In [13]:
```python
import sys
import subprocess

# 定義所需的套件和版本
packages = [
    "jupyter",
    "scikit-learn",
    "pandas",
    "numpy",
    "matplotlib==3.7.3",
    "plotly",
    "seaborn",
    "nltk",
    "PAMI",
    "umap-learn"
]

# 安裝套件的函數
def install_package(package):
    subprocess.check_call([sys.executable, "-m", "pip", "install", package])

# 逐一安裝套件
for package in packages:
    try:
        install_package(package)
        print(f"Successfully installed {package}")
    except Exception as e:
        print(f"Failed to install {package}: {e}")
```

```
Successfully installed jupyter
Successfully installed scikit-learn
Successfully installed pandas
Successfully installed numpy
Successfully installed matplotlib==3.7.3
Successfully installed plotly
Successfully installed seaborn
Successfully installed nltk
Successfully installed PAMI
Successfully installed umap-learn
```

In [16]:
```python
# TEST necessary for when working with external scripts
%load_ext autoreload
%autoreload 2
```

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

In [22]:
```python
import tarfile

# 設定壓縮檔的路徑和解壓目標資料夾
file_path = r"C:\Users\jon29\Downloads\20news-bydate.tar.gz"
extract_path = r"C:\Users\jon29\Downloads\20news-bydate"

# 解壓縮檔案
with tarfile.open(file_path, 'r:gz') as tar:
    tar.extractall(path=extract_path)

print("文件解壓完成！")
```

文件解壓完成！

In [23]:
```
import os
import pandas as pd

# 解壓後的資料夾路徑
data_dir = r"C:\Users\jon29\Downloads\20news-bydate\20news-bydate-train"

# 用於存儲文本和分類的列表
texts = []
labels = []

# 遍歷每個分類目錄
for category in os.listdir(data_dir):
    category_path = os.path.join(data_dir, category)
    if os.path.isdir(category_path):  # 確保是目錄
        for file_name in os.listdir(category_path):
            file_path = os.path.join(category_path, file_name)
            if os.path.isfile(file_path):  # 確保是文件
                with open(file_path, 'r', encoding='latin1') as file:
                    texts.append(file.read())
                    labels.append(category)

# 將文本和分類加入 DataFrame
df = pd.DataFrame({'text': texts, 'category': labels})
print(df.head())
print(f"共讀取了 {len(df)} 條記錄")
```

```
                                                text     category
0  From: mathew <mathew@mantis.co.uk>\nSubject: A...  alt.atheism
1  From: mathew <mathew@mantis.co.uk>\nSubject: A...  alt.atheism
2  From: I3150101@dbstu1.rz.tu-bs.de (Benedikt Ro...  alt.atheism
3  From: mathew <mathew@mantis.co.uk>\nSubject: R...  alt.atheism
4  From: strom@Watson.Ibm.Com (Rob Strom)\nSubjec...  alt.atheism
共讀取了 11314 條記錄
```

In [26]:
```
# categories
categories = ['alt.atheism', 'soc.religion.christian', 'comp.graphics', 'sci.me
```

In [29]:
```
from sklearn.datasets import load_files

# 載入本地數據集，指向解壓後的主目錄
data_dir = 'C:\\Users\\jon29\\Downloads\\20news-bydate\\20news-bydate-train'
data = load_files(data_dir, categories=categories, encoding='latin1')

# 構建 DataFrame
df = pd.DataFrame({
    'text': data.data,
    'category': [data.target_names[target] for target in data.target]
})

print(df.head())
```

```
                                                text                category
0  From: dpc47852@uxa.cso.uiuc.edu (Daniel Paul C...                 sci.med
1  From: yoo@engr.ucf.edu (Hoi Yoo)\nSubject: loo...            comp.graphics
2  From: fernandeza@merrimack.edu\nSubject: Re: T...  soc.religion.christian
3  From: mcelwre@cnsvax.uwec.edu\nSubject: NATURA...                 sci.med
4  From: mathew <mathew@mantis.co.uk>\nSubject: R...             alt.atheism
```

In [32]:
```
from sklearn.datasets import load_files

# 指定數據目錄和類別
data_dir = "C:\\Users\\jon29\\Downloads\\20news-bydate\\20news-bydate-train"
categories = ['alt.atheism', 'soc.religion.christian', 'comp.graphics', 'sci.med

# 使用 load_files 函數載入數據
data = load_files(data_dir, categories=categories, encoding='latin1')

# 構建類似於 fetch_20newsgroups 的數據結構
twenty_train = {
    'data': data.data,  # 文本數據
    'target': data.target,  # 類別索引
    'target_names': data.target_names,  # 類別名稱
    'filenames': data.filenames  # 每篇文章的文件名
}
```

In [33]:
```
import pandas as pd

# 將數據構建成 DataFrame
df = pd.DataFrame({
    'text': twenty_train['data'],
    'category': [twenty_train['target_names'][i] for i in twenty_train['target']
})

# 查看前幾行數據
print(df.head())
```

```
                                                text                category
0  From: dpc47852@uxa.cso.uiuc.edu (Daniel Paul C...                 sci.med
1  From: yoo@engr.ucf.edu (Hoi Yoo)\nSubject: loo...           comp.graphics
2  From: fernandeza@merrimack.edu\nSubject: Re: T...  soc.religion.christian
3  From: mcelwre@cnsvax.uwec.edu\nSubject: NATURA...                 sci.med
4  From: mathew <mathew@mantis.co.uk>\nSubject: R...             alt.atheism
```

In [35]:
```
import pandas as pd

# 將數據構建成 DataFrame
df = pd.DataFrame({
    'text': twenty_train['data'],
    'category': [twenty_train['target_names'][i] for i in twenty_train['target']
})

# 按照 'category' 欄位排序
df = df.sort_values(by='category').reset_index(drop=True)

# 查看前幾行數據
print(df.head())
```

```
                                                text     category
0  From:  (Rashid)\nSubject: Re: Yet more Rushdie...  alt.atheism
1  From: I3150101@dbstu1.rz.tu-bs.de (Benedikt Ro...  alt.atheism
2  From: keith@cco.caltech.edu (Keith Allan Schne...  alt.atheism
3  From: mas@Cadence.COM (Masud Khan)\nSubject: R...  alt.atheism
4  From: ingles@engin.umich.edu (Ray Ingles)\nSub...  alt.atheism
```

In [ ]:

In [ ]:
```
#exercise 1
```

In [36]:

```
# 確認數據集中前幾個樣本的文本數據是否已經正確載入
print("第一個樣本的文本數據:\n", df['text'].iloc[0])
print("\n第二個樣本的文本數據:\n", df['text'].iloc[1])
print("\n第三個樣本的文本數據:\n", df['text'].iloc[2])
```

第一個樣本的文本數據:

```
 From:  (Rashid)
Subject: Re: Yet more Rushdie [Re: ISLAMIC LAW]
Nntp-Posting-Host: 47.252.4.179
Organization: NH
Lines: 76

In article <1993Apr14.131032.15644@monu6.cc.monash.edu.au>,
darice@yoyo.cc.monash.edu.au (Fred Rice) wrote:
>
> It is my understanding that it is generally agreed upon by the ulema
> [Islamic scholars] that Islamic law applies only in an Islamic country,
> of which the UK is not.  Furthermore, to take the law into one's own
> hands is a criminal act, as these are matters for the state, not for
> individuals.  Nevertheless, Khomeini offered a cash prize for people to
> take the law into their own hands -- something which, to my
> understanding, is against Islamic law.

Yes, this is also my understanding of the majority of Islamic laws.
However, I believe there are also certain legal rulings which, in all
five schools of law (4 sunni and 1 jaffari), can be levelled against
muslim or non-muslims, both within and outside dar-al-islam. I do
not know if apostasy (when accompanied by active, persistent, and
open hostility to Islam) falls into this category of the law. I do know
that
historically, apostasy has very rarely been punished at all, let alone
by the death penalty.

My understanding is that Khomeini's ruling was not based on the
law of apostasy (alone). It was well known that Rushdie was an apostate
long before he wrote the offending novel and certainly there is no
precedent in the Qur'an, hadith, or in Islamic history for indiscriminantly
levelling death penalties for apostasy.

I believe the charge levelled against Rushdie was that of "fasad". This
ruling applies both within and outside the domain of an
Islamic state and it can be carried out by individuals. The reward was
not offered by Khomeini but by individuals within Iran.


> Stuff deleted
> Also, I think you are muddying the issue as you seem to assume that
> Khomeini's fatwa was issued due to the _distribution_ of the book.  My
> understanding is that Khomeini's fatwa was issued in response to the
> _writing_ and _publishing_ of the book.  If my view is correct, then
> your viewpoint that Rushdie was sentenced for a "crime in progress" is
> incorrect.
>
I would concur that the thrust of the fatwa (from what I remember) was
levelled at the author and all those who assisted in the publication
of the book. However, the charge of "fasad" can encompass a
number of lesser charges. I remember that when diplomatic relations
broke off between Britain and Iran over the fatwa - Iran stressed that
the condemnation of the author, and the removal of the book from
circulation were two preliminary conditions for resolving the
"crisis". But you are correct to point out that banning the book was not
the main thrust behind the fatwa. Islamic charges such as fasad are
levelled at people, not books.

The Rushdie situation was followed in Iran for several months before the
```

issuance of the fatwa. Rushdie went on a media blitz,
presenting himself as a lone knight guarding the sacred values of
secular democracy and mocking the foolish concerns of people
crazy enough to actually hold their religious beliefs as sacred.
Fanning the flames and milking the controversy to boost
his image and push the book, he was everywhere in the media. Then
Muslim demonstrators in several countries were killed while
protesting against the book. Rushdie appeared momentarily
concerned, then climbed back on his media horse to once again
attack the Muslims and defend his sacred rights. It was at this
point that the fatwa on "fasad" was issued.

