio\LM Studio.exe      N/A      |
|    0   N/A  N/A           13304    C+G   C:\Windows\explorer.exe               N/A      |
|    0   N/A  N/A           13748    C+G   ...s\Zscaler\ZSATray\ZSATray.exe      N/A      |
|    0   N/A  N/A           16364      C   ...grams\LM Studio\LM Studio.exe      N/A      |
|    0   N/A  N/A           20744      C   ...grams\LM Studio\LM Studio.exe      N/A      |
|    0   N/A  N/A           21684      C   ...grams\LM Studio\LM Studio.exe      N/A      |
|    0   N/A  N/A           22352      C   ...grams\LM Studio\LM Studio.exe      N/A      |
|    0   N/A  N/A           23896      C   ...grams\LM Studio\LM Studio.exe      N/A      |
|    0   N/A  N/A           28580      C   ...grams\LM Studio\LM Studio.exe      N/A      |
|    0   N/A  N/A           32572      C   ...grams\LM Studio\LM Studio.exe      N/A      |
+-----------------------------------------------------------------------------------------+
(meetingassistant) PS C:\Users\liuxing\OneDrive - HP Inc\AI_PC\meeting_assistant>
```



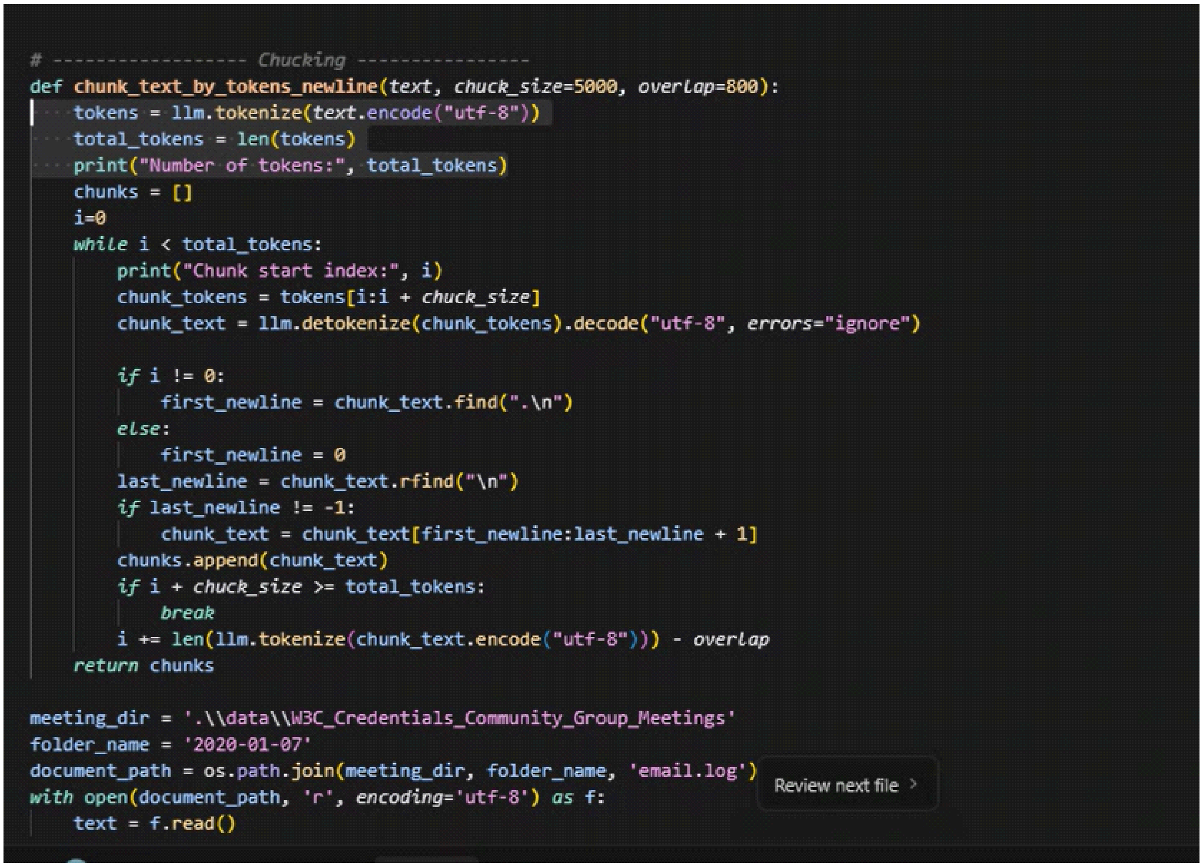
```python
from llama_cpp import Llama
import os

llm = Llama(
    model_path="models/gemma-2b-it-q4_k_m.gguf",
    n_gpu_layers=35,        # or 0 for CPU-only, -1 all layers on GPU
    n_ctx=8192, # maximum prompt + response tokens
    # use_mlock=True,          # optional: prevent swap
    verbose=False,
)

