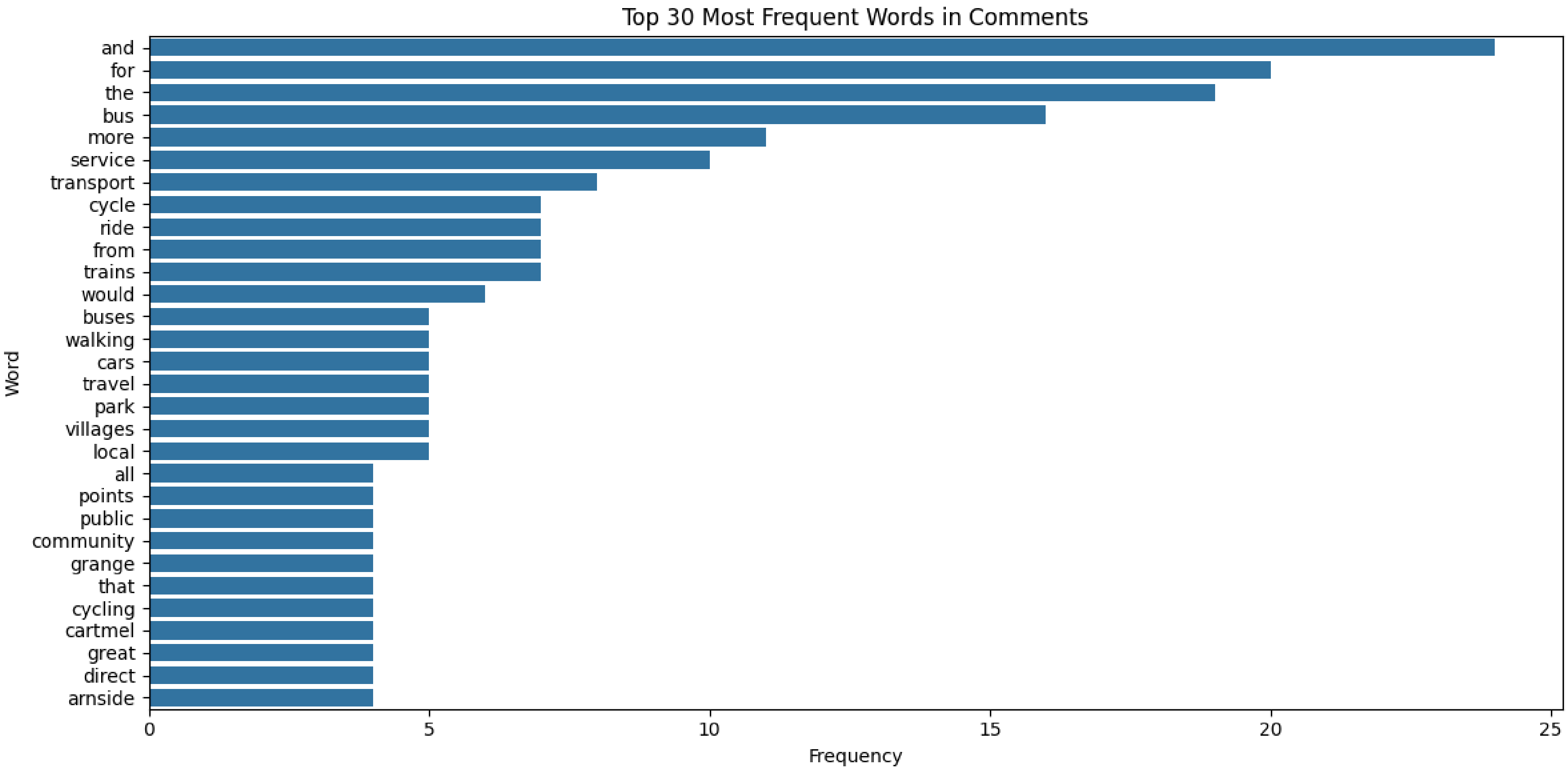
```
import pandas as pd
import matplotlib.pyplot as plt
from collections import Counter
import re
import seaborn as sns


import os
from google.colab import drive
drive.mount('/content/drive')

folder_path = "/content/drive/MyDrive/PEAT Project"
os.makedirs(folder_path, exist_ok=True)

# Load data manually from the text above into a list of comments
# Extract the meaningful comment rows only (excluding headers and blank lines)

data_raw = [
    "Having Uber available would be great….but maybe wishful thinking",
    "Park and Ride",
    "E-bikes/scooters",
    "A closer petrol station",
    "Speed bumps in villages to slow cars down. A lot of new young drivers and holiday makers (Haven) around. More pedestrian crossings in villages to make it safer for the elderly
    "Travel at peak times for working people. Early travel for those to do school runs as have to rely on cars as no transport before half 9. Earlier and later travel as taxis exper
    "Trains and buses so regular and co-ordinated that cars are unnecessary.",
    "Park and Ride facilities.",
    "All public transport run on clean energy. Charging points for bikes and cars in many locations.",
    "Safe, joined up cycle paths.",
    "Signpost distances between access points on the prom. Open up more crossing across the railway track e.g. lido – medical centre. Footpath across to Arnside. Improve pavements f
    "Hawes-Yorkshire Dales has the little white bus for the community, maybe a little bus would be good for the community here.",
    "Cycle paths and walkways that are joined up to avoid walking cycling on roads without a path to allow children to walk /ride safely.",
    "Direct link to Windermere via train, improved bus services to Kendal and Bowness. Much lower cost for the trains £8 for 1 adult and 2 children to Grange from Flookburgh.",
    "More connection out of the peninsula e.g. Saturday service and connection from villages to Grange to catch X6 service. X6 misses trains both ways only by 10mins.",
    "An accessible footpath for mobility scooters or adaptable equipment (3 wheel bike for example).",
