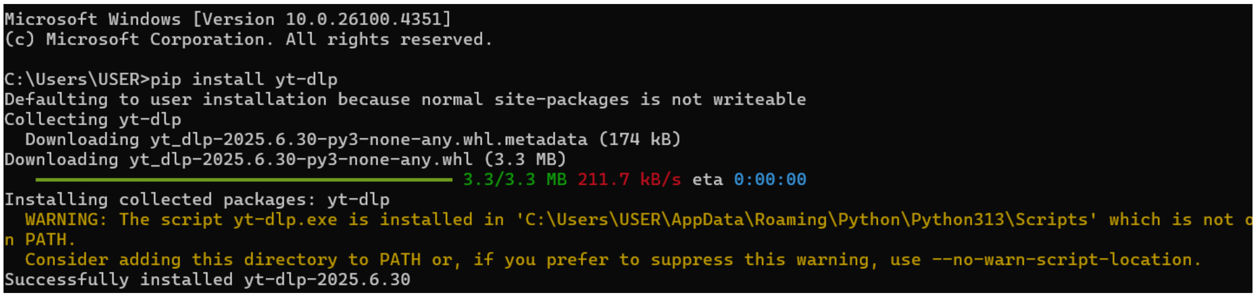
Video link: https://www.youtube.com/watch?v=i_LwzRVP7bg&t=616s

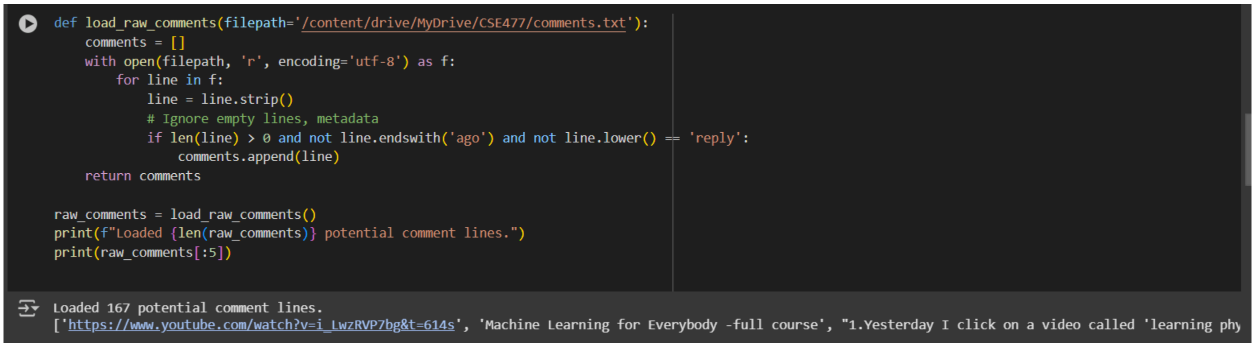
1.Python and yt-dlp installed:



```
Microsoft Windows [Version 10.0.26100.4351]
(c) Microsoft Corporation. All rights reserved.

C:\Users\USER>pip install yt-dlp
Defaulting to user installation because normal site-packages is not writeable
Collecting yt-dlp
  Downloading yt_dlp-2025.6.30-py3-none-any.whl.metadata (174 kB)
Downloading yt_dlp-2025.6.30-py3-none-any.whl (3.3 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.3/3.3 MB 211.7 kB/s eta 0:00:00
Installing collected packages: yt-dlp
  WARNING: The script yt-dlp.exe is installed in 'C:\Users\USER\AppData\Roaming\Python\Python313\Scripts' which is not o
n PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed yt-dlp-2025.6.30
```

2.