The fatwa was levelled at the person of Rushdie - any actions of
Rushdie that feed the situation contribute to the legitimization of
the ruling. The book remains in circulation not by some independant
will of its own but by the will of the author and the publishers. The fatwa
against the person of Rushdie encompasses his actions as well. The
crime was certainly a crime in progress (at many levels) and was being
played out (and played up) in the the full view of the media.

P.S. I'm not sure about this but I think the charge of "shatim" also
applies to Rushdie and may be encompassed under the umbrella
of the "fasad" ruling.

第二個樣本的文本數據:
From: I3150101@dbstu1.rz.tu-bs.de (Benedikt Rosenau)
Subject: Re: islamic genocide
Organization: Technical University Braunschweig, Germany
Lines: 23

In article <1qi83b$ec4@horus.ap.mchp.sni.de>
frank@D012S658.uucp (Frank O'Dwyer) writes:

(Deletion)
>#>Few people can imagine dying for capitalism, a few
>#>more can imagine dying for democracy, but a lot more will die for their
>#>Lord and Savior Jesus Christ who Died on the Cross for their Sins.
>#>Motivation, pure and simple.
>
>Got any cites for this nonsense? How many people will die for Mom?
>Patriotism? Freedom? Money? Their Kids? Fast cars and swimming pools?
>A night with Kim Basinger or Mel Gibson? And which of these things are evil?
>

Read a history book, Fred. And tell me why so many religions command to
commit genocide when it has got nothing to do with religion. Or why so many
religions say that not living up to the standards of the religion is worse
than dieing? Coincidence, I assume. Or ist part of the absolute morality
you describe so often?

Theism is strongly correlated with irrational belief in absolutes. Irrational
belief in absolutes is strongly correlated with fanatism.
   Benedikt

第三個樣本的文本數據:
From: keith@cco.caltech.edu (Keith Allan Schneider)
Subject: Re: "Cruel" (was Re: <Political Atheists?)
Organization: California Institute of Technology, Pasadena

```
Lines: 18
NNTP-Posting-Host: punisher.caltech.edu

livesey@solntze.wpd.sgi.com (Jon Livesey) writes:

>>They spent quite a bit of time on the wording of the Constitution.
>I realise that this is widely held belief in America, but in fact
>the clause on cruel and unusual punishments, like a lot of the
>rest, was lifted from the English Bill of Rights of 1689.

Just because the wording is elsewhere does not mean they didn't spend
much time on the wording.

>>We have already looked in the dictionary to define the word.  Isn't
>>this sufficient?
>Since the dictionary said that a lack of mercy or an intent to
>inflict injury or grief counted as "cruel", sure.

People can be described as cruel in this way, but punishments cannot.

keith
```

In [ ]:
```python
#exercise 2
```

In [38]:
```python
# 篩選出包含 "science" 的文本
df_science = df[df['text'].str.contains("science", case=False, na=False)]
print("包含 'science' 的記錄數量:", len(df_science))
print(df_science.head())

# 計算每個類別的文章數量
category_counts = df['category'].value_counts()
print("每個類別的文章數量:\n", category_counts)
```

```
包含 'science' 的記錄數量: 349
                                                 text     category
5   From: mayne@pipe.cs.fsu.edu (William Mayne)\nS...  alt.atheism
8   From: mangoe@cs.umd.edu (Charley Wingate)\nSub...  alt.atheism
17  Subject: Re: Is Morality Constant (was Re: Bib...  alt.atheism
29  From: jbrown@batman.bmd.trw.com\nSubject: Re: ...  alt.atheism
36  From: pww@spacsun.rice.edu (Peter Walker)\nSub...  alt.atheism
每個類別的文章數量:
 category
soc.religion.christian    599
sci.med                   594
comp.graphics             584
alt.atheism               480
Name: count, dtype: int64
```

In [ ]:
```python
#exercise 3
```

In [39]:
```python
# 篩選出 'sci.med' 類別的記錄
df_sci_med = df[df['category'] == 'sci.med']

# 每 10 筆選取一筆，並顯示前 5 筆
df_sci_med_every_10th = df_sci_med.iloc[::10].head(5)
print("sci.med 類別中每 10 筆選取一筆的前 5 條記錄:\n", df_sci_med_every_10th)
```

```
sci.med 類別中每 10 筆選取一筆的前 5 條記錄:
                                                   text category
1064  From: annick@cortex.physiol.su.oz.au (Annick A...  sci.med
1074  From: jnielsen@magnus.acs.ohio-state.edu (John...  sci.med
1084  From: bruce@Data-IO.COM (Bruce Reynolds)\nSubj...  sci.med
1094  From: geb@cs.pitt.edu (Gordon Banks)\nSubject:...  sci.med
1104  From: rjf@lzsc.lincroftnj.ncr.com (51351[efw]-...  sci.med
```

In [ ]:
```
#exercise 4
```

In [40]:
```
import pandas as pd

# 生成缺失值計數
missing_values_per_row = df.isnull().sum(axis=1)

# 將缺失值數量顯示在 DataFrame 中
df['missing_values_count'] = missing_values_per_row

# 查看前幾行記錄以及每行的缺失值數量
print(df[['text', 'category', 'missing_values_count']].head(10))
```

```
                                                text     category  \
0  From:  (Rashid)\nSubject: Re: Yet more Rushdie...  alt.atheism
1  From: I3150101@dbstu1.rz.tu-bs.de (Benedikt Ro...  alt.atheism
2  From: keith@cco.caltech.edu (Keith Allan Schne...  alt.atheism
3  From: mas@Cadence.COM (Masud Khan)\nSubject: R...  alt.atheism
4  From: ingles@engin.umich.edu (Ray Ingles)\nSub...  alt.atheism
5  From: mayne@pipe.cs.fsu.edu (William Mayne)\nS...  alt.atheism
6  From: keith@cco.caltech.edu (Keith Allan Schne...  alt.atheism
7  From: lmh@juliet.caltech.edu (Henling, Lawrenc...  alt.atheism
8  From: mangoe@cs.umd.edu (Charley Wingate)\nSub...  alt.atheism
9  Organization: Penn State University\nFrom: <MV...  alt.atheism

   missing_values_count
0                     0
1                     0
2                     0
3                     0
4                     0
5                     0
6                     0
7                     0
8                     0
9                     0
```

In [ ]:
```
#exercise 5
```

In [ ]:
```
#為什麼 .isnull() 沒有檢測到某些缺失值？.isnull() 主要檢查數據中的 NaN（非數值）和 No
```

In [41]:
```
import numpy as np
import pandas as pd

# 生成一個自定義函數來檢查是否為特殊的缺失值
def is_missing(x):
    # 檢查是否為 NaN 或 None
    if pd.isnull(x):
        return True
    # 檢查是否為空字符串或僅包含空白的字符串
    elif isinstance(x, str) and (x.strip() == "" or x.strip().lower() in ["na",
        return True
```

```
    else:
        return False

# 應用該自定義函數來檢查缺失值
missing_values = df.applymap(is_missing)

# 計算每一行的缺失值數量（包括非典型缺失值）
df['total_missing_count'] = missing_values.sum(axis=1)

# 查看前幾行數據
print(df[['text', 'category', 'total_missing_count']].head(10))
```

```
                                                text     category  \
0  From:  (Rashid)\nSubject: Re: Yet more Rushdie...  alt.atheism
1  From: I3150101@dbstu1.rz.tu-bs.de (Benedikt Ro...  alt.atheism
2  From: keith@cco.caltech.edu (Keith Allan Schne...  alt.atheism
3  From: mas@Cadence.COM (Masud Khan)\nSubject: R...  alt.atheism
4  From: ingles@engin.umich.edu (Ray Ingles)\nSub...  alt.atheism
5  From: mayne@pipe.cs.fsu.edu (William Mayne)\nS...  alt.atheism
6  From: keith@cco.caltech.edu (Keith Allan Schne...  alt.atheism
7  From: lmh@juliet.caltech.edu (Henling, Lawrenc...  alt.atheism
8  From: mangoe@cs.umd.edu (Charley Wingate)\nSub...  alt.atheism
9  Organization: Penn State University\nFrom: <MV...  alt.atheism

   total_missing_count
0                    0
1                    0
2                    0
3                    0
4                    0
5                    0
6                    0
7                    0
8                    0
9                    0
C:\Users\jon29\AppData\Local\Temp\ipykernel_22984\1349963896.py:16: FutureWarnin
g: DataFrame.applymap has been deprecated. Use DataFrame.map instead.
  missing_values = df.applymap(is_missing)
```

In [ ]:
```
#exercise  6
```

In [45]:
```
# 將現有的 df 作為 X 進行取樣
X = df

# 然後進行取樣
X_sample = X.sample(n=1000, random_state=42)

# 查看 X 和 X_sample 的行數
print("X 的行數:", len(X))
print("X_sample 的行數:", len(X_sample))
```

```
X 的行數: 2257
X_sample 的行數: 1000
```

In [46]:
```
# 隨機取樣 1000 條記錄
X_sample = X.sample(n=1000, random_state=42)

# 查看 X 和 X_sample 的行數
print("X 的行數:", len(X))
print("X_sample 的行數:", len(X_sample))
```

```
X 的行數: 2257
X_sample 的行數: 1000
```

In [47]:
```python
# 檢查 X_sample 的索引
print("X_sample 的索引值:\n", X_sample.index[:10])
```

```
X_sample 的索引值:
 Index([561, 440, 1513, 1360, 259, 535, 809, 2002, 2166, 1272], dtype='int64')
```

In [48]:
```python
# 計算 X 和 X_sample 中每個類別的分佈
print("X 中的類別分佈:\n", X['category'].value_counts())
print("\nX_sample 中的類別分佈:\n", X_sample['category'].value_counts())
```

