AIM:Building Robust spaCy Pipelines for Noisy Social Media Data Dataset Source

Twitter US Airline Sentiment Dataset

Enable browser notifications in Set to get alerts when executions comple

OK No th

Kaggle: Twitter US Airline Sentiment

Lab Objectives

Handle noisy and informal social media text

Customize a spaCy pipeline for preprocessing

Extract linguistic features and hashtags

Visualize insights from negative tweets

Step 1: Install Required Libraries and Load spaCy English Model

```
# Install libraries (run once)
!pip install spacy pandas matplotlib emoji



# Download spaCy English model
!python -m spacy download en_core_web_sm
import spacy
import pandas as pd
import re
import matplotlib.pyplot as plt
from collections import Counter
```

```
  Downloading emoji-2.15.0-py3-none-any.whl.metadata (5.7 kB)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.12/dist-packa
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.12/dist-packa
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.12/dist-package
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.12/dist-packages (fro
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.12/dist-packages (f
Requirement already satisfied: thinc<8.4.0,>=8.3.4 in /usr/local/lib/python3.12/dist-packages (fro
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.12/dist-packages (fr
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.12/dist-packages (fro
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: weasel<0.5.0,>=0.4.2 in /usr/local/lib/python3.12/dist-packages (fr
Requirement already satisfied: typer-slim<1.0.0,>=0.3.0 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.12/dist-packages (fro
Requirement already satisfied: numpy>=1.19.0 in /usr/local/lib/python3.12/dist-packages (from spac
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.12/d
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.1
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from spacy)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from sp
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from panda
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from par
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from m
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-packages (from matpl
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from 
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from 
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.12/dist-packages (from matplotl
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-packages (from m
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-da
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from reque
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (fron
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-p
Requirement already satisfied: blis<1.4.0,>=1.3.0 in /usr/local/lib/python3.12/dist-p
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.12/
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.12/dist-package
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.1
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in /usr/local/lib/python3.12/
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-pack
Requirement already satisfied: wrapt in /usr/local/lib/python3.12/dist-packages (from
Downloading emoji-2.15.0-py3-none-any.whl (608 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 608.4/608.4 kB 2.4 MB/s eta 0:00:00
Installing collected packages: emoji
Successfully installed emoji-2.15.0
Collecting en-core-web-sm==3.8.0
  Downloading https://github.com/explosion/spacy-models/releases/download/en_core_web_sm-3.8.0/en_
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.8/12.8 MB 107.4 MB/s eta 0:00:00
✔ Download and installation successful
You can now load the package via spacy.load('en_core_web_sm')
⚠ Restart to reload dependencies
If you are in a Jupyter or Colab notebook, you may need to restart Python in
order to load all the package's dependencies. You can do this by selecting the
'Restart kernel' or 'Restart runtime' option.
```

Enable browser notifications in Set to get alerts when executions comple

Step 2: Load the Twitter US Airline Sentiment Dataset

```
# Load dataset (update file path if required)
df = pd.read_csv("/content/Tweets (1).csv")


# Display first few rows
df.head()
```

	tweet_id	airline_sentiment	airline_sentiment_confidence	negativereason	negativerea
0	570306133677760513	neutral	1.0000	NaN	
1	570301130888122368	positive	0.3486	NaN	
2	570301083672813571	neutral	0.6837	NaN	
3	570301031407624196	negative	1.0000	Bad Flight	
4	570300817074462722	negative	1.0000	Can't Tell	

Next steps: Generate code with df New interactive sheet

Step 3: Select Tweet Text and Sentiment Columns and Remove Missing Values

```
# Select required columns
df = df[['text', 'airline_sentiment']]


# Remove missing values
df.dropna(inplace=True)


df.head()
```

Enable browser notifications in Set to get alerts when executions comple

	text	airline_sentiment
0	@VirginAmerica What @dhepburn said.	neutral
1	@VirginAmerica plus you've added commercials t...	positive
2	@VirginAmerica I didn't today... Must mean I n...	neutral
3	@VirginAmerica it's really aggressive to blast...	negative
4	@VirginAmerica and it's a really big bad thing...	negative

Next steps: Generate code with df New interactive sheet

Step 4: Clean Tweets Cleaning Operations:

Remove URLs

Remove mentions (@user)

Remove emojis and special characters

Convert text to lowercase

```
# Tweet cleaning function
def clean_tweet(text):
    text = re.sub(r'http\S+|www\S+', '', text) # Remove URLs
    text = re.sub(r'@\w+', '', text) # Remove mentions
    text = re.sub(r'#[A-Za-z0-9_]+', '', text) # Remove hashtags from clean text
    text = re.sub(r'[^A-Za-z\s]', '', text) # Remove emojis & special characters
    text = text.lower().strip()
    return text


# Apply cleaning
df['clean_text'] = df['text'].apply(clean_tweet)


df[['text', 'clean_text']].head()
```

	text	clean_text
0	@VirginAmerica What @dhepburn said.	what said
1	@VirginAmerica plus you've added commercials t...	plus youve added commercials to the experienc
2	@VirginAmerica I didn't today... Must mean I n...	i didnt today must mean i need to take anoth
3	@VirginAmerica it's really aggressive to blast...	its really aggressive to blast obnoxious ent
4	@VirginAmerica and it's a really big bad thing...	and its a really big bad thing abo