    "Ideally a bridge/causeway across the bay to Barrow and Arnside for disabled vehicles etc and bikes (even a small toll may help).",
    "It would be great if there was a night ride based at Grange.",
    "The mobility scooter hire on the station is quite difficult to book.",
    "Footpath to link Allithwaite with Cartmel. Combined travel tickets trains and bus. Public EV charging points at community centre.",
    "More trains on Saturdays, electrify trains and buses. Provide fast changes for visitors.",
    "Perhaps a mini bus that you could ring and book for short local trips for a small fee.",
    "Bus Pass starts at 09:30-23:00. Rail pass 09:00-18:00, don't meet up, Saturday reduced service, Sunday no service.",
    "A Saturday bus service would be great. The village service is fantastic.",
    "Hourly Service to Cartmel via Kents Bank and return. Transport availbe for events.",
    "Working with Cartmel Secondary, explore creating paths (walkers and cyclists). Budget cost to W&F to provide coaches for secondary pupils.",
    "Free parking for 1 hour. Consider Park & Ride buses that operate during season.",
    "More taxi services. Stop car parking on pavements. 20mph throughout the town. Cycle lanes on main routes.",
    "All local schools co-ordinate meeting points for walking buses. Mini bus circuits around peninsula.",
    "Renovate the walking / footpath signposts. Increase local bus service, weekends/evenings.",
    "I work in town and we get a lot of queries about public transport, older people find it hard to access info.",
    "Better intergration of all transport/travel services. Better information of all bus routes locally. Car share scheme."
    "Better intergration of all transport/travel services. Better information of all bus routes locally. Car share scheme.",
    "Extend 530 to villages. Be able to get back from Manchester.",
    "Country lanes around hamlets limited speed, 60mph too fast.",
    "Desperately need local transport to outlaying hamlets and more footpaths along the roads.",
    "Run buses for the villages e.g Newby Bridge to Cartmel picking up e.g Barber Green.",
    "Make transport free! then cars would be left at home.",