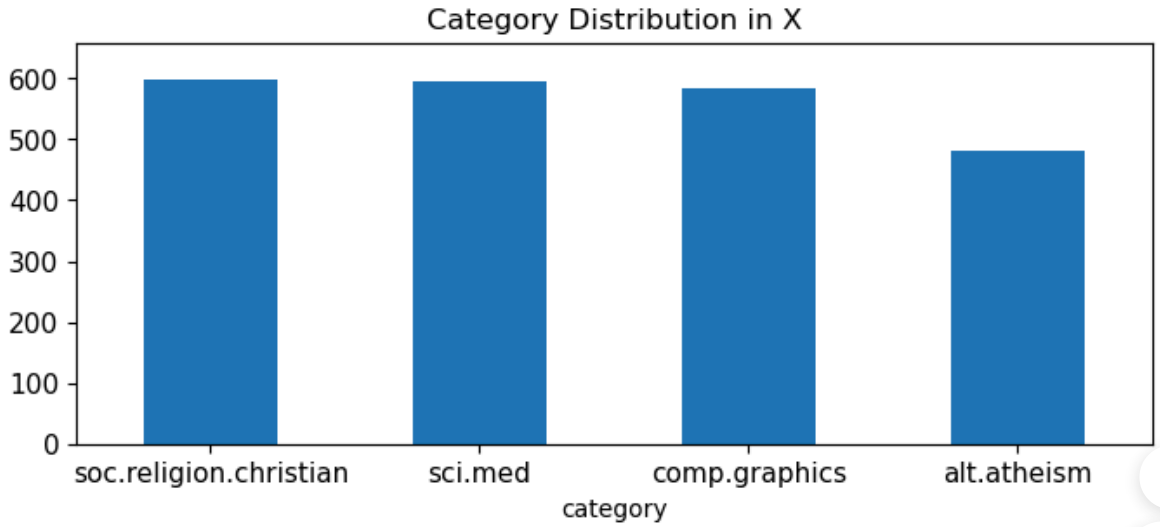
```
X 中的類別分佈:
 category
soc.religion.christian    599
sci.med                   594
comp.graphics             584
alt.atheism               480
Name: count, dtype: int64

X_sample 中的類別分佈:
 category
soc.religion.christian    269
sci.med                   263
comp.graphics             249
alt.atheism               219
Name: count, dtype: int64
```

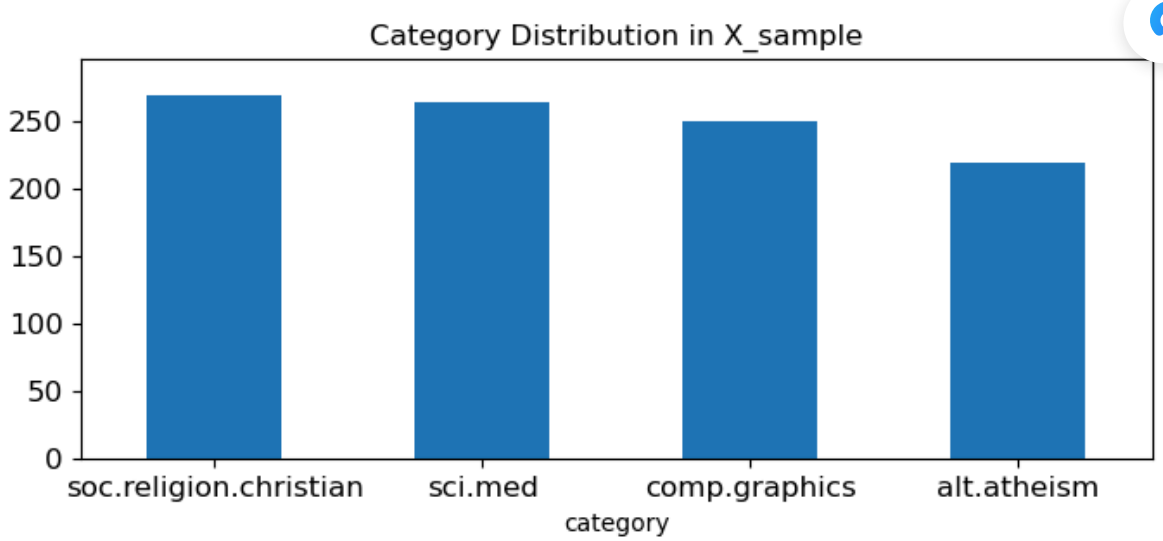
In [ ]:
```python
#總結變化
#X_sample 行數減少為 1000
#X_sample 保留了 X 的原始索引值，導致索引不連續
#類別分佈在 X_sample 中與 X 基本一致，但記錄數量減少符合預期
```

In [ ]:
```python

```

In [ ]:
```python
#exercise  7
```

In [49]:
```python
import matplotlib.pyplot as plt

# 自動設置 X 的 y 軸最大值
y_max_X = X['category'].value_counts().max() * 1.1  # 增加 10% 空間
X['category'].value_counts().plot(kind='bar',
                                  title='Category Distribution in X',
                                  ylim=[0, y_max_X],
                                  rot=0, fontsize=11, figsize=(8, 3))
plt.show()

# 自動設置 X_sample 的 y 軸最大值
y_max_X_sample = X_sample['category'].value_counts().max() * 1.1
X_sample['category'].value_counts().plot(kind='bar',
                                         title='Category Distribution in X_sampl
                                         ylim=[0, y_max_X_sample],
                                         rot=0, fontsize=12, figsize=(8, 3))
plt.show()
```

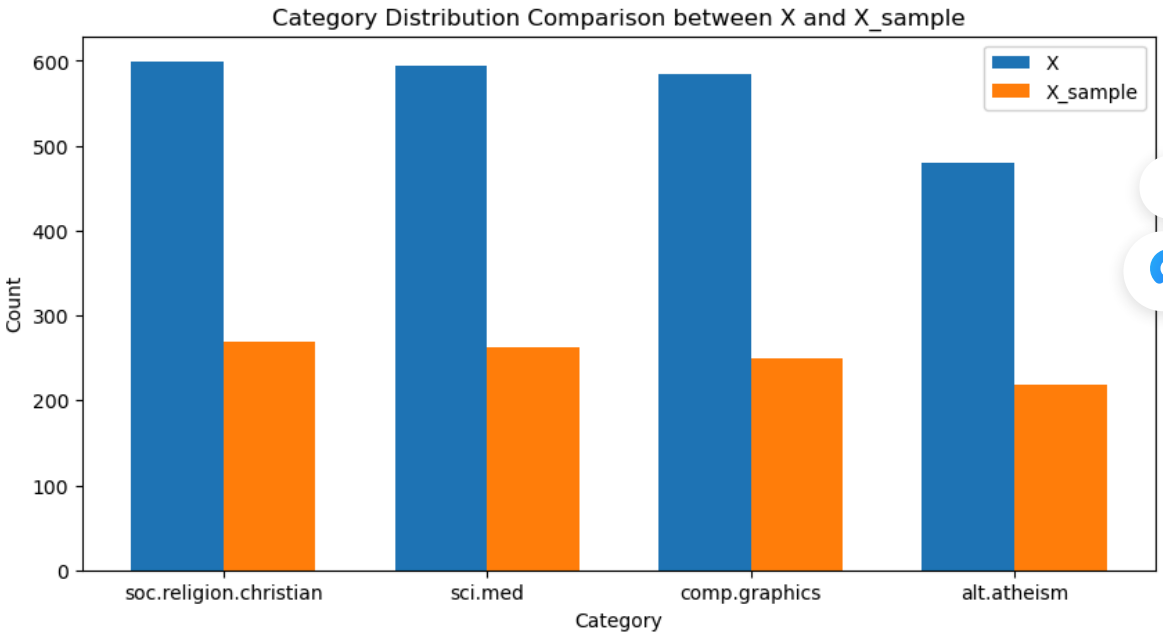




In [ ]:
```
#exercise  8
```

In [50]:
```
import matplotlib.pyplot as plt
import numpy as np

# 計算類別分佈
category_counts_X = X['category'].value_counts()
category_counts_X_sample = X_sample['category'].value_counts()

# 確保順序一致
categories = category_counts_X.index

# 設置條形寬度和位置
bar_width = 0.35
index = np.arange(len(categories))

# 創建並排的條形圖
fig, ax = plt.subplots(figsize=(10, 5))

# 繪製 X 的條形圖
bars_X = ax.bar(index, category_counts_X[categories], bar_width, label='X')

# 繪製 X_sample 的條形圖
bars_X_sample = ax.bar(index + bar_width, category_counts_X_sample[categories],

# 添加標題和標籤
```

```
ax.set_xlabel('Category')
ax.set_ylabel('Count')
ax.set_title('Category Distribution Comparison between X and X_sample')
ax.set_xticks(index + bar_width / 2)
ax.set_xticklabels(categories, rotation=0)
ax.legend()

plt.show()
```



In [ ]:
```
#exercise  9
```

In [51]:
```
from sklearn.feature_extraction.text import CountVectorizer

# 假設 count_vect 已經初始化
count_vect = CountVectorizer()

# 創建一個分析器（tokenizer 函數）
analyze = count_vect.build_analyzer()

# 獲取 X DataFrame 的第一條文本記錄，並將其 tokenized
first_record_tokens = analyze(X['text'].iloc[0])

# 打印分詞結果
print("第一條記錄的分詞結果:\n", first_record_tokens)
```

