EXP 7

September 25, 2025

[4]:
```
from bs4 import BeautifulSoup
import matplotlib.pyplot as plt
from collections import Counter
import re
import math
import html
import os

# ---------- Input: saved HTML file(s) ---------- FILES =
[r"C:\BIDA LAB\EXP 7\r_news.htm"]

# ---------- Lexicon ----------
LEX = {
    "good": 2,
    "great": 3,
    "excellent": 4,
    "amazing": 4,
    "love": 3,
    "like": 2,
    "nice": 2,
    "awesome": 4,
    "helpful": 2,
    "bad": -2,
    "terrible": -3,
    "awful": -3,
    "hate": -3,
    "slow": -2,
    "buggy": -3,
    "confusing": -2,
    "broken": -3,
    "issue": -2,
    "problem": -2,
    "worst": -4,
    "disappointed": -3,
    "frustrating": -3
}
```



```
NEG = {"not", "no", "never", "none", "hardly", "barely", "scarcely"}
```

```
    # ---------- Regex helpers ----------
    WORD_RE = re.compile(r"[A-Za-z][A-Za-z\-']+")
    URL_RE = re.compile(r"https?://\S+")
    EMOJI_RE = re.compile(r"[\U00010000-\U0010ffff]")

[5]: # ---------- Cleaning ----------
    def clean(text):
        text = html.unescape(text or "")
        text = URL_RE.sub(" ", text)
        text = EMOJI_RE.sub("", text)
        text = re.sub(r"\s+", " ", text).strip()
        return text

    def tokenize(text):
        return [w.lower() for w in WORD_RE.findall(text)]

    # ---------- Sentiment scoring ----------
    def score(text):
        words = tokenize(text)
        total = 0.0
        for i, w in enumerate(words):
            val = LEX.get(w, 0)
            if val:
                                    if any(words[i-j] in NEG for j in range(1, min(3, i) + 1)):
                    val *= -1
            total += val
        return total / max(1.0, math.log(len(words) + 1, 3))

[6]: # ---------- Extract comments ----------
    def grab_comments(html_doc):
        soup = BeautifulSoup(html_doc, "html.parser")
        for t in soup(["script", "style", "noscript", "iframe", "svg"]): t.decompose()
        cands = soup.find_all(attrs={"class": re.compile("comment|reply", re.I)}) cands +=
        soup.find_all(id=re.compile("comment|reply", re.I))
        texts = []
        for el in set(cands):
            txt = clean(el.get_text(" ", strip=True))
            if len(txt) > 12:
                texts.append(txt)
        return list(dict.fromkeys(texts))

[ ]: # ---------- Pipeline ----------
    comments = []
    for path in FILES:
```

```
        if os.path.exists(path):
                with open(path, encoding="utf-8", errors="ignore") as f:
                comments += grab_comments(f.read())

    comments = list(dict.fromkeys(comments))

    if not comments:
        raise SystemExit("No comments found. Save a page with comments and set␣
    ↪FILES.")

    scores = [score(c) for c in comments]

    labels = []
    for s in scores:
        if s >= 0.05:
            labels.append("positive")
        elif s <= -0.05:
            labels.append("negative")
        else:
            labels.append("neutral")

    cnt = Counter(labels)
    overall = max(cnt, key=cnt.get)

    print("Overall tone:", overall)
    print("Counts:", dict(cnt))
```

[ ]:
```
# ---------- Plot ----------
    order = ["positive", "neutral", "negative"]
    vals = [cnt.get(k, 0) for k in order]

    plt.bar(order, vals)
    plt.title("Sentiment Distribution")
    plt.ylabel("Count")
    plt.tight_layout()
    plt.show()
```