Author Name Standardization

Emerson Johnston

```
import os
import pandas as pd
import re

# Directories
output_directory = "/Users/emerson/Github/usenet_webpage"
threads_directory = os.path.join(output_directory, "CSV Files/Threads")
comments_directory = os.path.join(output_directory, "CSV Files/Comments")

# Load cleaned datasets
all_threads = pd.read_csv(os.path.join(threads_directory, "combined_threads.csv"))
all_comments = pd.read_csv(os.path.join(comments_directory, "combined_comments.csv"))

non_name_pattern = r"[^A-Za-z\s\-.]"
filtered_authors = all_comments[all_comments['Author'].str.contains(non_name_pattern, na=False, regex=T

def extract_real_name(full_text):
    if not isinstance(full_text, str):
        return None
    name_pattern = r"[---]? *(by|from)? *([A-Z][a-z]+(?: [A-Z][a-z]+)+)$"
    match = re.search(name_pattern, full_text)
    if match:
        return match.group(2)
    return None

filtered_authors.loc[:, 'Extracted_Real_Name'] = filtered_authors['Full.Text'].apply(extract_real_name)
```

```
## <string>:2: SettingWithCopyWarning:
## A value is trying to be set on a copy of a slice from a DataFrame.
## Try using .loc[row_indexer,col_indexer] = value instead
##
## See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexi
```

```
filtered_with_real_names = filtered_authors[filtered_authors['Extracted_Real_Name'].notnull()]

author_name_mapping = filtered_with_real_names.set_index('Author')['Extracted_Real_Name'].to_dict()

# Add manual mappings for stragglers
manual_mappings = {
    "wer...@aecom.uucp": "Craig Werner",
    "mi...@tekecs.uucp": "Mike Sellers",
    "#D.ANDERSON": "Dave Anderson",
    "SEVENER": "Tim Sevener",
```

```
    "The Polymath": "Jerry Hollombe",
    "fau...@ucbcad.uucp": "Wayne A. Christopher",
    "bi...@persci.uucp": "Bill Swan",
    "pam pincha": "Pam Pincha",
    "stephanie da silva": "Stephanie Da Silva",
    "JB": "Beth Christy"
    # Add additional mappings here
}
author_name_mapping.update(manual_mappings)

all_comments_AS = all_comments.copy()

all_comments_AS.rename(columns={'Author': 'Original_Username'}, inplace=True)

all_comments_AS['Author'] = all_comments_AS['Original_Username'].map(author_name_mapping).fillna(all_co

column_order = [
    "Thread.ID",
    "Comment.ID",
    "Unique.Comment.ID",
    "Author",
    "Date.and.Time",
    "Full.Text",
    "URL.String",
    "newsgroup",
    "Original_Username"
]

all_comments_AS = all_comments_AS[column_order]

print(all_comments_AS.head())
```

```
##   Thread.ID Comment.ID  ... newsgroup  Original_Username
## 0   TH01442    CM00001  ...    netmed  mi...@tekecs.uucp
## 1   TH01441    CM00001  ...    netmed  wer...@aecom.uucp
## 2   TH01440    CM00001  ...    netmed  wer...@aecom.uucp
## 3   TH01439    CM00001  ...    netmed  wer...@aecom.uucp
## 4   TH01439    CM00002  ...    netmed       Hank Buurman
##
## [5 rows x 9 columns]
```

```
print(f"Number of authors standardized: {len(all_comments_AS[all_comments_AS['Author'] != all_comments_A
```

```
## Number of authors standardized: 5317
```

```
valid_name_pattern = r"^[A-Z][a-z]+(?: [A-Z][a-z]+| [A-Z]\.)?(?: [A-Z][a-z]+)?$"

still_nonstandard_name = all_comments_AS[
    all_comments_AS['Author'].str.contains(non_name_pattern, na=False, regex=True) &
    ~all_comments_AS['Author'].str.match(valid_name_pattern, na=False)
]

print(f"Number of entries with still weird authors: {len(still_nonstandard_name)}")
```

```
## Number of entries with still weird authors: 8157
```

```
print(still_nonstandard_name.head())
```

```