第一條記錄的分詞結果：

['from', 'rashid', 'subject', 're', 'yet', 'more', 'rushdie', 're', 'islamic',
'law', 'nntp', 'posting', 'host', '47', '252', '179', 'organization', 'nh', 'line
s', '76', 'in', 'article', '1993apr14', '131032', '15644', 'monu6', 'cc', 'monas
h', 'edu', 'au', 'darice', 'yoyo', 'cc', 'monash', 'edu', 'au', 'fred', 'rice',
'wrote', 'it', 'is', 'my', 'understanding', 'that', 'it', 'is', 'generally', 'agr
eed', 'upon', 'by', 'the', 'ulema', 'islamic', 'scholars', 'that', 'islamic', 'la
w', 'applies', 'only', 'in', 'an', 'islamic', 'country', 'of', 'which', 'the', 'u
k', 'is', 'not', 'furthermore', 'to', 'take', 'the', 'law', 'into', 'one', 'own',
'hands', 'is', 'criminal', 'act', 'as', 'these', 'are', 'matters', 'for', 'the',
'state', 'not', 'for', 'individuals', 'nevertheless', 'khomeini', 'offered', 'cas
h', 'prize', 'for', 'people', 'to', 'take', 'the', 'law', 'into', 'their', 'own',
'hands', 'something', 'which', 'to', 'my', 'understanding', 'is', 'against', 'isl
amic', 'law', 'yes', 'this', 'is', 'also', 'my', 'understanding', 'of', 'the', 'm
ajority', 'of', 'islamic', 'laws', 'however', 'believe', 'there', 'are', 'also',
'certain', 'legal', 'rulings', 'which', 'in', 'all', 'five', 'schools', 'of', 'la
w', 'sunni', 'and', 'jaffari', 'can', 'be', 'levelled', 'against', 'muslim', 'o
r', 'non', 'muslims', 'both', 'within', 'and', 'outside', 'dar', 'al', 'islam',
'do', 'not', 'know', 'if', 'apostasy', 'when', 'accompanied', 'by', 'active', 'pe
rsistent', 'and', 'open', 'hostility', 'to', 'islam', 'falls', 'into', 'this', 'c
ategory', 'of', 'the', 'law', 'do', 'know', 'that', 'historically', 'apostasy',
'has', 'very', 'rarely', 'been', 'punished', 'at', 'all', 'let', 'alone', 'by',
'the', 'death', 'penalty', 'my', 'understanding', 'is', 'that', 'khomeini', 'ruli
ng', 'was', 'not', 'based', 'on', 'the', 'law', 'of', 'apostasy', 'alone', 'it',
'was', 'well', 'known', 'that', 'rushdie', 'was', 'an', 'apostate', 'long', 'befo
re', 'he', 'wrote', 'the', 'offending', 'novel', 'and', 'certainly', 'there', 'i
s', 'no', 'precedent', 'in', 'the', 'qur', 'an', 'hadith', 'or', 'in', 'islamic',
'history', 'for', 'indiscriminantly', 'levelling', 'death', 'penalties', 'for',
'apostasy', 'believe', 'the', 'charge', 'levelled', 'against', 'rushdie', 'was',
'that', 'of', 'fasad', 'this', 'ruling', 'applies', 'both', 'within', 'and', 'out
side', 'the', 'domain', 'of', 'an', 'islamic', 'state', 'and', 'it', 'can', 'be',
'carried', 'out', 'by', 'individuals', 'the', 'reward', 'was', 'not', 'offered',
'by', 'khomeini', 'but', 'by', 'individuals', 'within', 'iran', 'stuff', 'delete
d', 'also', 'think', 'you', 'are', 'muddying', 'the', 'issue', 'as', 'you', 'see
m', 'to', 'assume', 'that', 'khomeini', 'fatwa', 'was', 'issued', 'due', 'to', 't
he', '_distribution_', 'of', 'the', 'book', 'my', 'understanding', 'is', 'that',
'khomeini', 'fatwa', 'was', 'issued', 'in', 'response', 'to', 'the', '_writing_',
'and', '_publishing_', 'of', 'the', 'book', 'if', 'my', 'view', 'is', 'correct',
'then', 'your', 'viewpoint', 'that', 'rushdie', 'was', 'sentenced', 'for', 'crim
e', 'in', 'progress', 'is', 'incorrect', 'would', 'concur', 'that', 'the', 'thrus
t', 'of', 'the', 'fatwa', 'from', 'what', 'remember', 'was', 'levelled', 'at', 't
he', 'author', 'and', 'all', 'those', 'who', 'assisted', 'in', 'the', 'publicatio
n', 'of', 'the', 'book', 'however', 'the', 'charge', 'of', 'fasad', 'can', 'encom
pass', 'number', 'of', 'lesser', 'charges', 'remember', 'that', 'when', 'diplomat
ic', 'relations', 'broke', 'off', 'between', 'britain', 'and', 'iran', 'over', 't
he', 'fatwa', 'iran', 'stressed', 'that', 'the', 'condemnation', 'of', 'the', 'au
thor', 'and', 'the', 'removal', 'of', 'the', 'book', 'from', 'circulation', 'wer
e', 'two', 'preliminary', 'conditions', 'for', 'resolving', 'the', 'crisis', 'bu
t', 'you', 'are', 'correct', 'to', 'point', 'out', 'that', 'banning', 'the', 'boo
k', 'was', 'not', 'the', 'main', 'thrust', 'behind', 'the', 'fatwa', 'islamic',
'charges', 'such', 'as', 'fasad', 'are', 'levelled', 'at', 'people', 'not', 'book
s', 'the', 'rushdie', 'situation', 'was', 'followed', 'in', 'iran', 'for', 'sever
al', 'months', 'before', 'the', 'issuance', 'of', 'the', 'fatwa', 'rushdie', 'wen
t', 'on', 'media', 'blitz', 'presenting', 'himself', 'as', 'lone', 'knight', 'gua
rding', 'the', 'sacred', 'values', 'of', 'secular', 'democracy', 'and', 'mockin
g', 'the', 'foolish', 'concerns', 'of', 'people', 'crazy', 'enough', 'to', 'actua
lly', 'hold', 'their', 'religious', 'beliefs', 'as', 'sacred', 'fanning', 'the',
'flames', 'and', 'milking', 'the', 'controversy', 'to', 'boost', 'his', 'image',
'and', 'push', 'the', 'book', 'he', 'was', 'everywhere', 'in', 'the', 'media', 't
hen', 'muslim', 'demonstrators', 'in', 'several', 'countries', 'were', 'killed',

```
'while', 'protesting', 'against', 'the', 'book', 'rushdie', 'appeared', 'momentar
ily', 'concerned', 'then', 'climbed', 'back', 'on', 'his', 'media', 'horse', 't
o', 'once', 'again', 'attack', 'the', 'muslims', 'and', 'defend', 'his', 'sacre
d', 'rights', 'it', 'was', 'at', 'this', 'point', 'that', 'the', 'fatwa', 'on',
'fasad', 'was', 'issued', 'the', 'fatwa', 'was', 'levelled', 'at', 'the', 'perso
n', 'of', 'rushdie', 'any', 'actions', 'of', 'rushdie', 'that', 'feed', 'the', 's
ituation', 'contribute', 'to', 'the', 'legitimization', 'of', 'the', 'ruling', 't
he', 'book', 'remains', 'in', 'circulation', 'not', 'by', 'some', 'independant',
'will', 'of', 'its', 'own', 'but', 'by', 'the', 'will', 'of', 'the', 'author', 'a
nd', 'the', 'publishers', 'the', 'fatwa', 'against', 'the', 'person', 'of', 'rush
die', 'encompasses', 'his', 'actions', 'as', 'well', 'the', 'crime', 'was', 'cert
ainly', 'crime', 'in', 'progress', 'at', 'many', 'levels', 'and', 'was', 'being',
'played', 'out', 'and', 'played', 'up', 'in', 'the', 'the', 'full', 'view', 'of',
'the', 'media', 'not', 'sure', 'about', 'this', 'but', 'think', 'the', 'charge',
'of', 'shatim', 'also', 'applies', 'to', 'rushdie', 'and', 'may', 'be', 'encompa
sed', 'under', 'the', 'umbrella', 'of', 'the', 'fasad', 'ruling']
```

In [ ]:
```python
#exercise  10
```

In [53]:
```python
from sklearn.feature_extraction.text import CountVectorizer

# 初始化 CountVectorizer 並生成 term-document matrix
count_vect = CountVectorizer()
X_counts = count_vect.fit_transform(X['text'])  # 假設 X 是您的文本數據的 DataFram
```

In [54]:
```python
from sklearn.feature_extraction.text import CountVectorizer

# 假設第五條記錄的行索引是4
fifth_record_index = 4

# 獲取第五條記錄的所有詞彙向量並將其轉為普通數組格式
fifth_record_vector = X_counts[fifth_record_index].toarray()

# 找到非零（出現）特徵的索引
non_zero_indices = fifth_record_vector[0].nonzero()[0]

# 獲取這些特徵名稱
terms_in_fifth_record = count_vect.get_feature_names_out()[non_zero_indices]

# 打印所有非零詞彙及其索引
for index, term in zip(non_zero_indices, terms_in_fifth_record):
    print(f"索引 {index} 的詞為 '{term}'")
```