Enable browser notifications in Set to get alerts when executions comple

step 5: Create Cleaned Tweet Corpus

```
cleaned_corpus = df['clean_text'].tolist()


# Sample from corpus
cleaned_corpus[:5]
```

```
['what  said',
 'plus youve added commercials to the experience tacky',
 'i didnt today must mean i need to take another trip',
 'its really aggressive to blast obnoxious entertainment in your guests faces amp they have little
recourse',
 'and its a really big bad thing about it']
```

Step 6: Initialize the spaCy NLP Pipeline

```
nlp = spacy.load("en_core_web_sm")
```

Step 7: Create and Add a Custom spaCy Pipeline Component to Detect Hashtags

```
from spacy.tokens import Doc
from spacy.language import Language


# Register custom extension
Doc.set_extension("hashtags", default=[], force=True)


# Custom hashtag detection component
@Language.component('hashtag_component')
def hashtag_component(doc):
    hashtags = [token.text for token in doc if token.text.startswith('#')]
    doc._.hashtags = hashtags
    return doc


# Add component to spaCy pipeline
nlp.add_pipe('hashtag_component', last=True)


nlp.pipe_names
```

```
['tok2vec',
 'tagger',
 'parser',
 'attribute_ruler',
 'lemmatizer',
 'ner',
 'hashtag_component']
```

Step 8: Process the Cleaned Tweets Using Customized spaCy Pipeline

Enable browser notifications in Set to get alerts when executions comple

```
# Process cleaned tweets
df['doc'] = df['clean_text'].apply(nlp)
```

Step 9: Extract Lemmas and Part-of-Speech Tags

```
# Function to extract lemmas and POS tags
def extract_lemmas_pos(doc):
    return [(token.lemma_, token.pos_) for token in doc if not token.is_stop]


# Apply extraction
df['lemmas_pos'] = df['doc'].apply(extract_lemmas_pos)


df['lemmas_pos'].head()
```

```
                                               lemmas_pos
0                                  [( , SPACE), (say, VERB)]
1  [(plus, CCONJ), (ve, AUX), (add, VERB), (comme...
2   [(not, PART), (today, NOUN), (mean, VERB), (ne...
3      [(aggressive, ADJ), (blast, VERB), (obnoxious,...
4                  [(big, ADJ), (bad, ADJ), (thing, NOUN)]

dtype: object
```

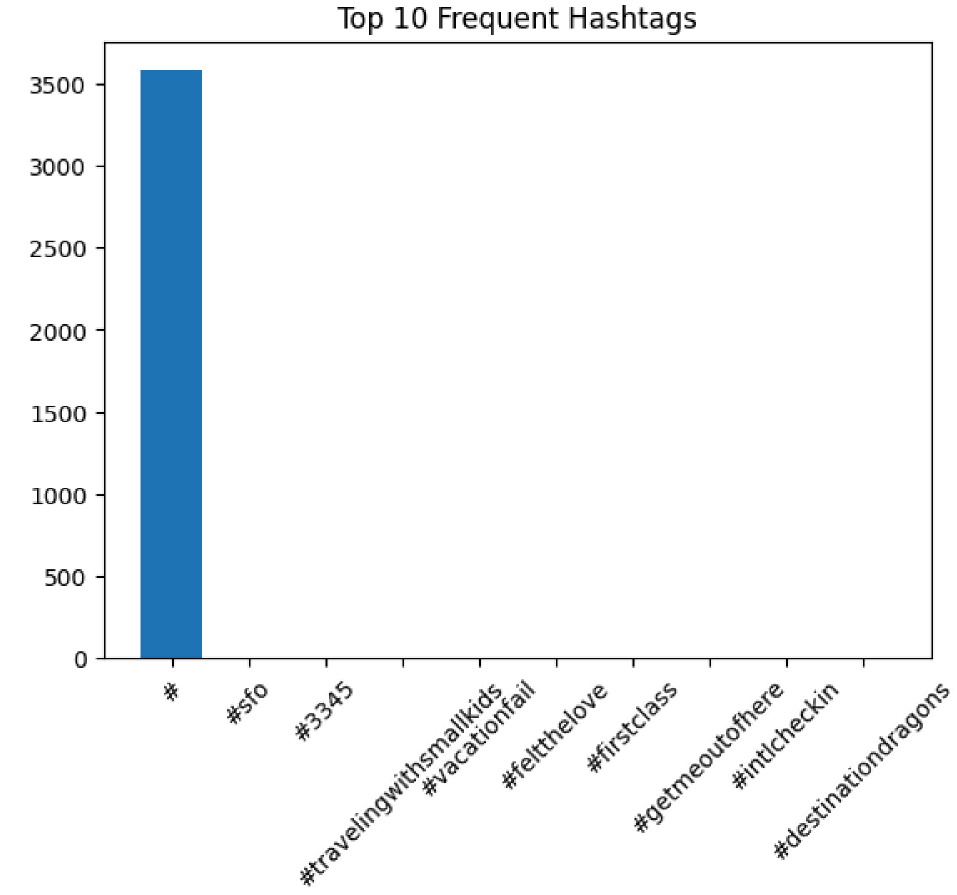
step 10: Extract Hashtags from Original Tweets and Compute Frequencies

```
# Process original tweets to capture hashtags
df['doc_full'] = df['text'].apply(nlp)


# Extract hashtags
all_hashtags = []
for doc in df['doc_full']:
    all_hashtags.extend(doc._.hashtags)


# Compute frequencies
hashtag_freq = Counter([tag.lower() for tag in all_hashtags])


hashtag_freq.most_common(10)
```

```
[('#', 3575),
 ('#sfo', 1),
 ('#3345', 1),
 ('#travelingwithsmallkids', 1),
 ('#vacationfail', 1),
 ('#feltthelove', 1),
 ('#firstclass', 1),
 ('#getmeoutofhere', 1),
 ('#intlcheckin', 1),
 ('#destinationdragons', 1)]
```

Step 11: Visualize the Most Frequent Hashtags

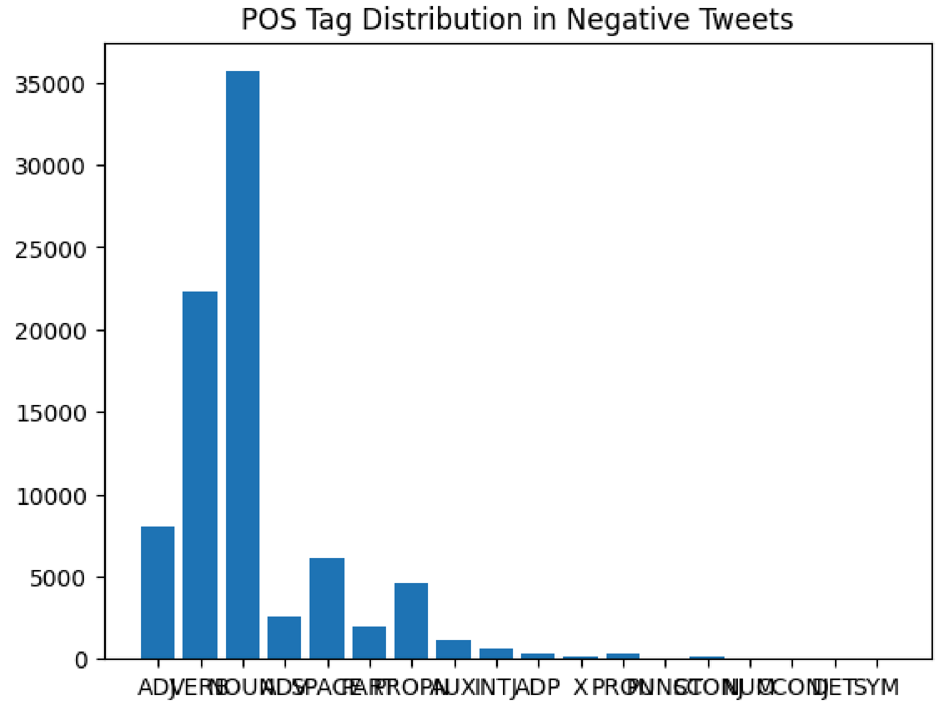
```
# Get top 10 hashtags
top_hashtags = hashtag_freq.most_common(10)
labels, values = zip(*top_hashtags)


plt.figure()
plt.bar(labels, values)
plt.xticks(rotation=45)
plt.title("Top 10 Frequent Hashtags")
plt.show()
```



Step 12: Filter Negative Tweets and Visualize POS Tag Distribution

```
# Filter negative tweets
# Ensure 'clean_text' column exists. If not, create it.
if 'clean_text' not in df.columns:
    # Assuming 'clean_tweet' function is defined from a previous cell (Step 4)
    df['clean_text'] = df['text'].apply(clean_tweet)

# Ensure 'doc' column exists. If not, create it by applying nlp to 'clean_text'.
if 'doc' not in df.columns:
    df['doc'] = df['clean_text'].apply(nlp)
    df['doc'] = df['clean_text'].apply(nlp)

negative_docs = df[df['airline_sentiment'] == 'negative']['doc']


# Collect POS tags
pos_tags = []
for doc in negative_docs:
    pos_tags.extend([token.pos_ for token in doc if not token.is_stop])


# POS frequency
pos_freq = Counter(pos_tags)


plt.figure()
plt.bar(pos_freq.keys(), pos_freq.values())
plt.title("POS Tag Distribution in Negative Tweets")
plt.show()
```

Enable browser
notifications in Set
to get alerts when
executions comple


POS Tag Distribution in Negative Tweets
ADJ
VERB
NOUN
ADV
SPACE
PART
PROPN
AUX
INTJ
ADP
X
PRON
PUNCT
SCONJ
NUM
CCONJ
DET
SYM