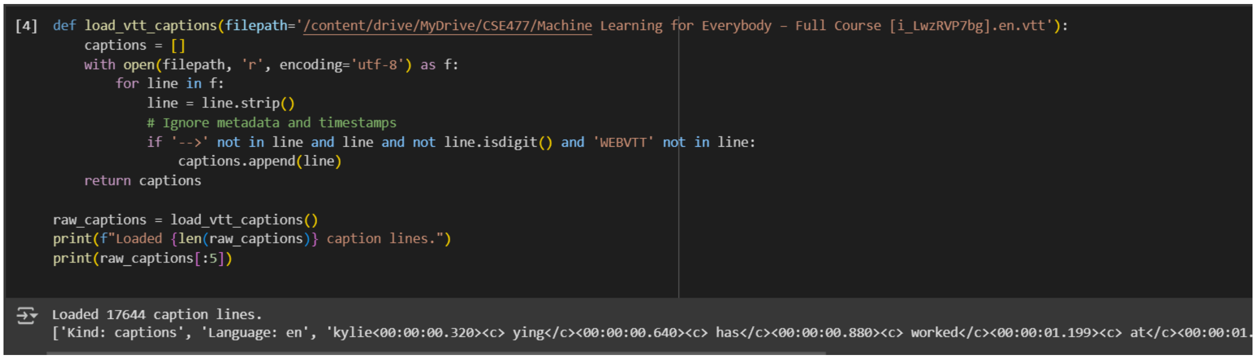


```python
def load_raw_comments(filepath='/content/drive/MyDrive/CSE477/comments.txt'):
    comments = []
    with open(filepath, 'r', encoding='utf-8') as f:
        for line in f:
            line = line.strip()
            # Ignore empty lines, metadata
            if len(line) > 0 and not line.endswith('ago') and not line.lower() == 'reply':
                comments.append(line)
    return comments

raw_comments = load_raw_comments()
print(f"Loaded {len(raw_comments)} potential comment lines.")
print(raw_comments[:5])
```

```
Loaded 167 potential comment lines.
['https://www.youtube.com/watch?v=i_LwzRVP7bg&t=614s', 'Machine Learning for Everybody -full course', "1.Yesterday I click on a video called 'learning phy
```

3.

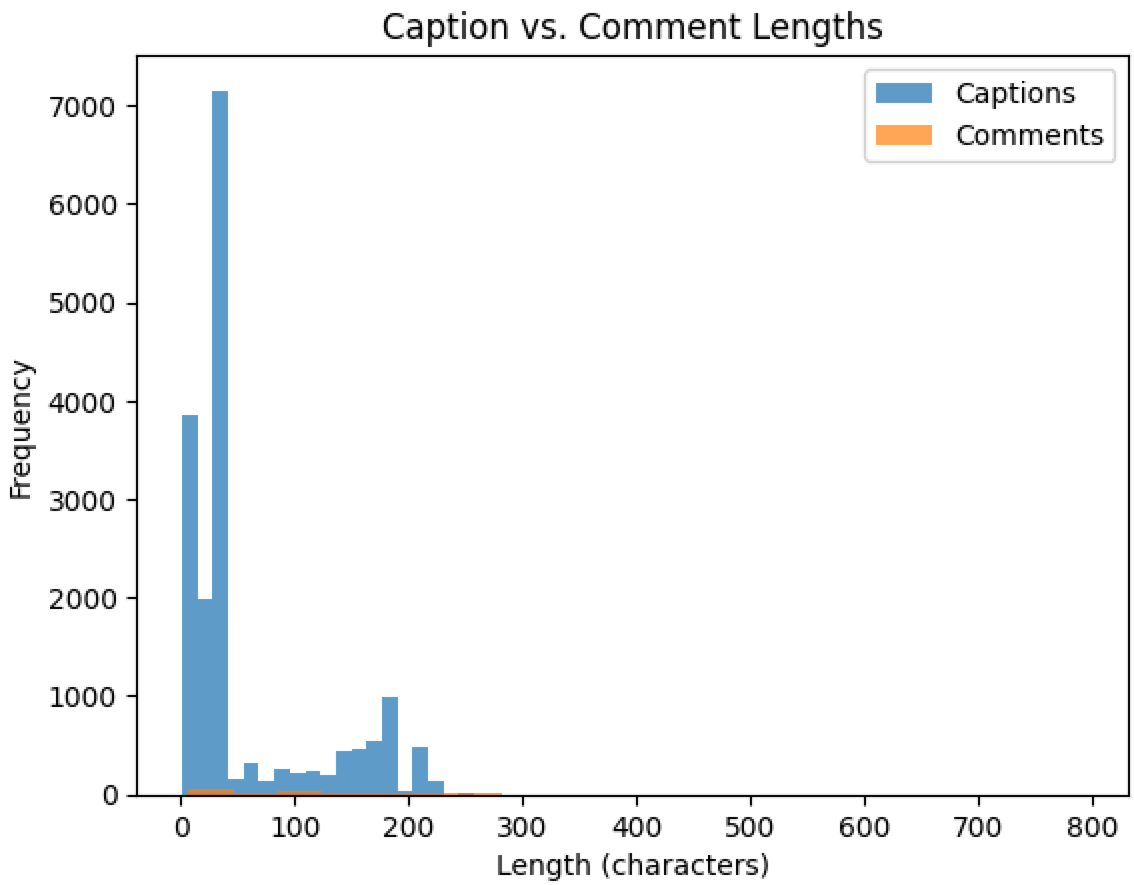


```python
def load_vtt_captions(filepath='/content/drive/MyDrive/CSE477/Machine Learning for Everybody – Full Course [i_LwzRVP7bg].en.vtt'):
    captions = []
    with open(filepath, 'r', encoding='utf-8') as f:
        for line in f:
            line = line.strip()
            # Ignore metadata and timestamps
            if '-->' not in line and line and not line.isdigit() and 'WEBVTT' not in line:
                captions.append(line)
    return captions

raw_captions = load_vtt_captions()
print(f"Loaded {len(raw_captions)} caption lines.")
print(raw_captions[:5])
```

```
Loaded 17644 caption lines.
['Kind: captions', 'Language: en', 'kylie<00:00:00.320><c> ying</c><00:00:00.640><c> has</c><00:00:00.880><c> worked</c><00:00:01.199><c> at</c><00:00:01.
```