索引 2350 的詞為 '49'
索引 2716 的詞為 '66014'
索引 3958 的詞為 'about'
索引 3980 的詞為 'absent'
索引 4084 的詞為 'accountable'
索引 4489 的詞為 'agar'
索引 4542 的詞為 'ah'
索引 4720 的詞為 'all'
索引 4780 的詞為 'almost'
索引 4852 的詞為 'am'
索引 4938 的詞為 'an'
索引 4951 的詞為 'analogous'
索引 4992 的詞為 'and'
索引 5066 的詞為 'ann'
索引 5195 的詞為 'any'
索引 5373 的詞為 'arbor'
索引 5410 的詞為 'are'
索引 5529 的詞為 'article'
索引 5549 的詞為 'as'
索引 5591 的詞為 'aspects'
索引 5698 的詞為 'at'
索引 5906 的詞為 'available'
索引 5948 的詞為 'away'
索引 6057 的詞為 'bad'
索引 6212 的詞為 'based'
索引 6298 的詞為 'be'
索引 6412 的詞為 'being'
索引 6472 的詞為 'benedikt'
索引 6557 的詞為 'better'
索引 6787 的詞為 'blamed'
索引 7480 的詞為 'but'
索引 7505 的詞為 'by'
索引 7766 的詞為 'can'
索引 7965 的詞為 'case'
索引 8199 的詞為 'certainly'
索引 8336 的詞為 'charitable'
索引 8342 的詞為 'charley'
索引 8553 的詞為 'christian'
索引 8717 的詞為 'claims'
索引 9093 的詞為 'come'
索引 9167 的詞為 'communicate'
索引 9376 的詞為 'conceptualization'
索引 9393 的詞為 'concluding'
索引 9394 的詞為 'conclusion'
索引 9671 的詞為 'contain'
索引 9738 的詞為 'contradict'
索引 9746 的詞為 'contradictory'
索引 9898 的詞為 'core'
索引 10324 的詞為 'cs'
索引 11021 的詞為 'deity'
索引 11235 的詞為 'described'
索引 11243 的詞為 'descriptive'
索引 11484 的詞為 'dictation'
索引 11904 的詞為 'distribution'
索引 12014 的詞為 'do'
索引 12051 的詞為 'does'
索引 12163 的詞為 'doubting'
索引 12266 的詞為 'driving'
索引 12495 的詞為 'earth'
索引 12626 的詞為 'edu'

索引 12761 的詞為 'elements'
索引 13041 的詞為 'engagement'
索引 13046 的詞為 'engin'
索引 13051 的詞為 'engineering'
索引 13503 的詞為 'ever'
索引 13682 的詞為 'exists'
索引 13715 的詞為 'experience'
索引 13716 的詞為 'experienced'
索引 13733 的詞為 'explain'
索引 13996 的詞為 'familiar'
索引 14212 的詞為 'few'
索引 14601 的詞為 'for'
索引 14666 的詞為 'form'
索引 14887 的詞為 'from'
索引 15319 的詞為 'get'
索引 15521 的詞為 'god'
索引 15545 的詞為 'going'
索引 15636 的詞為 'grabs'
索引 15682 的詞為 'grant'
索引 16131 的詞為 'happen'
索引 16216 的詞為 'has'
索引 16254 的詞為 'have'
索引 16393 的詞為 'heinlein'
索引 16400 的詞為 'held'
索引 16482 的詞為 'here'
索引 16563 的詞為 'hides'
索引 16603 的詞為 'him'
索引 16803 的詞為 'honest'
索引 16881 的詞為 'host'
索引 16999 的詞為 'human'
索引 17008 的詞為 'humans'
索引 17029 的詞為 'hundred'
索引 17268 的詞為 'if'
索引 17526 的詞為 'impressed'
索引 17556 的詞為 'in'
索引 17633 的詞為 'inconsistent'
索引 17772 的詞為 'inexact'
索引 17812 的詞為 'infinitely'
索引 17885 的詞為 'ingles'
索引 18119 的詞為 'intellectual'
索引 18474 的詞為 'is'
索引 18551 的詞為 'it'
索引 19737 的詞為 'language'
索引 19924 的詞為 'least'
索引 20048 的詞為 'let'
索引 20198 的詞為 'like'
索引 20253 的詞為 'lines'
索引 20463 的詞為 'long'
索引 20866 的詞為 'makes'
索引 20928 的詞為 'mangoe'
索引 20978 的詞為 'many'
索引 21322 的詞為 'me'
索引 21366 的詞為 'mechanics'
索引 21412 的詞為 'meek'
索引 21592 的詞為 'metaphysics'
索引 21668 的詞為 'michigan'
索引 21801 的詞為 'mimsy'
索引 21955 的詞為 'mistake'
索引 22528 的詞為 'mutually'
索引 23122 的詞為 'nntp'

索引 23250 的詞為 'not'
索引 23264 的詞為 'nothing'
索引 23301 的詞為 'now'
索引 23482 的詞為 'objectivist'
索引 23513 的詞為 'observations'
索引 23516 的詞為 'observed'
索引 23519 的詞為 'observes'
索引 23538 的詞為 'obtains'
索引 23542 的詞為 'obviously'
索引 23610 的詞為 'of'
索引 23733 的詞為 'on'
索引 23741 的詞為 'one'
索引 23746 的詞為 'ones'
索引 23757 的詞為 'only'
索引 23897 的詞為 'ordinary'
索引 23915 的詞為 'organization'
索引 24188 的詞為 'own'
索引 24431 的詞為 'paraphrased'
索引 24504 的詞為 'particularly'
索引 24895 的詞為 'person'
索引 24900 的詞為 'personal'
索引 24903 的詞為 'personally'
索引 24910 的詞為 'perspective'
索引 25009 的詞為 'phenomena'
索引 25553 的詞為 'poor'
索引 25663 的詞為 'posting'
索引 25709 的詞為 'powerful'
索引 26093 的詞為 'problem'
索引 26097 的詞為 'problems'
索引 26284 的詞為 'proposed'
索引 26676 的詞為 'quantum'
索引 26709 的詞為 'question'
索引 26948 的詞為 'rather'
索引 26986 的詞為 'ray'
索引 27031 的詞為 're'
索引 27468 的詞為 'relativity'
索引 27504 的詞為 'religion'
索引 27846 的詞為 'rest'
索引 27950 的詞為 'revelation'
索引 27951 的詞為 'revelations'
索引 28089 的詞為 'right'
索引 28136 的詞為 'risk'
索引 28197 的詞為 'robert'
索引 28306 的詞為 'rosenau'
索引 28331 的詞為 'rough'
索引 28573 的詞為 'said'
索引 28755 的詞為 'say'
索引 28816 的詞為 'scenario'
索引 28863 的詞為 'schmuck'
索引 29130 的詞為 'seem'
索引 29135 的詞為 'seen'
索引 29756 的詞為 'sincerely'
索引 29813 的詞為 'situation'
索引 30068 的詞為 'so'
索引 30173 的詞為 'some'
索引 30659 的詞為 'stars'
索引 31004 的詞為 'strongly'
索引 31077 的詞為 'subject'
索引 31180 的詞為 'such'
索引 31217 的詞為 'suggest'

```
索引 31364 的詞為 'support'
索引 31375 的詞為 'supposedly'
索引 31710 的詞為 'take'
索引 31715 的詞為 'takes'
索引 31716 的詞為 'taking'
索引 31725 的詞為 'talking'
索引 32038 的詞為 'terms'
索引 32139 的詞為 'that'
索引 32142 的詞為 'the'
索引 32152 的詞為 'their'
索引 32164 的詞為 'then'
索引 32202 的詞為 'there'
索引 32221 的詞為 'these'
索引 32249 的詞為 'thing'
索引 32270 的詞為 'this'
索引 32298 的詞為 'thought'
索引 32417 的詞為 'time'
索引 32493 的詞為 'to'
索引 32997 的詞為 'trying'
索引 33302 的詞為 'umd'
索引 33304 的詞為 'umich'
索引 33450 的詞為 'understanding'
索引 33597 的詞為 'university'
索引 33847 的詞為 'us'
索引 33873 的詞為 'uses'
索引 34775 的詞為 'we'
索引 34866 的詞為 'well'
索引 34923 的詞為 'what'
索引 34954 的詞為 'which'
索引 34982 的詞為 'who'
索引 34987 的詞為 'whole'
索引 35057 的詞為 'will'
索引 35107 的詞為 'wingate'
索引 35157 的詞為 'with'
索引 35249 的詞為 'words'
索引 35275 的詞為 'world'
索引 35350 的詞為 'writes'
索引 35583 的詞為 'yeah'
索引 35587 的詞為 'years'
索引 35638 的詞為 'you'
索引 35648 的詞為 'your'
```

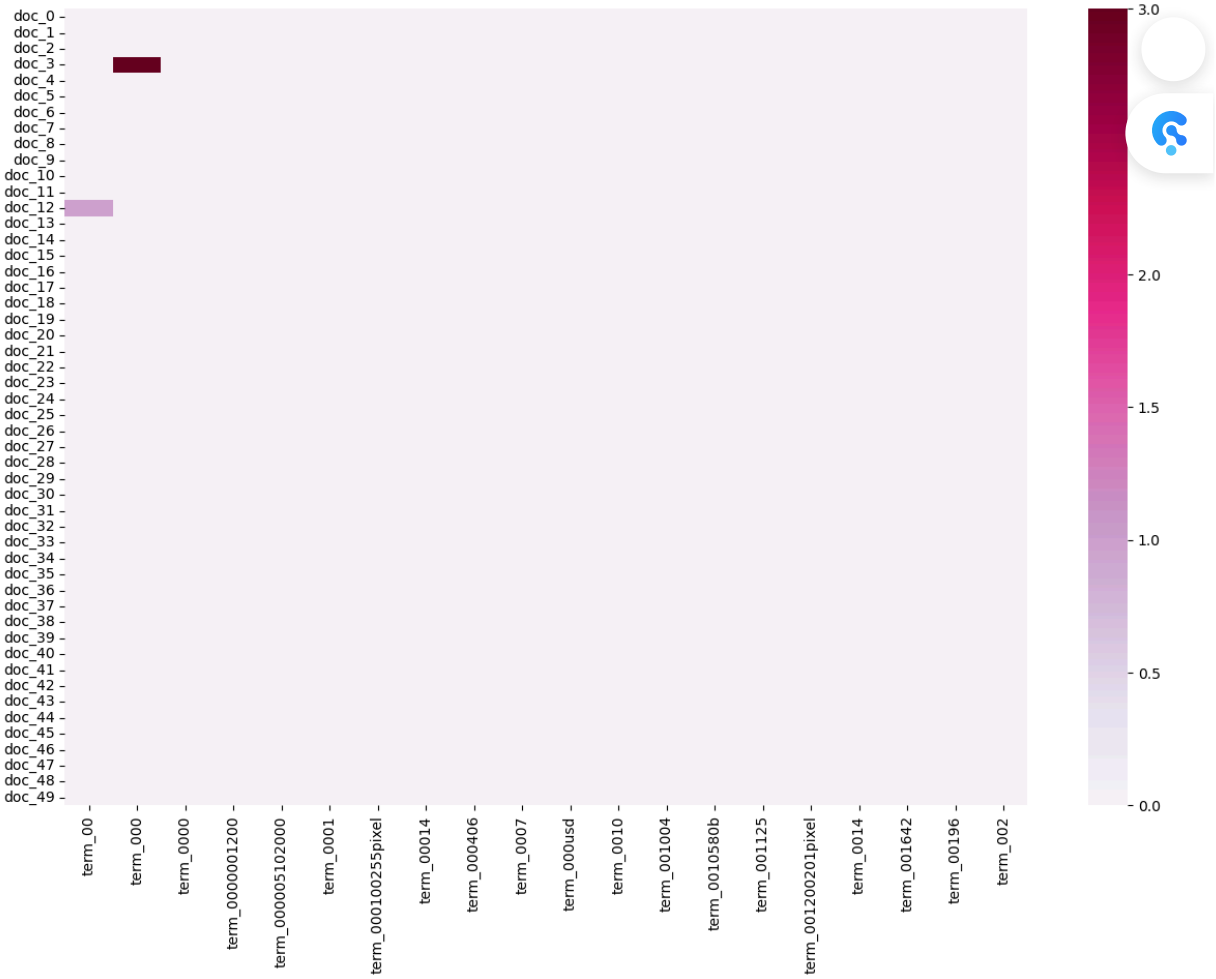
In [ ]:

In [ ]:
```
#Exercise 11
```

In [ ]:
```
# 使用隨機選取的文檔和特徵樣本
sample_docs = 50  # 可以增加到50到100
sample_terms = 100  # 可以增加到100到200
plot_z_sample = X_counts[:sample_docs, :sample_terms].toarray()
```

In [56]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

# 定義文檔和特徵的取樣數量
sample_docs = 50   # 文檔數量樣本
sample_terms = 20  # 特徵數量樣本
```

```
# 使用 count_vect 和 X_counts 的第一部分
plot_x_sample = ["term_" + str(i) for i in count_vect.get_feature_names_out()[:s
plot_y_sample = ["doc_" + str(i) for i in range(sample_docs)]

# 提取 X_counts 的前幾行和列作為樣本
plot_z_sample = X_counts[:sample_docs, :sample_terms].toarray()

# 創建 DataFrame 並繪製熱圖
df_todraw_sample = pd.DataFrame(plot_z_sample, columns=plot_x_sample, index=plot
plt.figure(figsize=(15, 10))
ax = sns.heatmap(df_todraw_sample, cmap="PuRd", vmin=0, vmax=3, annot=False)
plt.show()
```



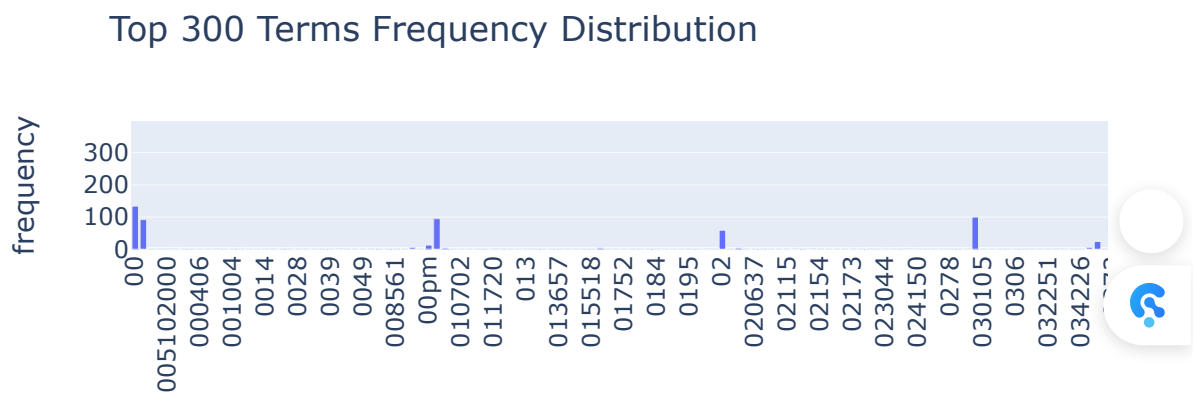
In [ ]:
```
#Exercise 12 13 14
```

In [ ]:
```
pip install plotly
```

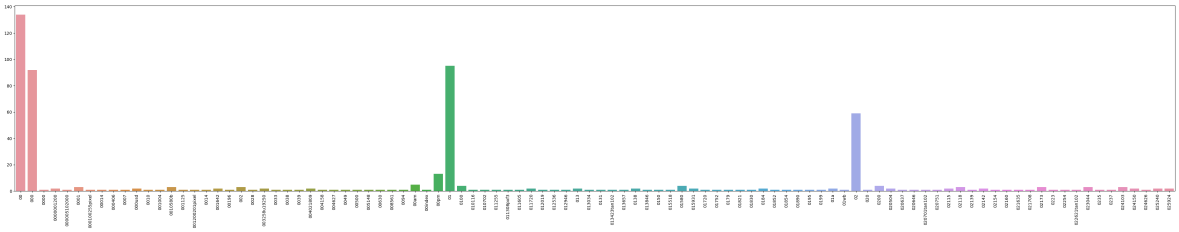
In [59]:
```
import numpy as np


term_frequencies = np.asarray(X_counts.sum(axis=0)).flatten()
```

In [60]:
```
import pandas as pd
import plotly.express as px


term_freq_df = pd.DataFrame({
    'term': count_vect.get_feature_names_out()[:300],
    'frequency': term_frequencies[:300]
```

```
})

# Create an interactive bar chart with Plotly
fig = px.bar(term_freq_df, x='term', y='frequency', title="Top 300 Terms Frequen
fig.update_layout(xaxis_tickangle=-90)
fig.show()
```

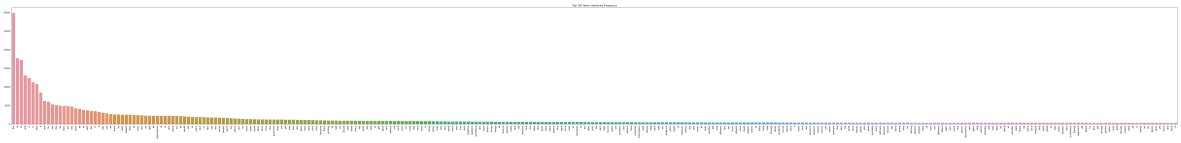


In [61]:
```
plt.subplots(figsize=(50, 8))
g = sns.barplot(x=count_vect.get_feature_names_out()[:100], y=term_frequencies[:
g.set_xticklabels(count_vect.get_feature_names_out()[:100], rotation=90)
plt.show()
```



In [63]:
```
sorted_indices = np.argsort(term_frequencies)[::-1]
sorted_terms = count_vect.get_feature_names_out()[sorted_indices[:300]]
sorted_frequencies = term_frequencies[sorted_indices[:300]]


plt.subplots(figsize=(100, 10))
g = sns.barplot(x=sorted_terms, y=sorted_frequencies)
g.set_xticklabels(sorted_terms, rotation=90)
```

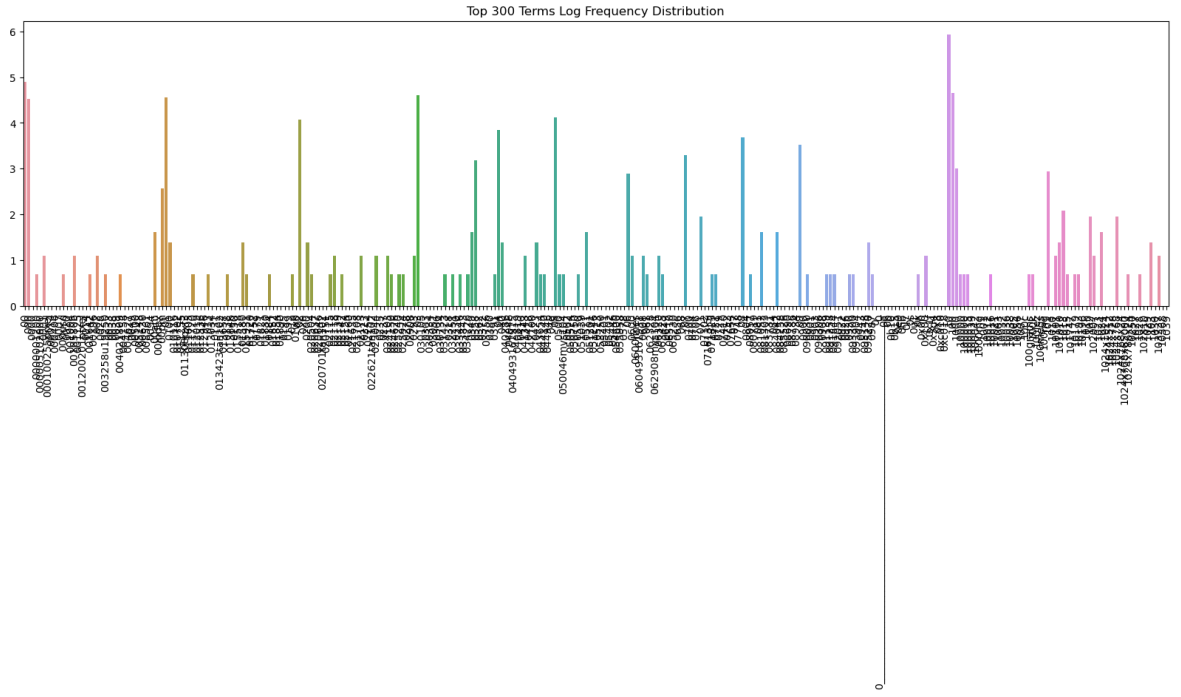
```
plt.title("Top 300 Terms Sorted by Frequency")
plt.show()
```



In [ ]:

In [ ]:
```
#Exercise 15
```

In [64]:
```
import math
import seaborn as sns
import matplotlib.pyplot as plt