    "Bring back 20p parking time. Double yellow lines",
    "I think there should be electric planes",
    "Community funded bus?",
    "Reduce speed limits on country roads - its unsafe to ride my electric bike.",
    "More taxis. Park and ride. Tourist train from main st car park.",
    "Pedestrian crossing Kents Bank road, between library and close street.",
    "More regular bus route",
    "More cycling connectivity. Arnside via direct walking/cycle path",
    "Increase bus service(affordable). Cycle path over the viaduct.",
    "Direct bus service to/from hospitals. Tourist Info signs",
    "More obvious bus stops. Over 60's pass on trains. Fill potholes.",
    "Reinstate the direct bus route from Grange to Furness General Hospital.",
    "Expanding cycling safety would be great. A walking/cycling route over the viaduct to Arnside.",
    "Cycle tracks so people could cycle safely with rentable 'Borsi bikes'.",
    "Public transport more available, later times. Reduce car use.",
]

# Preprocess text
words = []
for comment in data_raw:
    comment = comment.lower()
    tokens = re.findall(r'\b[a-z]{3,}\b', comment)  # words of at least 3 letters
    words.extend(tokens)

# Count frequency
word_freq = Counter(words)
df_freq = pd.DataFrame(word_freq.items(), columns=["Word", "Frequency"])
df_freq = df_freq.sort_values(by="Frequency", ascending=False).head(30)
df_freq = df_freq.reset_index(drop=True)

from IPython.display import display
display(df_freq.head(30))

df_freq.to_csv(f"{folder_path}/word_freq.csv", index=False)

# Plot
plt.figure(figsize=(12,6))
sns.barplot(x="Frequency", y="Word", data=df_freq)
plt.title("Top 30 Most Frequent Words in Comments")
plt.xlabel("Frequency")
plt.ylabel("Word")
plt.tight_layout()
plt.show()
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

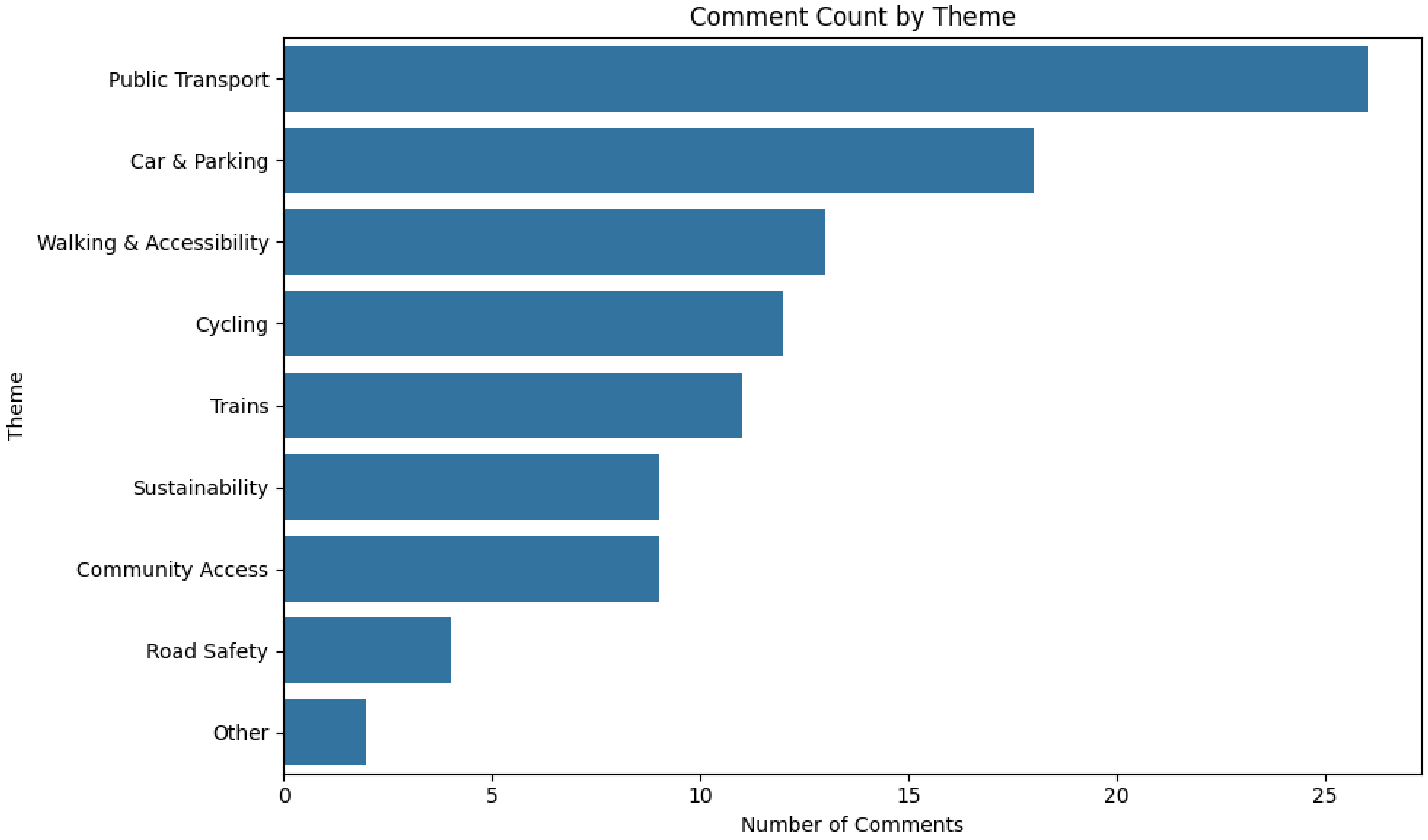
	Word	Frequency
0	and	24
1	for	20
2	the	19
3	bus	16
4	more	11
5	service	10
6	transport	8
7	cycle	7
8	ride	7
9	from	7
10	trains	7
11	would	6
12	buses	5
13	walking	5
14	cars	5
15	travel	5
16	park	5
17	villages	5
18	local	5
19	all	4
20	points	4
21	public	4
22	community	4
23	grange	4
24	that	4
25	cycling	4
26	cartmel	4
27	great	4
28	direct	4