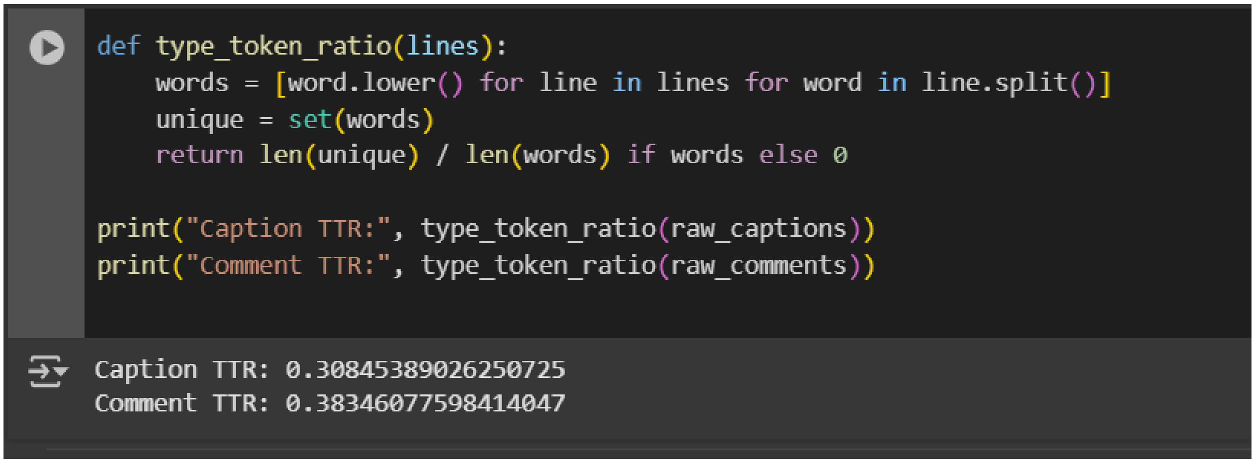
4. Plots and results interpreted



□ Captions tend to be much longer and more structured than comments.

□ Most captions are very short, possibly due to trends like hashtags, emojis, or minimal text.