# 將每個詞的頻率取對數
term_frequencies_log = [math.log(i) if i > 0 else 0 for i in term_frequencies]

# 繪製對數頻率分布圖
plt.subplots(figsize=(20, 5))  # 縮小尺寸以適應較小畫面
g = sns.barplot(x=count_vect.get_feature_names_out()[:300], y=term_frequencies_l
g.set_xticklabels(count_vect.get_feature_names_out()[:300], rotation=90)
plt.title("Top 300 Terms Log Frequency Distribution")
plt.show()
```



In [ ]:
```
#Exercise 16
```

In [ ]:
```
#相似性：

#各類別的高頻詞通常都是常用詞或停用詞（如 "the"、"is"），在各類別中都會被過濾掉。
#低頻詞是一些特定的專有名詞或稀有詞，不影響分類效果。

#差異性：
#各類別的高頻詞中有一些各自領域的特徵詞，例如 comp.graphics 類別會出現與圖形學相關的詞
#不同類別的低頻詞有各自的專業性，例如 soc.religion.christian 中有宗教專有名詞，而 alt
```

In [ ]:
```
#Exercise 17
```

In [66]:
```
pip install --upgrade PAMI
```

```
Requirement already satisfied: PAMI in c:\users\jon29\anaconda3\lib\site-packages
(2024.10.24.2)
Requirement already satisfied: psutil in c:\users\jon29\anaconda3\lib\site-packag
es (from PAMI) (5.9.0)
Requirement already satisfied: pandas in c:\users\jon29\anaconda3\lib\site-packag
es (from PAMI) (2.1.4)
Requirement already satisfied: plotly in c:\users\jon29\anaconda3\lib\site-packag
es (from PAMI) (5.9.0)
Requirement already satisfied: matplotlib in c:\users\jon29\anaconda3\lib\site-pa
ckages (from PAMI) (3.7.3)
Requirement already satisfied: resource in c:\users\jon29\anaconda3\lib\site-pack
ages (from PAMI) (0.2.1)
Requirement already satisfied: validators in c:\users\jon29\anaconda3\lib\site-pa
ckages (from PAMI) (0.18.2)
Requirement already satisfied: urllib3 in c:\users\jon29\anaconda3\lib\site-pack
ges (from PAMI) (2.0.7)
Requirement already satisfied: Pillow in c:\users\jon29\anaconda3\lib\site-packl
es (from PAMI) (10.2.0)
Requirement already satisfied: numpy in c:\users\jon29\anaconda3\lib\site-package
s (from PAMI) (1.26.4)
Requirement already satisfied: sphinx in c:\users\jon29\anaconda3\lib\site-packag
es (from PAMI) (8.1.3)
Requirement already satisfied: sphinx-rtd-theme in c:\users\jon29\anaconda3\lib\s
ite-packages (from PAMI) (3.0.1)
Requirement already satisfied: discord.py in c:\users\jon29\anaconda3\lib\site-pa
ckages (from PAMI) (2.4.0)
Requirement already satisfied: networkx in c:\users\jon29\anaconda3\lib\site-pack
ages (from PAMI) (3.1)
Requirement already satisfied: deprecated in c:\users\jon29\anaconda3\lib\site-pa
ckages (from PAMI) (1.2.14)
Requirement already satisfied: wrapt<2,>=1.10 in c:\users\jon29\anaconda3\lib\sit
e-packages (from deprecated->PAMI) (1.14.1)
Requirement already satisfied: aiohttp<4,>=3.7.4 in c:\users\jon29\anaconda3\lib
\site-packages (from discord.py->PAMI) (3.9.3)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\jon29\anaconda3\lib\s
ite-packages (from matplotlib->PAMI) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\jon29\anaconda3\lib\site-
packages (from matplotlib->PAMI) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\jon29\anaconda3\lib
\site-packages (from matplotlib->PAMI) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\jon29\anaconda3\lib
\site-packages (from matplotlib->PAMI) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\jon29\anaconda3\lib\si
te-packages (from matplotlib->PAMI) (23.1)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\jon29\anaconda3\lib\s
ite-packages (from matplotlib->PAMI) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\jon29\anaconda3\l
ib\site-packages (from matplotlib->PAMI) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\jon29\anaconda3\lib\site-
packages (from pandas->PAMI) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\jon29\anaconda3\lib\sit
e-packages (from pandas->PAMI) (2023.3)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\jon29\anaconda3\lib\si
te-packages (from plotly->PAMI) (8.2.2)
Requirement already satisfied: JsonForm>=0.0.2 in c:\users\jon29\anaconda3\lib\si
te-packages (from resource->PAMI) (0.0.2)
Requirement already satisfied: JsonSir>=0.0.2 in c:\users\jon29\anaconda3\lib\sit
e-packages (from resource->PAMI) (0.0.2)
Requirement already satisfied: python-easyconfig>=0.1.0 in c:\users\jon29\anacond
a3\lib\site-packages (from resource->PAMI) (0.1.7)
```

```
Requirement already satisfied: sphinxcontrib-applehelp>=1.0.7 in c:\users\jon29\a
naconda3\lib\site-packages (from sphinx->PAMI) (2.0.0)
Requirement already satisfied: sphinxcontrib-devhelp>=1.0.6 in c:\users\jon29\ana
conda3\lib\site-packages (from sphinx->PAMI) (2.0.0)
Requirement already satisfied: sphinxcontrib-htmlhelp>=2.0.6 in c:\users\jon29\an
aconda3\lib\site-packages (from sphinx->PAMI) (2.1.0)
Requirement already satisfied: sphinxcontrib-jsmath>=1.0.1 in c:\users\jon29\anac
onda3\lib\site-packages (from sphinx->PAMI) (1.0.1)
Requirement already satisfied: sphinxcontrib-qthelp>=1.0.6 in c:\users\jon29\anac
onda3\lib\site-packages (from sphinx->PAMI) (2.0.0)
Requirement already satisfied: sphinxcontrib-serializinghtml>=1.1.9 in c:\users\j
on29\anaconda3\lib\site-packages (from sphinx->PAMI) (2.0.0)
Requirement already satisfied: Jinja2>=3.1 in c:\users\jon29\anaconda3\lib\site-p
ackages (from sphinx->PAMI) (3.1.3)
Requirement already satisfied: Pygments>=2.17 in c:\users\jon29\anaconda3\lib\si
e-packages (from sphinx->PAMI) (2.18.0)
Requirement already satisfied: docutils<0.22,>=0.20 in c:\users\jon29\anaconda3
ib\site-packages (from sphinx->PAMI) (0.21.2)
Requirement already satisfied: snowballstemmer>=2.2 in c:\users\jon29\anaconda3\l
ib\site-packages (from sphinx->PAMI) (2.2.0)
Requirement already satisfied: babel>=2.13 in c:\users\jon29\anaconda3\lib\site-p
ackages (from sphinx->PAMI) (2.16.0)
Requirement already satisfied: alabaster>=0.7.14 in c:\users\jon29\anaconda3\lib
\site-packages (from sphinx->PAMI) (1.0.0)
Requirement already satisfied: imagesize>=1.3 in c:\users\jon29\anaconda3\lib\sit
e-packages (from sphinx->PAMI) (1.4.1)
Requirement already satisfied: requests>=2.30.0 in c:\users\jon29\anaconda3\lib\s
ite-packages (from sphinx->PAMI) (2.31.0)
Requirement already satisfied: colorama>=0.4.6 in c:\users\jon29\anaconda3\lib\si
te-packages (from sphinx->PAMI) (0.4.6)
Requirement already satisfied: sphinxcontrib-jquery<5,>=4 in c:\users\jon29\anaco
nda3\lib\site-packages (from sphinx-rtd-theme->PAMI) (4.1)
Requirement already satisfied: six>=1.4.0 in c:\users\jon29\anaconda3\lib\site-pa
ckages (from validators->PAMI) (1.16.0)
Requirement already satisfied: decorator>=3.4.0 in c:\users\jon29\anaconda3\lib\s
ite-packages (from validators->PAMI) (5.1.1)
Requirement already satisfied: aiosignal>=1.1.2 in c:\users\jon29\anaconda3\lib\s
ite-packages (from aiohttp<4,>=3.7.4->discord.py->PAMI) (1.2.0)
Requirement already satisfied: attrs>=17.3.0 in c:\users\jon29\anaconda3\lib\site
-packages (from aiohttp<4,>=3.7.4->discord.py->PAMI) (23.1.0)
Requirement already satisfied: frozenlist>=1.1.1 in c:\users\jon29\anaconda3\lib
\site-packages (from aiohttp<4,>=3.7.4->discord.py->PAMI) (1.4.0)
Requirement already satisfied: multidict<7.0,>=4.5 in c:\users\jon29\anaconda3\li
b\site-packages (from aiohttp<4,>=3.7.4->discord.py->PAMI) (6.0.4)
Requirement already satisfied: yarl<2.0,>=1.0 in c:\users\jon29\anaconda3\lib\sit
e-packages (from aiohttp<4,>=3.7.4->discord.py->PAMI) (1.9.3)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\jon29\anaconda3\lib\si
te-packages (from Jinja2>=3.1->sphinx->PAMI) (2.1.3)
Requirement already satisfied: jsonschema in c:\users\jon29\anaconda3\lib\site-pa
ckages (from JsonForm>=0.0.2->resource->PAMI) (4.19.2)
Requirement already satisfied: PyYAML in c:\users\jon29\anaconda3\lib\site-packag
es (from python-easyconfig>=0.1.0->resource->PAMI) (6.0.1)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\jon29\anacond
a3\lib\site-packages (from requests>=2.30.0->sphinx->PAMI) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\jon29\anaconda3\lib\site-
packages (from requests>=2.30.0->sphinx->PAMI) (3.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\jon29\anaconda3\lib
\site-packages (from requests>=2.30.0->sphinx->PAMI) (2024.2.2)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in c:\users\j
on29\anaconda3\lib\site-packages (from jsonschema->JsonForm>=0.0.2->resource->PAM
```