```
29    arnside    4
```


Top 30 Most Frequent Words in Comments
Word
Frequency
and
for
the
bus
more
service
transport
cycle
ride
from
trains
would
buses
walking
cars
travel
park
villages
local
all
points
public
community
grange
that
cycling
cartmel
great
direct
arnside
0
5
10
15
20
25

```
import numpy as np
import os
from google.colab import drive
drive.mount('/content/drive')

folder_path = "/content/drive/MyDrive/PEAT Project"
os.makedirs(folder_path, exist_ok=True)


# Add categories/themes based on keywords
def categorize(comment):
    comment_lower = comment.lower()
    categories = []
    if any(word in comment_lower for word in ["bus", "buses", "route", "mini bus", "530", "public transport"]):
        categories.append("Public Transport")
    if any(word in comment_lower for word in ["train", "rail", "station", "platform"]):
        categories.append("Trains")
    if any(word in comment_lower for word in ["cycle", "bike", "cycling", "bikes", "e-bike"]):
        categories.append("Cycling")
    if any(word in comment_lower for word in ["walk", "footpath", "pedestrian", "pavement"]):
        categories.append("Walking & Accessibility")
    if any(word in comment_lower for word in ["car", "parking", "park", "vehicles", "lorry"]):
        categories.append("Car & Parking")
    if any(word in comment_lower for word in ["electric", "ev", "mobility scooter", "charging"]):
        categories.append("Sustainability")
    if any(word in comment_lower for word in ["speed", "limit", "traffic calming", "20mph", "bumps"]):
        categories.append("Road Safety")
    if any(word in comment_lower for word in ["community", "schools", "elderly", "disabled", "access"]):
        categories.append("Community Access")
    return categories if categories else ["Other"]

# Create structured DataFrame
df_comments = pd.DataFrame(data_raw, columns=["Comment"])
df_comments["Themes"] = df_comments["Comment"].apply(categorize)

# Explode multi-label rows for easier analysis
df_exploded = df_comments.explode("Themes")

# Count themes
theme_counts = df_exploded["Themes"].value_counts().reset_index()
theme_counts.columns = ["Theme", "Count"]

# Save the datasets
df_comments.to_csv(f"{folder_path}/transport_feedback_comments_with_themes.csv", index=False)
df_freq.to_csv(f"{folder_path}/top_word_frequencies.csv", index=False)
theme_counts.to_csv(f"{folder_path}/theme_counts.csv", index=False)
# Plot Theme Counts
plt.figure(figsize=(10,6))
sns.barplot(data=theme_counts, y="Theme", x="Count")
plt.title("Comment Count by Theme")
plt.xlabel("Number of Comments")
plt.ylabel("Theme")
plt.tight_layout()
plt.show()
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```


Comment Count by Theme
Public Transport
Car & Parking
Walking & Accessibility
Cycling
Trains
Sustainability
Community Access
Road Safety
Other
Theme
Number of Comments