□ Comments are fewer in number, and rarely exceed 200 characters.



```
def type_token_ratio(lines):
    words = [word.lower() for line in lines for word in line.split()]
    unique = set(words)
    return len(unique) / len(words) if words else 0

print("Caption TTR:", type_token_ratio(raw_captions))
print("Comment TTR:", type_token_ratio(raw_comments))
```

```
Caption TTR: 0.30845389026250725
Comment TTR: 0.38346077598414047
```

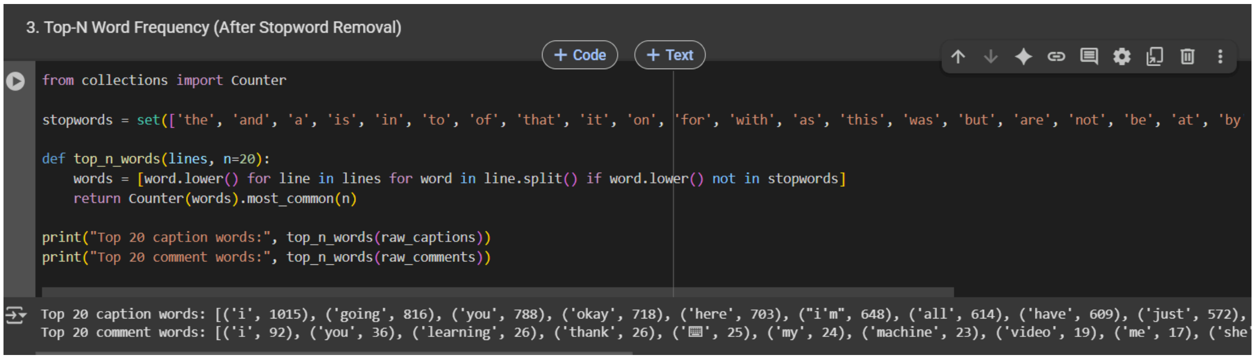
Metric	Value	Interpretation
Caption TTR	0.308	Captions use a moderate but limited variety of words. Many repeated/common terms.
Comment TTR	0.383	Comments use **more unique vocabulary**, showing **higher lexical diversity**.

Comments have a higher TTR (0.383):

- ☐ Viewers are using a wider range of words when writing comments.
- ☐ They express more varied opinions, questions, emotions, and reactions.
- ☐ Indicates richer and more personalized language use.

Captions have a lower TTR (0.308):

- ☐ Captions tend to reuse common terms like “awesome”, “tutorial”, “🔥”, “ML”, etc.
- ☐ Captions are more formulaic or standardized, likely optimized for attention or SEO.



```
from collections import Counter

stopwords = set(['the', 'and', 'a', 'is', 'in', 'to', 'of', 'that', 'it', 'on', 'for', 'with', 'as', 'this', 'was', 'but', 'are', 'not', 'be', 'at', 'by

def top_n_words(lines, n=20):
    words = [word.lower() for line in lines for word in line.split() if word.lower() not in stopwords]
    return Counter(words).most_common(n)

print("Top 20 caption words:", top_n_words(raw_captions))
print("Top 20 comment words:", top_n_words(raw_comments))
```

```
Top 20 caption words: [('i', 1015), ('going', 816), ('you', 788), ('okay', 718), ('here', 703), ("i'm", 648), ('all', 614), ('have', 609), ('just', 572),
Top 20 comment words: [('i', 92), ('you', 36), ('learning', 26), ('thank', 26), ('⌨', 25), ('my', 24), ('machine', 23), ('video', 19), ('me', 17), ('she
```

Caption:
Captions reflect spoken narration, so the language is instructional, casual, and process-oriented.

Words like *“okay”, “here”, “right”, “now”, “just”* are common in live explanations and demonstrations.

Many function words (like *"i", "you", "have", "know"*) dominate, suggesting repetitive but necessary scaffolding in explanation.

Comments:
Comments contain gratitude (*thank*, *thanks*), emotional praise (*best*, *great*, *really*), and personal insight (*i*, *my*, *understand*).

More content-focused terms like *machine*, *learning*, *ml*, *course* suggest engagement with the topic.

Comments are often short personal reflections or feedback.

Interpretation:

Captions are dominated by spoken instructional language, focused on guiding the viewer through the material.

Comments are centered on emotional responses, feedback, and reflection on learning.

The instructor's focus is on teaching; the audience's focus is on appreciation and learning outcome.

Learners feel more confident.

They understand complex topics better (many mention how this tutorial finally helped them "get it").

The tutorial successfully communicates complex ML concepts to a general or semi-technical audie