# --------------- Context window size ---------------
# 'gemma.context_length' = Max tokens the model supports (prompt + out
# This is the maximum number of tokens the model can process at once
```

```
(model_pa
llama_cpp
tensor_sp
= None, v
use_mlock
float | s
llama_cpp
```

Word->token->embedding
Context window
Know how many your tokens
Temperature=0, for fact given context. 1, creativity

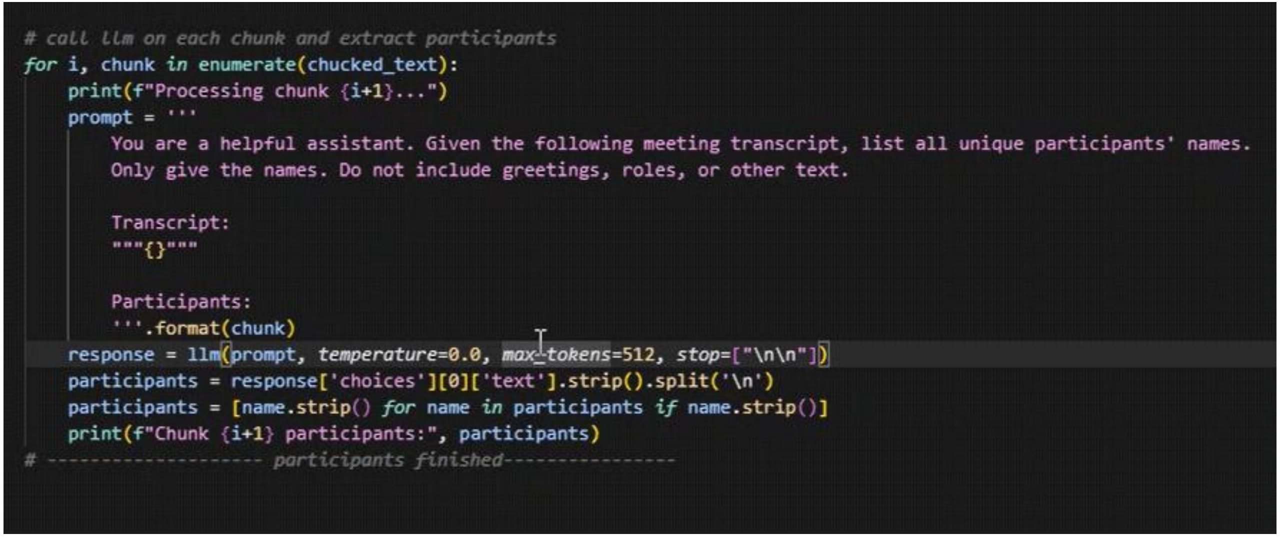


```
# ------------------ Chucking ----------------
def chunk_text_by_tokens_newline(text, chuck_size=5000, overlap=800):
    tokens = llm.tokenize(text.encode("utf-8"))
    total_tokens = len(tokens)
    print("Number of tokens:", total_tokens)
    chunks = []
    i=0
    while i < total_tokens:
        print("Chunk start index:", i)
        chunk_tokens = tokens[i:i + chuck_size]
        chunk_text = llm.detokenize(chunk_tokens).decode("utf-8", errors="ignore")

        if i != 0:
            first_newline = chunk_text.find(".\n")
        else:
            first_newline = 0
        last_newline = chunk_text.rfind("\n")
        if last_newline != -1:
            chunk_text = chunk_text[first_newline:last_newline + 1]
        chunks.append(chunk_text)
        if i + chuck_size >= total_tokens:
            break
        i += len(llm.tokenize(chunk_text.encode("utf-8"))) - overlap
    return chunks

meeting_dir = '.\\data\\W3C_Credentials_Community_Group_Meetings'
folder_name = '2020-01-07'
document_path = os.path.join(meeting_dir, folder_name, 'email.log')
with open(document_path, 'r', encoding='utf-8') as f:
    text = f.read()
```

Inference token limit: response max tokens



```
# call llm on each chunk and extract participants
for i, chunk in enumerate(chucked_text):
    print(f"Processing chunk {i+1}...")
    prompt = '''
        You are a helpful assistant. Given the following meeting transcript, list all unique participants' names.
        Only give the names. Do not include greetings, roles, or other text.

        Transcript:
        """{}"""

        Participants:
        '''.format(chunk)
    response = llm(prompt, temperature=0.0, max_tokens=512, stop=["\n\n"])
    participants = response['choices'][0]['text'].strip().split('\n')
    participants = [name.strip() for name in participants if name.strip()]
    print(f"Chunk {i+1} participants:", participants)
# ------------------- participants finished---------------
```

Ground truth labeled using gpt (api)
Response difference embedding similarity or
Feed responses to gpt to analyze similarity

Frequency, repeatedly listen a sad song (larger weight) vs. multiple sad songs
Search music (larger weight)