##     Thread.ID Comment.ID  ... newsgroup                Original_Username
## 47    TH01433    CM00038  ...    netmed              Gabor Fencsik@ex2642
## 68    TH01432    CM00003  ...    netmed  Rob Vetter;1044;92-725;LP=A;60YB
## 99    TH01417    CM00001  ...    netmed               Alan T. Bowler [SDG]
## 114   TH01407    CM00001  ...    netmed         Tom Slone [(415)486-5954]
## 127   TH01405    CM00001  ...    netmed                ki...@kestrel.uucp
##
## [5 rows x 9 columns]
```

```
still_nonstandard_name_output_path = os.path.join(comments_directory, "still_nonstandard_name.csv")
still_nonstandard_name.to_csv(still_nonstandard_name_output_path, index=False)
print(f"Still weird authors saved to: {still_nonstandard_name_output_path}")
```

```
## Still weird authors saved to: /Users/emerson/Github/usenet_webpage/CSV Files/Comments/still_nonstanda
```

```
def extract_clean_name(author):
    """
    Extracts a name from the author string if it matches a valid name pattern.
    Handles cases with weird characters, such as 'Firstname Lastname' or 'Firstname M. Lastname'.
    """
    if not isinstance(author, str):
        return None
    # Enhanced regex to handle names with surrounding noise
    name_pattern = r"([A-Z][a-z]+(?: [A-Z]\.?)?(?: [A-Z][a-z]+)?)"
    match = re.search(name_pattern, author)  # Use search to find names within noisy data
    if match:
        return match.group(1)  # Return the matched name
    return None

still_nonstandard_name['Extracted_Name'] = still_nonstandard_name['Author'].apply(extract_clean_name)
```

```
## <string>:2: SettingWithCopyWarning:
## A value is trying to be set on a copy of a slice from a DataFrame.
## Try using .loc[row_indexer,col_indexer] = value instead
##
## See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexi
```

```
resolved_names = still_nonstandard_name[still_nonstandard_name['Extracted_Name'].notnull()][['Author',

all_comments_AS['Author'] = all_comments_AS['Original_Username'].map(
    resolved_names.set_index('Author')['Extracted_Name']
).fillna(all_comments_AS['Author'])

still_nonstandard_name = still_nonstandard_name[still_nonstandard_name['Extracted_Name'].isnull()].drop

print(f"Number of entries still unresolved: {len(still_nonstandard_name)}")
```

```
## Number of entries still unresolved: 7273
```

```
print(still_nonstandard_name.head())
```

```
##     Thread.ID Comment.ID  ... newsgroup   Original_Username
## 127   TH01405    CM00001  ...    netmed  ki...@kestrel.uucp
## 129   TH01405    CM00003  ...    netmed    jo...@quad1.uucp
## 135   TH01403    CM00003  ...    netmed   pre...@valid.uucp
## 145   TH01400    CM00005  ...    netmed  er...@chronon.uucp
## 155   TH01396    CM00002  ...    netmed  ki...@kestrel.uucp
##
## [5 rows x 9 columns]
```

```
print(f"Number of authors resolved and updated in all_comments_AS: {len(resolved_names)}")
```

```
## Number of authors resolved and updated in all_comments_AS: 226
```

```
print(all_comments_AS.head())
```

```
##   Thread.ID Comment.ID  ... newsgroup  Original_Username
## 0   TH01442    CM00001  ...    netmed  mi...@tekecs.uucp
## 1   TH01441    CM00001  ...    netmed  wer...@aecom.uucp
## 2   TH01440    CM00001  ...    netmed  wer...@aecom.uucp
## 3   TH01439    CM00001  ...    netmed  wer...@aecom.uucp
## 4   TH01439    CM00002  ...    netmed       Hank Buurman
##
## [5 rows x 9 columns]
```

```
still_nonstandard_name_output_path = os.path.join(comments_directory, "still_nonstandard_name.csv")
still_nonstandard_name.to_csv(still_nonstandard_name_output_path, index=False)
print(f"Updated still_nonstandard_name saved to: {still_nonstandard_name_output_path}")
```

```
## Updated still_nonstandard_name saved to: /Users/emerson/Github/usenet_webpage/CSV Files/Comments/stil
```

```
def find_full_name(author, full_text):
    """
    Searches for the last name in Full.Text when the Author column contains only one name.
    """
    if not isinstance(author, str) or not isinstance(full_text, str):
        return None

    escaped_author = re.escape(author)

    if len(author.split()) == 1:
        name_pattern = rf"\b{escaped_author} [A-Z](?:[a-z]+|\.)(?: [A-Z][a-z]+)?\b"  # Matches 'Firstna
        match = re.search(name_pattern, full_text)
        if match:
            return match.group(0)  # Return the full name

    return None

still_nonstandard_name['Extracted_Full_Name'] = still_nonstandard_name.apply(
```

```
    lambda row: find_full_name(row['Author'], row['Full.Text']), axis=1
)

resolved_full_names = still_nonstandard_name[still_nonstandard_name['Extracted_Full_Name'].notnull()][[

all_comments_AS['Author'] = all_comments_AS['Original_Username'].map(
    resolved_full_names.set_index('Author')['Extracted_Full_Name']
).fillna(all_comments_AS['Author'])

still_nonstandard_name = still_nonstandard_name[still_nonstandard_name['Extracted_Full_Name'].isnull()]

if 'Possible_Full_Name' in all_comments_AS.columns:
    all_comments_AS.drop(columns=['Possible_Full_Name'], inplace=True)

print(f"Number of entries still unresolved: {len(still_nonstandard_name)}")
```

```
## Number of entries still unresolved: 7272
```

```
print(still_nonstandard_name.head())
```

```
##     Thread.ID Comment.ID  ... newsgroup   Original_Username
## 127   TH01405    CM00001  ...    netmed  ki...@kestrel.uucp
## 129   TH01405    CM00003  ...    netmed    jo...@quad1.uucp
## 135   TH01403    CM00003  ...    netmed   pre...@valid.uucp
## 145   TH01400    CM00005  ...    netmed  er...@chronon.uucp
## 155   TH01396    CM00002  ...    netmed  ki...@kestrel.uucp
##
## [5 rows x 9 columns]
```

```
print(f"Number of authors updated with full names in all_comments_AS: {len(resolved_full_names)}")
```

```
## Number of authors updated with full names in all_comments_AS: 1
```

```
print(all_comments_AS.head())
```

```
##   Thread.ID Comment.ID  ... newsgroup  Original_Username
## 0   TH01442    CM00001  ...    netmed  mi...@tekecs.uucp
## 1   TH01441    CM00001  ...    netmed  wer...@aecom.uucp
## 2   TH01440    CM00001  ...    netmed  wer...@aecom.uucp
## 3   TH01439    CM00001  ...    netmed  wer...@aecom.uucp
## 4   TH01439    CM00002  ...    netmed       Hank Buurman
##
## [5 rows x 9 columns]
```

```
still_nonstandard_name_output_path = os.path.join(comments_directory, "still_nonstandard_name.csv")
still_nonstandard_name.to_csv(still_nonstandard_name_output_path, index=False)
print(f"Updated still_nonstandard_name saved to: {still_nonstandard_name_output_path}")
```

```
## Updated still_nonstandard_name saved to: /Users/emerson/Github/usenet_webpage/CSV Files/Comments/stil
```

```
updated_authors_count = (all_comments_AS['Author'] != all_comments_AS['Original_Username']).sum()
print(f"Number of authors updated with full names: {updated_authors_count}")
```

```
## Number of authors updated with full names: 6203
```

```
def extract_name_after_dash(full_text):
    """
    Extracts a name from the Full.Text column if it appears after "--" or " --".
    Matches names like "Firstname Lastname", "Firstname Middlename Lastname", or "Firstname M. Lastname
    """
    if not isinstance(full_text, str):
        return None

    # Regex to match names after "--" or " --"
    pattern = r"--\s*([A-Z][a-z]+(?: [A-Z][a-z]+| [A-Z]\.)?(?: [A-Z][a-z]+)?)"
    match = re.search(pattern, full_text)
    if match:
        return match.group(1)  # Return the matched name
    return None

still_nonstandard_name.loc[:, 'Extracted_Name_After_Dash'] = still_nonstandard_name['Full.Text'].apply(

resolved_dash_names = still_nonstandard_name[still_nonstandard_name['Extracted_Name_After_Dash'].notnul

resolved_dash_names = resolved_dash_names.drop_duplicates(subset=['Author']).set_index('Author')

all_comments_AS['Author'] = all_comments_AS['Original_Username'].map(
    resolved_dash_names['Extracted_Name_After_Dash']
).fillna(all_comments_AS['Author'])

still_nonstandard_name = still_nonstandard_name[still_nonstandard_name['Extracted_Name_After_Dash'].isn

print(f"Number of entries still unresolved: {len(still_nonstandard_name)}")
```

```
## Number of entries still unresolved: 5865
```

```
print(still_nonstandard_name.head())
```

```
##     Thread.ID Comment.ID  ... newsgroup   Original_Username
## 127   TH01405    CM00001  ...    netmed  ki...@kestrel.uucp
## 129   TH01405    CM00003  ...    netmed    jo...@quad1.uucp
## 135   TH01403    CM00003  ...    netmed   pre...@valid.uucp
## 155   TH01396    CM00002  ...    netmed  ki...@kestrel.uucp
## 159   TH01393    CM00001  ...    netmed   gn...@oliveb.uucp
##
## [5 rows x 9 columns]
```

```
print(f"Number of authors resolved with names after '--': {len(resolved_dash_names)}")
```

```
## Number of authors resolved with names after '--': 545
```

```
print(all_comments_AS.head())
```

```
##   Thread.ID Comment.ID  ... newsgroup  Original_Username
## 0   TH01442    CM00001  ...    netmed  mi...@tekecs.uucp
## 1   TH01441    CM00001  ...    netmed  wer...@aecom.uucp
## 2   TH01440    CM00001  ...    netmed  wer...@aecom.uucp
## 3   TH01439    CM00001  ...    netmed  wer...@aecom.uucp
## 4   TH01439    CM00002  ...    netmed       Hank Buurman
##
## [5 rows x 9 columns]
```

```
still_nonstandard_name_output_path = os.path.join(comments_directory, "still_nonstandard_name.csv")
still_nonstandard_name.to_csv(still_nonstandard_name_output_path, index=False)
print(f"Updated still_nonstandard_name saved to: {still_nonstandard_name_output_path}")
```

```
## Updated still_nonstandard_name saved to: /Users/emerson/Github/usenet_webpage/CSV Files/Comments/stil
```

```
updated_authors_count = (all_comments_AS['Author'] != all_comments_AS['Original_Username']).sum()
print(f"Number of authors updated with names after '--': {updated_authors_count}")
```

```
## Number of authors updated with names after '--': 9240
```

```
output_file_path = os.path.join(comments_directory, "combined_comments_AS.csv")

all_comments_AS.to_csv(output_file_path, index=False)

print(f"DataFrame saved to: {output_file_path}")
```

```
## DataFrame saved to: /Users/emerson/Github/usenet_webpage/CSV Files/Comments/combined_comments_AS.csv
```