```
I) (2023.7.1)
Requirement already satisfied: referencing>=0.28.4 in c:\users\jon29\anaconda3\li
b\site-packages (from jsonschema->JsonForm>=0.0.2->resource->PAMI) (0.30.2)
Requirement already satisfied: rpds-py>=0.7.1 in c:\users\jon29\anaconda3\lib\sit
e-packages (from jsonschema->JsonForm>=0.0.2->resource->PAMI) (0.10.6)
Note: you may need to restart the kernel to use updated packages.
```

In [ ]:
```
#Exercise 18
```

In [74]:
```
print(X.columns)
```

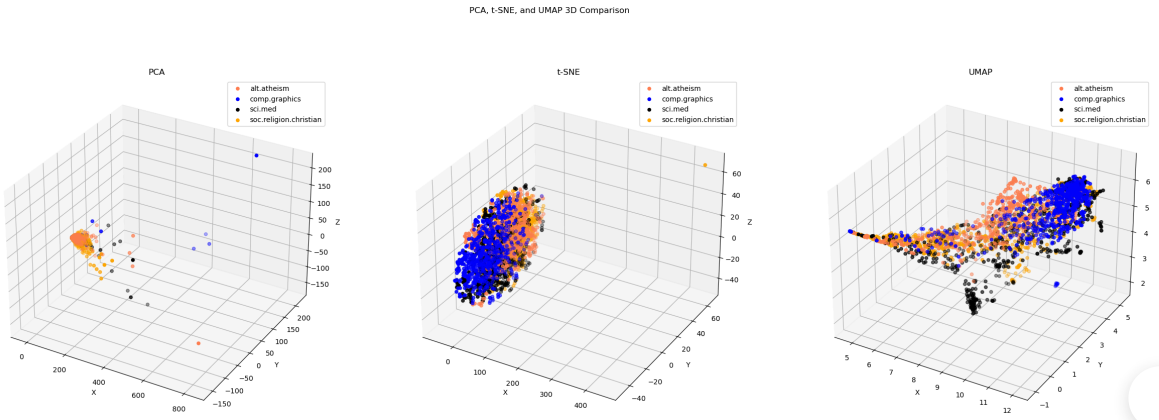
```
Index(['text', 'category', 'missing_values_count', 'total_missing_count'], dtype
='object')
```

In [77]:
```
categories = X['category'].unique()
```

In [79]:
```
from sklearn.feature_extraction.text import CountVectorizer

# 使用 CountVectorizer 構建文件-詞頻矩陣
count_vect = CountVectorizer()
X_tdm = count_vect.fit_transform(X['text'])  # 使用您 DataFrame 中的 'text' 欄位
tdm_df = pd.DataFrame(X_tdm.toarray(), columns=count_vect.get_feature_names_out(
```

In [80]:
```
# 設定顏色和類別
col = ['coral', 'blue', 'black', 'orange']
categories = X['category'].unique()  # 使用 'category' 欄位來獲取類別標籤

# 應用 PCA、t-SNE 和 UMAP 降維至 3 維
X_pca_3d = PCA(n_components=3).fit_transform(tdm_df.values)
X_tsne_3d = TSNE(n_components=3).fit_transform(tdm_df.values)
X_umap_3d = umap.UMAP(n_components=3).fit_transform(tdm_df.values)

# 3D 繪圖設置
fig = plt.figure(figsize=(30, 10))
fig.suptitle('PCA, t-SNE, and UMAP 3D Comparison')

# 定義繪製 3D 散佈圖的函數
def plot_3d_scatter(ax, X_reduced, title):
    for c, category in zip(col, categories):
        xs = X_reduced[X['category'] == category][:, 0]
        ys = X_reduced[X['category'] == category][:, 1]
        zs = X_reduced[X['category'] == category][:, 2]
        ax.scatter(xs, ys, zs, c=c, marker='o', label=category)

    ax.set_title(title)
    ax.set_xlabel('X')
    ax.set_ylabel('Y')
    ax.set_zlabel('Z')
    ax.legend(loc='upper right')

# 建立 3 個子圖來展示 PCA、t-SNE 和 UMAP 的結果
ax1 = fig.add_subplot(131, projection='3d')
plot_3d_scatter(ax1, X_pca_3d, 'PCA')

ax2 = fig.add_subplot(132, projection='3d')
plot_3d_scatter(ax2, X_tsne_3d, 't-SNE')

ax3 = fig.add_subplot(133, projection='3d')
plot_3d_scatter(ax3, X_umap_3d, 'UMAP')
```

```
plt.show()
```



In [ ]:
```
#exercise 19
```

In [82]:
```
from sklearn import preprocessing

# 初始化 LabelBinarizer 並適用於 category 欄位
mlb = preprocessing.LabelBinarizer()
mlb.fit(X['category'])  # 使用 category 欄位進行擬合
X['bin_category'] = mlb.transform(X['category']).tolist()  # 生成二值化欄位

# 顯示結果
print(X[['category', 'bin_category']].head(10))
```

```
      category  bin_category
0  alt.atheism  [1, 0, 0, 0]
1  alt.atheism  [1, 0, 0, 0]
2  alt.atheism  [1, 0, 0, 0]
3  alt.atheism  [1, 0, 0, 0]
4  alt.atheism  [1, 0, 0, 0]
5  alt.atheism  [1, 0, 0, 0]
6  alt.atheism  [1, 0, 0, 0]
7  alt.atheism  [1, 0, 0, 0]
8  alt.atheism  [1, 0, 0, 0]
9  alt.atheism  [1, 0, 0, 0]
```

```
C:\Users\jon29\anaconda3\Lib\site-packages\sklearn\utils\validation.py:605: Depre
cationWarning:

is_sparse is deprecated and will be removed in a future version. Check `isinstanc
e(dtype, pd.SparseDtype)` instead.

C:\Users\jon29\anaconda3\Lib\site-packages\sklearn\utils\validation.py:614: Depre
cationWarning:

is_sparse is deprecated and will be removed in a future version. Check `isinstanc
e(dtype, pd.SparseDtype)` instead.

C:\Users\jon29\anaconda3\Lib\site-packages\sklearn\utils\validation.py:605: Depre
cationWarning:

is_sparse is deprecated and will be removed in a future version. Check `isinstanc
e(dtype, pd.SparseDtype)` instead.

C:\Users\jon29\anaconda3\Lib\site-packages\sklearn\utils\validation.py:614: Depre
cationWarning:

is_sparse is deprecated and will be removed in a future version. Check `isinstanc
e(dtype, pd.SparseDtype)` instead.

C:\Users\jon29\anaconda3\Lib\site-packages\sklearn\utils\validation.py:605: Depre
cationWarning:

is_sparse is deprecated and will be removed in a future version. Check `isinstanc
e(dtype, pd.SparseDtype)` instead.

C:\Users\jon29\anaconda3\Lib\site-packages\sklearn\utils\validation.py:614: Depre
cationWarning:

is_sparse is deprecated and will be removed in a future version. Check `isinstanc
e(dtype, pd.SparseDtype)` instead.

C:\Users\jon29\anaconda3\Lib\site-packages\sklearn\utils\validation.py:605: Depre
cationWarning:

is_sparse is deprecated and will be removed in a future version. Check `isinstanc
e(dtype, pd.SparseDtype)` instead.

C:\Users\jon29\anaconda3\Lib\site-packages\sklearn\utils\validation.py:614: Depre
cationWarning:

is_sparse is deprecated and will be removed in a future version. Check `isinstanc
e(dtype, pd.SparseDtype)` instead.

C:\Users\jon29\anaconda3\Lib\site-packages\sklearn\utils\validation.py:605: Depre
cationWarning:

is_sparse is deprecated and will be removed in a future version. Check `isinstanc
e(dtype, pd.SparseDtype)` instead.

C:\Users\jon29\anaconda3\Lib\site-packages\sklearn\utils\validation.py:614: Depre
cationWarning:

is_sparse is deprecated and will be removed in a future version. Check `isinstanc
e(dtype, pd.SparseDtype)` instead.
```

```
C:\Users\jon29\anaconda3\Lib\site-packages\sklearn\utils\validation.py:605: Depre
cationWarning:

is_sparse is deprecated and will be removed in a future version. Check `isinstanc
e(dtype, pd.SparseDtype)` instead.

C:\Users\jon29\anaconda3\Lib\site-packages\sklearn\utils\validation.py:614: Depre
cationWarning:

is_sparse is deprecated and will be removed in a future version. Check `isinstanc
e(dtype, pd.SparseDtype)` instead.

C:\Users\jon29\anaconda3\Lib\site-packages\sklearn\utils\validation.py:605: Depre
cationWarning:

is_sparse is deprecated and will be removed in a future version. Check `isinstanc
e(dtype, pd.SparseDtype)` instead.

C:\Users\jon29\anaconda3\Lib\site-packages\sklearn\utils\validation.py:614: Depre
cationWarning:

is_sparse is deprecated and will be removed in a future version. Check `isinstanc
e(dtype, pd.SparseDtype)` instead.
```

In [ ]: