```
import os
import json
import time
import csv
import re
import math
import ast
import unicodedata
from collections import Counter, deque
from pathlib import Path
from IPython.display import HTML, display
import random

import numpy as np
import pandas as pd
import scipy.sparse as sp
from scipy.sparse import csr_matrix, vstack
from scipy.stats import linregress
import matplotlib.pyplot as plt

from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import hamming_loss, jaccard_score
import joblib

import orjson
import kagglehub
from tqdm import tqdm

from pecos.utils.featurization.text.preprocess import Preprocessor
from pecos.xmc.xlinear.model import XLinearModel
from pecos.xmc import Indexer, LabelEmbeddingFactory
from pecos.utils import smat_util
```

```
pd.set_option('display.float_format', '{:,.2f}'.format)
```

```
os.makedirs('data', exist_ok=True)

for number in range(1, 38):
    file_name = f"enwiki_namespace_0/enwiki_namespace_0_{number}.jsonl"
    print(f"\nDownloading {file_name}...")

    try:

        downloaded_path = kagglehub.dataset_download(
            "wikimedia-foundation/wikipedia-structured-contents",
            path=file_name
        )

        base_name = os.path.basename(downloaded_path)

        new_path = os.path.join("data", base_name)

        if os.path.exists(downloaded_path):
            if os.path.exists(new_path):
                os.remove(new_path)
            os.rename(downloaded_path, new_path)
            print(f"Successfully saved to {new_path}")
        else:
            print(f"Warning: Downloaded file not found at {downloaded_path}")

    except Exception as e:
        print(f"Error downloading {file_name}: {str(e)}")

print("\nDownload process completed")
```

```
Downloading enwiki_namespace_0/enwiki_namespace_0_1.jsonl...
Successfully saved to data/enwiki_namespace_0_1.jsonl

Downloading enwiki_namespace_0/enwiki_namespace_0_2.jsonl...
Successfully saved to data/enwiki_namespace_0_2.jsonl

Downloading enwiki_namespace_0/enwiki_namespace_0_3.jsonl...
Successfully saved to data/enwiki_namespace_0_3.jsonl

Downloading enwiki_namespace_0/enwiki_namespace_0_4.jsonl...
Successfully saved to data/enwiki_namespace_0_4.jsonl
```

```
Downloading enwiki_namespace_0/enwiki_namespace_0_5.jsonl...
Successfully saved to data/enwiki_namespace_0_5.jsonl

Downloading enwiki_namespace_0/enwiki_namespace_0_6.jsonl...
Successfully saved to data/enwiki_namespace_0_6.jsonl

Downloading enwiki_namespace_0/enwiki_namespace_0_7.jsonl...
Successfully saved to data/enwiki_namespace_0_7.jsonl

Downloading enwiki_namespace_0/enwiki_namespace_0_8.jsonl...
Successfully saved to data/enwiki_namespace_0_8.jsonl

Downloading enwiki_namespace_0/enwiki_namespace_0_9.jsonl...
Successfully saved to data/enwiki_namespace_0_9.jsonl

Downloading enwiki_namespace_0/enwiki_namespace_0_10.jsonl...
Successfully saved to data/enwiki_namespace_0_10.jsonl

Downloading enwiki_namespace_0/enwiki_namespace_0_11.jsonl...
Successfully saved to data/enwiki_namespace_0_11.jsonl

Downloading enwiki_namespace_0/enwiki_namespace_0_12.jsonl...
Successfully saved to data/enwiki_namespace_0_12.jsonl

Downloading enwiki_namespace_0/enwiki_namespace_0_13.jsonl...
Successfully saved to data/enwiki_namespace_0_13.jsonl

Downloading enwiki_namespace_0/enwiki_namespace_0_14.jsonl...
Successfully saved to data/enwiki_namespace_0_14.jsonl

Downloading enwiki_namespace_0/enwiki_namespace_0_15.jsonl...
Successfully saved to data/enwiki_namespace_0_15.jsonl

Downloading enwiki_namespace_0/enwiki_namespace_0_16.jsonl...
Successfully saved to data/enwiki_namespace_0_16.jsonl

Downloading enwiki_namespace_0/enwiki_namespace_0_17.jsonl...
Successfully saved to data/enwiki_namespace_0_17.jsonl

Downloading enwiki_namespace_0/enwiki_namespace_0_18.jsonl...
Successfully saved to data/enwiki_namespace_0_18.jsonl

Downloading enwiki_namespace_0/enwiki_namespace_0_19.jsonl...
Successfully saved to data/enwiki_namespace_0_19.jsonl
```

```
def parse_json(x):
    if isinstance(x, str):
        try:
            return orjson.loads(x)
        except orjson.JSONDecodeError:
            return None
    elif isinstance(x, float) and math.isnan(x):
        return None
    return x

def extract_links(section):
    links = set()
    stack = [section]
    while stack:
        item = stack.pop()
        if isinstance(item, dict):
            for link in item.get('links', []):
                url = link.get('url')
                if url: links.add(url)
                for img in link.get('images', []):
                    img_url = img.get('content_url')
                    if img_url: links.add(img_url)
            for img in item.get('images', []):
                img_url = img.get('content_url')
                if img_url: links.add(img_url)
            stack.extend(item.get('has_parts', []))
        elif isinstance(item, list):
            stack.extend(item)
    return list(links)

def calculate_section_word_count(section):
    if section is None:
        return 0
    word_count = 0
    queue = deque([section])
    while queue:
        item = queue.popleft()
        if isinstance(item, dict):
            text_value = item.get('value')
            if isinstance(text_value, str):
```

```
                word_count += len(text_value.strip().split())
            queue.extend(item.get('has_parts', []))
        elif isinstance(item, list):
            queue.extend(item)
    return word_count

def create_links_summary_and_counts(base_dir, output_dir):
    output_dir = Path(output_dir)
    output_dir.mkdir(parents=True, exist_ok=True)

    summary_path = output_dir / "all_links_summary.csv"
    link_counts_path = output_dir / "unique_links_counts.csv"

    link_counter = Counter()

    with open(summary_path, mode='w', newline='', encoding='utf-8') as summary_file:
        summary_writer = csv.writer(summary_file)
        summary_writer.writerow(['identifier', 'name', 'url', 'total_num_links', 'text_length', 'link_density'])

        for i in tqdm(range(38), desc="Processing JSONL shards"):
            input_file = Path(base_dir) / f"enwiki_namespace_0_{i}.jsonl"
            if not input_file.exists():
                continue

            with open(input_file, 'rb') as f:
                for line in f:
                    try:
                        data = orjson.loads(line)

                        section_links = extract_links(data.get('sections', []))
                        infobox_links = extract_links(data.get('infoboxes', []))
                        unique_links = list(set(section_links + infobox_links))
                        total_links = len(unique_links)

                        link_counter.update(unique_links)

                        text_length = calculate_section_word_count(data.get('sections', []))
                        link_density = total_links / text_length if text_length > 0 else 0

                        summary_writer.writerow([
                            data['identifier'],
                            data['name'],
                            data.get('url', ''),
                            total_links,
                            text_length,
                            link_density
                        ])

                    except orjson.JSONDecodeError:
                        continue

    with open(link_counts_path, mode='w', newline='', encoding='utf-8') as counts_file:
        counts_writer = csv.writer(counts_file)
        counts_writer.writerow(['url', 'count'])
        for url, count in link_counter.items():
            counts_writer.writerow([url, count])

    tqdm.write(f"Saved per-document summary to: {summary_path}")
    tqdm.write(f"Saved unique link counts to: {link_counts_path}")


base_dir = "data"
output_dir = Path(base_dir) / "processed_output"
output_dir.mkdir(parents=True, exist_ok=True)

create_links_summary_and_counts(base_dir, output_dir)
```

```
Processing JSONL shards: 100%|██████████| 38/38 [28:04<00:00, 44.32s/it]
Saved per-document summary to: data/processed_output/all_links_summary.csv
Saved unique link counts to: data/processed_output/unique_links_counts.csv
```

```
base_dir = "data"
output_dir = Path(base_dir) / "processed_output"
summary_df = pd.read_csv(output_dir / "all_links_summary.csv")
counts_df = pd.read_csv(output_dir / "unique_links_counts.csv")
```

```
summary_df
```

	identifier	name	url	total_num_links	text_length	link_density
0	40477619	2013 Tashkent Open – Doubles	https://en.wikipedia.org/wiki/2013_Tashkent_Op...	39	121	0.32
1	25829972	Thomas Claughton (MP)	https://en.wikipedia.org/wiki/Thomas_Claughton...	15	72	0.21
2	28049975	1999–2000 Newcastle United F.C. season	https://en.wikipedia.org/wiki/1999%E2%80%93200...	37	364	0.10
3	12895105	Gwendolyn Holbrow	https://en.wikipedia.org/wiki/Gwendolyn_Holbrow	21	1063	0.02
4	53572039	2017 İstanbul Cup	https://en.wikipedia.org/wiki/2017_%C4%B0stanb...	69	224	0.31
...	...	...	...	...	...	...

```
url_to_title = dict(zip(summary_df['url'], summary_df['name']))
```

```
counts_df['title'] = counts_df['url'].map(url_to_title)
```

```
missing_titles_mask = counts_df['title'].isna()
```

```
pattern = r'/([^/#?]+)(?:[#?].*)?$'
counts_df.loc[missing_titles_mask, 'title'] = counts_df.loc[missing_titles_mask, 'url'].str.extract(pattern)[0]
```

```
counts_df.sort_values('count', ascending=False)
```

	url	count	title
288	https://upload.wikimedia.org/wikipedia/commons...	485104	6px-Red_pog.svg.png
186	https://en.wikipedia.org/wiki/Taxonomy_(biology)	444000	Taxonomy (biology)
296	https://en.wikipedia.org/wiki/Time_zone	435673	Time zone
198	https://upload.wikimedia.org/wikipedia/commons...	398690	15px-OOjs_UI_icon_edit-ltr.svg.png
116	https://en.wikipedia.org/wiki/United_States	355943	United States
...	...	...	...
8969970	https://geohack.toolforge.org/geohack.php?page...	1	geohack.php
24526271	https://en.wikipedia.org/wiki/Sindhi_Canadians...	1	Sindhi_Canadians
24526270	https://en.wikipedia.org/wiki/Sindhi_Canadians...	1	Sindhi_Canadians
24526269	https://en.wikipedia.org/wiki/Digimon_Vermilli...	1	Digimon_Vermillion
24526268	https://upload.wikimedia.org/wikipedia/commons...	1	220px-Brass_Knuckles_%281927_film%29_lobby_car...

24526277 rows × 3 columns

```
pattern = r'\.(?:svg|png|jpg|jpeg|gif|webp|asp|pdf|html|php|phtml|page|aspx)(?:\?[^/]+)?$'

counts_df = counts_df[
    ~counts_df['title'].isna() &
    ~counts_df['title'].str.contains(pattern, na=False, regex=True, flags=re.IGNORECASE) &
    ~counts_df['title'].str.contains('wiki', na=False, case=False) &
    (counts_df['count'] >= 50)
]
```

```
counts_df['count'].describe()
```

	count
count	554,300.00
mean	273.93
std	2,089.26
min	50.00
25%	66.00
50%	97.00
75%	186.00
max	444,000.00

dtype: float64

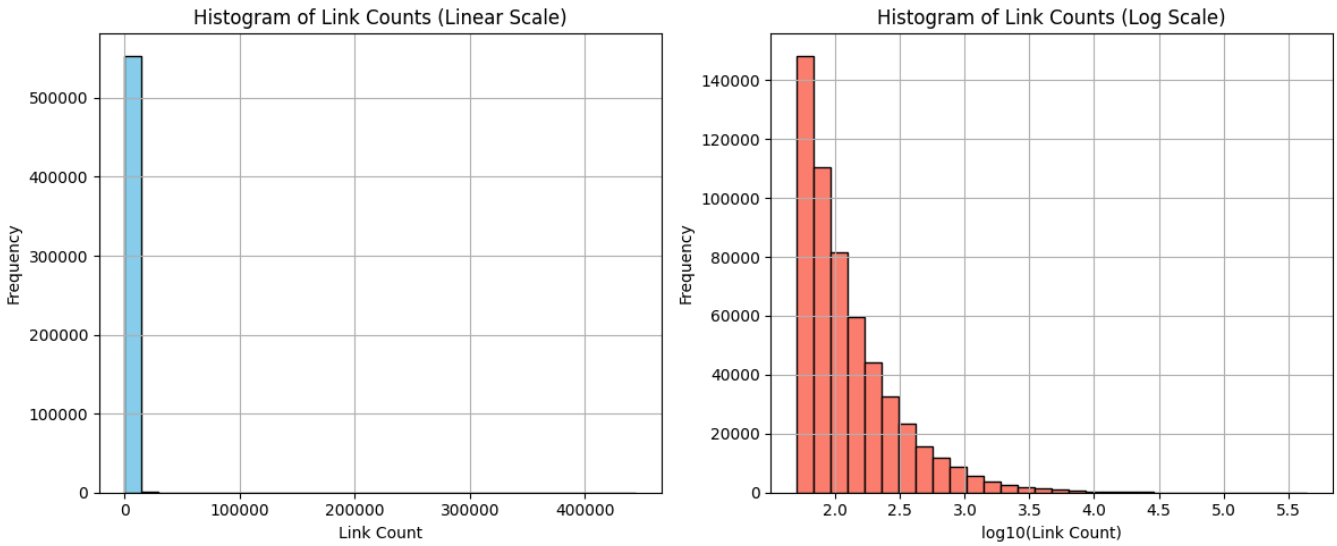
```
counts_df.isna().all()
```

	0
url	False
count	False
title	False

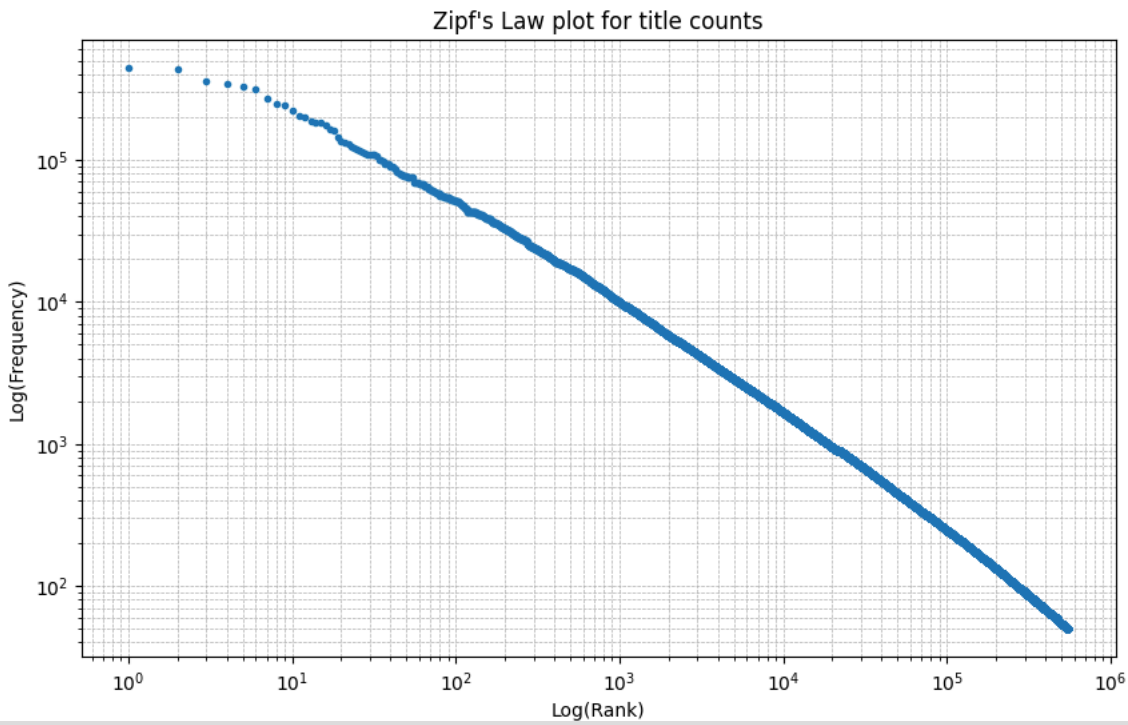
dtype: bool

```
counts_df.to_csv('links.csv',index = False)
```

```
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
plt.hist(counts_df['count'], bins=30, color='skyblue', edgecolor='black')
plt.title('Histogram of Link Counts (Linear Scale)')
plt.xlabel('Link Count')
plt.ylabel('Frequency')
plt.grid(True)


plt.subplot(1, 2, 2)
counts = counts_df['count']
counts = counts[counts > 0]
plt.hist(np.log10(counts), bins=30, color='salmon', edgecolor='black')
plt.title('Histogram of Link Counts (Log Scale)')
plt.xlabel('log10(Link Count)')
plt.ylabel('Frequency')
plt.grid(True)

plt.tight_layout()
plt.show()
```

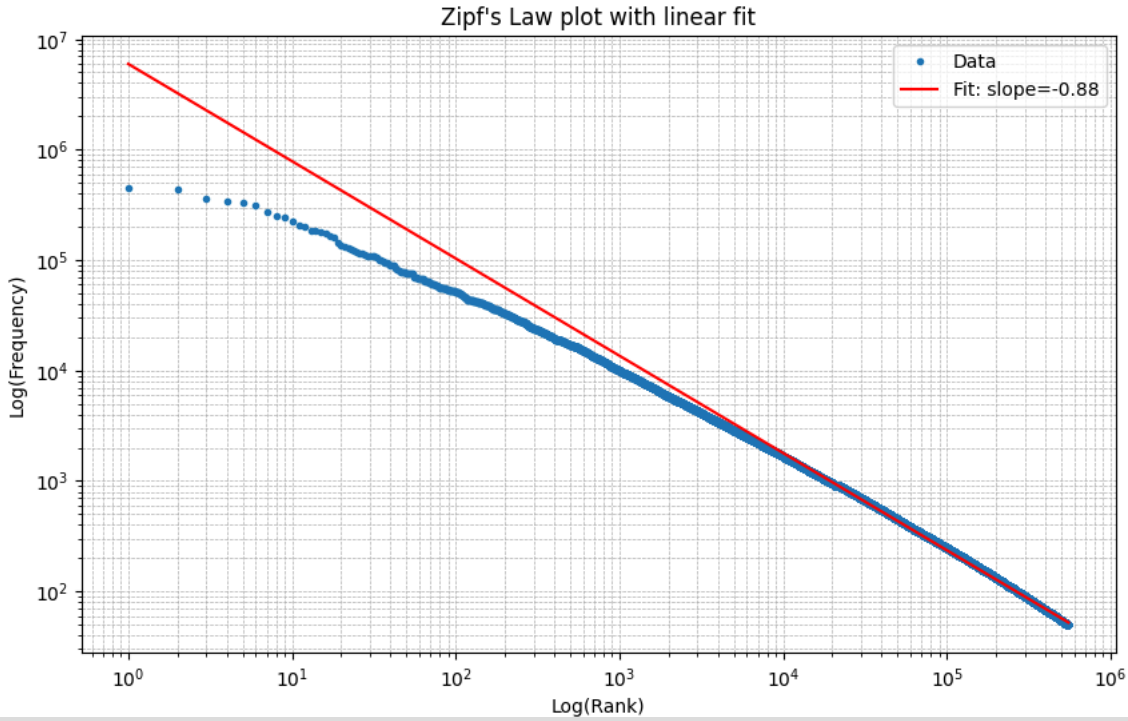


```
sorted_counts = counts_df['count'].sort_values(ascending=False).reset_index(drop=True)

ranks = np.arange(1, len(sorted_counts) + 1)

plt.figure(figsize=(10, 6))
plt.loglog(ranks, sorted_counts, marker=".", linestyle='none')

plt.xlabel("Log(Rank)")
plt.ylabel("Log(Frequency)")
plt.title("Zipf's Law plot for title counts")

plt.grid(True, which="both", ls="--", linewidth=0.5)
plt.show()
```




```
log_ranks = np.log(ranks)
log_counts = np.log(sorted_counts)

slope, intercept, r_value, p_value, std_err = linregress(log_ranks, log_counts)

print(f"Slope: {slope:.3f}, R-squared: {r_value**2:.3f}")

plt.figure(figsize=(10, 6))
plt.loglog(ranks, sorted_counts, marker=".", linestyle='none', label='Data')
plt.loglog(ranks, np.exp(intercept) * ranks ** slope, label=f'Fit: slope={slope:.2f}', color='red')

plt.xlabel("Log(Rank)")
plt.ylabel("Log(Frequency)")
plt.title("Zipf's Law plot with linear fit")
plt.legend()
plt.grid(True, which="both", ls="--", linewidth=0.5)
plt.show()
```

```
Slope: -0.880, R-squared: 0.997
```




```
len(summary_df)
```

```
2106911
```

```
summary_df = summary_df[~summary_df['name'].astype(str).str.lower().str.startswith(('index', 'list', 'alphabetical', 'timeli

summary_df = summary_df[(summary_df['total_num_links'] > 25) & (summary_df['text_length'] > 100)]

summary_df = summary_df.sort_values('total_num_links', ascending=False).drop_duplicates(subset='url', keep='first')

summary_df.loc[summary_df['name'].isna(), 'name'] = (
    summary_df.loc[summary_df['name'].isna(), 'url']
    .astype(str)
    .apply(lambda x: x.rstrip('/').split('/')[-1])
)
```

```
idx = summary_df.groupby('identifier')['link_density'].idxmax()
```

```
summary_df = summary_df.loc[idx]
```

```
summary_df
```

	identifier	name	url	total_num_links	text_length	link_density
7062855	12	Anarchism	https://en.wikipedia.org/wiki/Anarchism	431	6519	0.07
5454002	39	Albedo	https://en.wikipedia.org/wiki/Albedo	123	3810	0.03
5981290	290	A	https://en.wikipedia.org/wiki/A	154	2235	0.07
6206740	303	Alabama	https://en.wikipedia.org/wiki/Alabama	737	12213	0.06
6573377	305	Achilles	https://en.wikipedia.org/wiki/Achilles	392	7874	0.05
...	...	...	...	...	...	...
7074534	77880350	Muwaqqar Chalk-Marl Formation	https://en.wikipedia.org/wiki/Muwaqqar_Chalk-M...	30	293	0.10
7070438	77880411	Royal Crescent Court	https://en.wikipedia.org/wiki/Royal_Crescent_C...	28	268	0.10
7092763	77880702	David J. Danelski	https://en.wikipedia.org/wiki/David_J._Danelski	46	1316	0.03

```
sorted_df = summary_df.sort_values('link_density', ascending=False)

training_data = sorted_df.head(100000)
```

```
testing_data = sorted_df.iloc[len(training_data):len(training_data)+1000]
val_data = sorted_df.tail(1000)

training_data.to_csv('training_dataset_page_details.csv', index=False)
testing_data.to_csv('testing_dataset_page_details.csv', index=False)
val_data.to_csv('validation.csv', index=False)
```

```
def parse_json(x):
    if isinstance(x, str):
        try:
            return orjson.loads(x)
        except orjson.JSONDecodeError:
            return None
    elif isinstance(x, float) and math.isnan(x):
        return None
    return x

def extract_all_links(item):
    links = set()
    stack = [item]

    while stack:
        current = stack.pop()

        if isinstance(current, dict):
            for link in current.get('links', []):
                if 'url' in link:
                    links.add(link['url'])

            for img in current.get('images', []):
                if 'content_url' in img:
                    links.add(img['content_url'])

            for value in current.values():
                if isinstance(value, (dict, list)):
                    stack.append(value)

        elif isinstance(current, list):
            stack.extend(current)

    return list(links)

def calculate_word_count(text):
    return len(text.split()) if text.strip() else 0

def extract_sections_recursive(section, path=None):
    if path is None:
        path = []

    current_name = section.get('name') or section.get('type')
    if current_name:
        path = path + [current_name]

    results = []
    has_parts = section.get('has_parts', [])

    if section.get('type') in {'paragraph', 'list_item'} and section.get('value'):
        results.append({
            'path': ' > '.join(path),
            'text': section['value'].strip(),
            'node': section
        })
    elif isinstance(has_parts, list) and has_parts:
        for sub in has_parts:
            results.extend(extract_sections_recursive(sub, path))

    return results

def aggregate_links_from_node(node):
    return extract_all_links(node)


def process_file(input_path, training_pages, testing_pages, validation_pages, output_paths, first_flags, url_to_title):
    processed_pages = {'training': set(), 'testing': set(), 'validation': set()}

    def replace_urls_with_titles(url_list):
        titles = []
        for url in url_list:
            title = url_to_title.get(url)
            if title is None and isinstance(url, str):
                title = url.rstrip('/').split('/')[-1]
            if title:
```

```
                titles.append(title)
        return titles

    with open(input_path, 'rb') as f, tqdm(desc=f"Reading {input_path.name}", unit="lines") as pbar:
        for line in f:
            try:
                obj = orjson.loads(line)
                identifier = obj.get("identifier")
                title = obj.get("name", "")

                dataset = None
                if identifier in training_pages and not training_pages[identifier]['processed']:
                    dataset = 'training'
                elif identifier in testing_pages and not testing_pages[identifier]['processed']:
                    dataset = 'testing'
                elif identifier in validation_pages and not validation_pages[identifier]['processed']:
                    dataset = 'validation'

                if not dataset:
                    pbar.update(1)
                    continue

                obj['sections'] = parse_json(obj.get('sections', []))
                obj['infoboxes'] = parse_json(obj.get('infoboxes', []))

                rows = []
                sections = obj.get('sections', [])
                if not isinstance(sections, list):
                    sections = [sections] if sections else []

                for section in sections:
                    flattened_sections = extract_sections_recursive(section)

                    for item in flattened_sections:
                        section_text = item['text']
                        if not section_text:
                            continue

                        section_links = aggregate_links_from_node(item['node'])
                        infobox_links = extract_all_links(obj.get('infoboxes', []))
                        all_links = list(set(section_links + infobox_links))
                        link_titles = replace_urls_with_titles(all_links)

                        word_count = calculate_word_count(section_text)
                        link_density = round(len(link_titles)/word_count, 5) if word_count > 0 else 0

                        row = {
                            "identifier": identifier,
                            "title": title,
                            "url": obj.get("url"),
                            "section_text": section_text,
                            "unique_links": link_titles,
                            "text_length": word_count,
                            "links_to_text_ratio": link_density,
                            "section_link_count": len(link_titles),
                            "section_path": item['path']
                        }
                        rows.append(row)

                merged_rows = []
                temp_row = None
                for row in rows:
                    if temp_row is None:
                        temp_row = row
                    else:
                        combined_text = temp_row['section_text'] + ' ' + row['section_text']
                        combined_word_count = calculate_word_count(combined_text)

                        if combined_word_count <= 450:
                            temp_row['section_text'] = combined_text
                            temp_row['unique_links'] = list(set(temp_row['unique_links'] + row['unique_links']))
                            temp_row['text_length'] = combined_word_count
                            temp_row['section_link_count'] = len(temp_row['unique_links'])
                            temp_row['links_to_text_ratio'] = round(temp_row['section_link_count'] / temp_row['text_length']
                        else:
                            merged_rows.append(temp_row)
                            temp_row = row

                if temp_row:
                    merged_rows.append(temp_row)

                rows = merged_rows
```

```
                if not rows:
                    pbar.update(1)
                    continue

                df = pd.DataFrame(rows)
                df.to_csv(output_paths[dataset], mode='a', header=first_flags[dataset], index=False)
                first_flags[dataset] = False

                if dataset == 'training':
                    training_pages[identifier]['processed'] = True
                elif dataset == 'testing':
                    testing_pages[identifier]['processed'] = True
                elif dataset == 'validation':
                    validation_pages[identifier]['processed'] = True

                processed_pages[dataset].add(identifier)
                pbar.update(1)

            except orjson.JSONDecodeError:
                pbar.update(1)
                continue

    return processed_pages


if __name__ == "__main__":
    base_dir = "./data"
    output_dir = Path(base_dir) / "processed_output"
    output_dir.mkdir(parents=True, exist_ok=True)

    links_df = pd.read_csv(output_dir / "all_links_summary.csv")
    url_to_title = dict(zip(links_df['url'], links_df['name']))

    output_paths = {
        'training': output_dir / "training_processed_data.csv",
        'testing': output_dir / "testing_processed_data.csv",
        'validation': output_dir / "validation_processed_data.csv"
    }

    columns = [
        "identifier", "title", "url", "section_text",
        "unique_links", "text_length",
        "links_to_text_ratio", "section_link_count", "section_path"
    ]

    for path in output_paths.values():
        if not path.exists():
            pd.DataFrame(columns=columns).to_csv(path, index=False)

    first_flags = {k: not output_paths[k].exists() for k in output_paths}

    def load_page_details(path):
        if not path.exists():
            return {}

        df = pd.read_csv(path)
        page_details_dict = {}
        for _, row in df.iterrows():
            page_details_dict[row['identifier']] = {
                'processed': False
            }
        return page_details_dict

    training_pages = load_page_details(Path('training_dataset_page_details.csv'))
    testing_pages = load_page_details(Path('testing_dataset_page_details.csv'))
    validation_pages = load_page_details(Path('validation.csv'))

    all_processed = {'training': set(), 'testing': set(), 'validation': set()}

    for i in tqdm(range(38), desc="Processing files"):
        input_file = Path(base_dir) / f"enwiki_namespace_0_{i}.jsonl"
        if not input_file.exists():
            tqdm.write(f"Skipping {input_file.name}")
            continue
        tqdm.write(f"Processing {input_file.name}")
        processed = process_file(input_file, training_pages, testing_pages, validation_pages,
                                 output_paths, first_flags, url_to_title)
        for k in processed:
            all_processed[k].update(processed[k])

    def save_page_details(path, pages):
```

```
        if not path.exists():
            return

        df = pd.read_csv(path)
        df['processed'] = df['identifier'].isin(pages.keys())
        df.to_csv(path, index=False)

    save_page_details(Path('training_dataset_page_details.csv'), training_pages)
    save_page_details(Path('testing_dataset_page_details.csv'), testing_pages)
    save_page_details(Path('validation.csv'), validation_pages)

    tqdm.write("All files processed.")
    for key, path in output_paths.items():
        tqdm.write(f" {key.capitalize()} data saved to: {path}")
    for dataset in all_processed:
        tqdm.write(f" {dataset.capitalize()} - Processed {len(all_processed[dataset])} new pages")
```

```
Reading enwiki_namespace_0_36.jsonl: 68737lines [00:17, 4071.14lines/s]
Reading enwiki_namespace_0_36.jsonl: 69206lines [00:17, 4227.83lines/s]
Reading enwiki_namespace_0_36.jsonl: 69645lines [00:17, 4195.10lines/s]
Reading enwiki_namespace_0_36.jsonl: 70246lines [00:17, 4692.76lines/s]
Reading enwiki_namespace_0_36.jsonl: 70728lines [00:17, 2075.47lines/s]
Reading enwiki_namespace_0_36.jsonl: 71304lines [00:18, 2628.13lines/s]
Reading enwiki_namespace_0_36.jsonl: 71731lines [00:18, 2882.17lines/s]
Reading enwiki_namespace_0_36.jsonl: 72223lines [00:18, 3280.23lines/s]
Reading enwiki_namespace_0_36.jsonl: 72662lines [00:18, 3407.44lines/s]
Reading enwiki_namespace_0_36.jsonl: 73093lines [00:18, 3612.63lines/s]
Reading enwiki_namespace_0_36.jsonl: 73517lines [00:18, 2974.12lines/s]
Reading enwiki_namespace_0_36.jsonl: 74152lines [00:18, 3716.43lines/s]
Reading enwiki_namespace_0_36.jsonl: 74701lines [00:18, 4139.61lines/s]
Reading enwiki_namespace_0_36.jsonl: 75175lines [00:18, 4274.70lines/s]
Reading enwiki_namespace_0_36.jsonl: 75647lines [00:19, 4363.33lines/s]
Reading enwiki_namespace_0_36.jsonl: 76115lines [00:19, 4296.15lines/s]
Reading enwiki_namespace_0_36.jsonl: 76637lines [00:19, 4548.54lines/s]
Reading enwiki_namespace_0_36.jsonl: 77110lines [00:19, 4530.21lines/s]
Reading enwiki_namespace_0_36.jsonl: 77576lines [00:19, 4557.17lines/s]
Reading enwiki_namespace_0_36.jsonl: 78057lines [00:19, 4627.81lines/s]
Reading enwiki_namespace_0_36.jsonl: 78588lines [00:19, 4824.53lines/s]
Reading enwiki_namespace_0_36.jsonl: 79076lines [00:19, 4822.68lines/s]
Reading enwiki_namespace_0_36.jsonl: 79562lines [00:19, 4765.64lines/s]
Reading enwiki_namespace_0_36.jsonl: 80042lines [00:19, 4621.55lines/s]
Reading enwiki_namespace_0_36.jsonl: 80534lines [00:20, 4640.30lines/s]
Reading enwiki_namespace_0_36.jsonl: 81000lines [00:20, 4422.21lines/s]
Reading enwiki_namespace_0_36.jsonl: 81446lines [00:20, 4328.57lines/s]
Reading enwiki_namespace_0_36.jsonl: 81960lines [00:20, 4556.49lines/s]
Reading enwiki_namespace_0_36.jsonl: 82419lines [00:20, 4260.64lines/s]
Reading enwiki_namespace_0_36.jsonl: 82860lines [00:20, 4296.13lines/s]
Reading enwiki_namespace_0_36.jsonl: 83294lines [00:21, 1891.54lines/s]
Reading enwiki_namespace_0_36.jsonl: 83779lines [00:21, 2337.73lines/s]
Reading enwiki_namespace_0_36.jsonl: 84322lines [00:21, 2885.56lines/s]
Reading enwiki_namespace_0_36.jsonl: 84932lines [00:21, 3534.90lines/s]
Reading enwiki_namespace_0_36.jsonl: 85413lines [00:21, 3797.44lines/s]
Reading enwiki_namespace_0_36.jsonl: 85890lines [00:21, 4003.74lines/s]
Reading enwiki_namespace_0_36.jsonl: 86363lines [00:21, 3928.71lines/s]
Reading enwiki_namespace_0_36.jsonl: 86965lines [00:21, 3959.82lines/s]
Processing files:  97%|██████████| 37/38 [19:37<00:26, 26.60s/it]🔄 Processing enwiki_namespace_0_37.jsonl

Reading enwiki_namespace_0_37.jsonl: 0lines [00:00, ?lines/s]
Reading enwiki_namespace_0_37.jsonl: 518lines [00:00, 5171.55lines/s]
Reading enwiki_namespace_0_37.jsonl: 1036lines [00:00, 4220.47lines/s]
Reading enwiki_namespace_0_37.jsonl: 1554lines [00:00, 4606.20lines/s]
Reading enwiki_namespace_0_37.jsonl: 2026lines [00:00, 4588.40lines/s]
Reading enwiki_namespace_0_37.jsonl: 2503lines [00:00, 4640.36lines/s]
Reading enwiki_namespace_0_37.jsonl: 2972lines [00:00, 4056.80lines/s]
Reading enwiki_namespace_0_37.jsonl: 3391lines [00:00, 4065.96lines/s]
Reading enwiki_namespace_0_37.jsonl: 3807lines [00:00, 3895.47lines/s]
Reading enwiki_namespace_0_37.jsonl: 4531lines [00:01, 4179.52lines/s]
Processing files: 100%|██████████| 38/38 [19:38<00:00, 31.02s/it]
All files processed.
📂 Training data saved to: data/processed_output/training_processed_data.csv
📂 Testing data saved to: data/processed_output/testing_processed_data.csv
📂 Validation data saved to: data/processed_output/validation_processed_data.csv
📊 Training - Processed 100000 new pages
📊 Testing - Processed 1000 new pages
📊 Validation - Processed 1000 new pages
```

```
def safe_literal_eval(val):
    if pd.isna(val) or not isinstance(val, str):
        return []
    try:
        return ast.literal_eval(val)
    except (ValueError, SyntaxError):

        try:
            return [item.strip() for item in val.split(',') if item.strip()]
        except:
            return []
```

```
train_df = pd.read_csv('/content/data/processed_output/training_processed_data.csv',converters={'unique_links': safe_literal
train_df.head()
```

	identifier	title	url	section_text	unique_links	text_l
0	40477619	2013 Tashkent Open – Doubles	https://en.wikipedia.org/wiki/2013_Tashkent_Op...	Paula Kania and Polina Pekhova were the defend...	[Yaroslava Shvedova, 23px-Flag_of_Luxembourg.s...	
1	53572039	2017 İstanbul Cup	https://en.wikipedia.org/wiki/2017_%C4%B0stanb...	The 2017 İstanbul Cup (also known as the TEB B...	[16px-Flag_of_Switzerland_%28Pantone%29.svg.pn...	
2	53860077	2017 İstanbul Cup – Doubles	https://en.wikipedia.org/wiki/2017_%C4%B0stanb...	Andreea Mitu and İpek Soylu were the defending...	[23px-Flag_of_Chinese_Taipei_for_Olympic_games...	
3	32099256	Bill Winfrey	https://en.wikipedia.org/wiki/Bill_Winfrey	William Colin Winfrey (May 9, 1916 – April 14,...	[Palos_Verdes_Handicap, Detroit, Prioress Stak...	
4	14667413	Glycoside hydrolase family 1	https://en.wikipedia.org/wiki/Glycoside_hydrol...	Glycoside hydrolase family 1 is a family of gl...	[3.2.1.86, IPR001360, 3.2.1.21, GetPfamStr.pl?...	

```
test_df = pd.read_csv('/content/data/processed_output/testing_processed_data.csv',converters={'unique_links': safe_literal_e
test_df.head()
```

	identifier	title	url	section_text	unique_links	text_length	links_
0	38099978	2013 Blossom Cup	https://en.wikipedia.org/wiki/2013_Blossom_Cup	The 2013 Blossom Cup was a professional tennis...	[Lu Jingjing, Nadiya_Kichenok, Tennis, Liu Fan...	111	
1	10097677	3C-BZ	https://en.wikipedia.org/wiki/3C-BZ	3C-BZ (4-benzyloxy-3,5-dimethoxyamphetamine) i...	[7px-X_mark.svg.png, index.php?title=Special:C...	169	
2	47774306	2005–06 AS Monaco FC season	https://en.wikipedia.org/wiki/2005%E2%80%9306_...	The 2005–06 season was AS Monaco FC 's 49th se...	[Kit (association football), Didier Deschamps,...	140	
3	18839978	Allium howellii	https://en.wikipedia.org/wiki/Allium_howellii	Allium howellii is a North American species of...	[Santa Barbara County, California, San_Bernard...	148	
4	45128029	Commitment (Harold Vick album)	https://en.wikipedia.org/wiki/Commitment_(Haro...	Commitment is an album led by American saxopho...	[Double bass, Flute, Soprano saxophone, Drum k...	111	

Next steps: Generate code with `test_df` View recommended plots New interactive sheet

```
val_df = pd.read_csv('/content/data/processed_output/validation_processed_data.csv',converters={'unique_links': safe_literal
val_df.head()
```

	identifier	title	url	section_text	unique_links	text_length	links_to_text_rat
0	27216689	Coding theory approaches to nucleic acid design	https://en.wikipedia.org/wiki/Coding_theory_ap...	DNA code construction refers to the applicatio...	[Nucleic acid double helix, Parallel computing...	355	0.
1	27216689	Coding theory approaches to nucleic acid design	https://en.wikipedia.org/wiki/Coding_theory_ap...	Novel constructions of such codes include usin...	[Pyrimidine, Thymine, Adenine, Mutations, Puri...	394	0.
2	27216689	Coding theory approaches to nucleic acid design	https://en.wikipedia.org/wiki/Coding_theory_ap...	For any pair of length- n {\displaystyle {\mat...	[GC_content, Hamming distance]	214	0.
3	27216689	Coding theory approaches to nucleic acid design	https://en.wikipedia.org/wiki/Coding_theory_ap...	A generalized Hadamard matrix H ≡ H (n, C m) {...	[Finite field, Hadamard matrix]	402	0.

Next steps: Generate code with val_df | View recommended plots | New interactive sheet

```
links = pd.read_csv("./links.csv")
```

```
links
```

	url	count	title
0	https://en.wikipedia.org/wiki/Vesna_Dolonc	99	Vesna Dolonc
1	https://en.wikipedia.org/wiki/Tímea_Babos	699	Tímea_Babos
2	https://en.wikipedia.org/wiki/Glossary_of_tenn...	16328	Glossary_of_tennis_terms
3	https://en.wikipedia.org/wiki/Olga_Govortsova	326	Olga Govortsova
4	https://en.wikipedia.org/wiki/Hungary	38478	Hungary
...	...	...	...
554295	https://en.wikipedia.org/wiki/Graciela_Maturo?...	50	Graciela_Maturo
554296	https://en.wikipedia.org/wiki/2024_Big_Machine...	53	2024 Big Machine Music City Grand Prix
554297	https://en.wikipedia.org/wiki/Trump_Internatio...	142	Trump_International_Golf_Club_shooting
554298	https://en.wikipedia.org/wiki/Trump_Internatio...	120	Trump_International_Golf_Club_shooting
554299	https://en.wikipedia.org/wiki/AK-47-style_rifle	102	AK-47-style_rifle

554300 rows × 3 columns

```
valid_links = set(links['title'])

def filter_links(unique_links):
    return [link for link in unique_links if link in valid_links]

train_df['unique_links'] = train_df['unique_links'].apply(filter_links)
test_df['unique_links'] = test_df['unique_links'].apply(filter_links)
val_df['unique_links'] = val_df['unique_links'].apply(filter_links)
```

```
train_df.to_csv('/content/data/processed_output/training_processed_data.csv', index=False)
test_df.to_csv('/content/data/processed_output/testing_processed_data.csv', index=False)
val_df.to_csv('/content/data/processed_output/validation_processed_data.csv', index=False)
```

Model

```
!pip install numpy pandas scipy scikit-learn joblib
```

```
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (2.0.2)
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (2.2.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.11/dist-packages (1.15.2)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/dist-packages (1.6.1)
Requirement already satisfied: joblib in /usr/local/lib/python3.11/dist-packages (1.4.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.9.0.po
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
```

```
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (3.6.
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas)
```

```
train_path = '/content/data/processed_output/training_processed_data.csv'
test_path = '/content/data/processed_output/testing_processed_data.csv'
val_path = '/content/data/processed_output/validation_processed_data.csv'
TEXT_COL = "section_text"
LABELS_COL = "unique_links"
OUTPUT_DIR = "data_splits"
```

```
def load_and_clean_data(filepath, text_col, labels_col):
    if not os.path.exists(filepath):
        raise FileNotFoundError(f"Data file {filepath} not found")

    try:
        df = pd.read_csv(filepath, usecols=[text_col, labels_col])
    except Exception as e:
        print(f"Data loading failed: {str(e)}")
        raise

    def remove_accents(text):
      nfkd_form = unicodedata.normalize('NFKD', text)
      return ''.join([c for c in nfkd_form if not unicodedata.combining(c)])

    df[text_col] = (
        df[text_col]
        .astype(str)
        .str.lower()
        .str.replace(r'[^\w\s]', ' ', regex=True)
        .str.replace(r'\s+', ' ', regex=True)
        .str.strip()
        .apply(remove_accents)
    )


    def parse_labels(val):
        if pd.isna(val):
          return []
        try:
            return ast.literal_eval(val)
        except (ValueError, SyntaxError):
            return []

    df[labels_col] = df[labels_col].apply(parse_labels)

    print(f"Loaded {len(df)} samples")
    return df
```

```
train_df = load_and_clean_data(train_path, TEXT_COL, LABELS_COL)
train_df.head(2)
train_df.to_csv('/content/train_df.csv', index=False)
```

```
Loaded 105995 samples
```

```
test_df = load_and_clean_data(test_path, TEXT_COL, LABELS_COL)
test_df.head(2)
test_df.to_csv('test_df.csv', index=False)
```

```
Loaded 1088 samples
```

```
val_df = load_and_clean_data(val_path, TEXT_COL, LABELS_COL)
val_df.head(2)
val_df.to_csv('val_df.csv', index=False)
```

```
Loaded 23048 samples
```

```
links = pd.read_csv("./links.csv")
```

cleaning

```
flattened_links = train_df['unique_links'].explode()
link_counts = flattened_links.value_counts().reset_index()
link_counts.columns = ['link', 'count']
```

```
link_counts = link_counts[link_counts['link'].isin(links['title'].tolist())].copy()
```

```
link_counts.head()
```

	link	count
0	Taxonomy (biology)	17264
2	United States	13675
3	Eukaryote	12874
4	Animal	12697
6	France	12169

```
len(link_counts)
```

```
256368
```

```
link_counts['encoded'] = pd.factorize(link_counts['link'])[0]
```

```
link_counts['link'].to_csv('output-labels.txt', index=False, header=False)
```

```
valid_links = set(link_counts['link'])

def filter_links(unique_links):
    return [link for link in unique_links if link in valid_links]

train_df['unique_links'] = train_df['unique_links'].apply(filter_links)
test_df['unique_links'] = test_df['unique_links'].apply(filter_links)
val_df['unique_links'] = val_df['unique_links'].apply(filter_links)
```

```
link_mapping = dict(zip(link_counts['link'], link_counts['encoded']))

def encode_links(link_list):
    return [link_mapping.get(link, -1) for link in link_list]

train_df['unique_links'] = train_df['unique_links'].apply(encode_links)
test_df['unique_links'] = test_df['unique_links'].apply(encode_links)
val_df['unique_links'] = val_df['unique_links'].apply(encode_links)
```

```
train_df.head()
```

	section_text	unique_links
0	paula kania and polina pekhova were the defend...	[2435, 7245, 141, 126, 3076, 53, 6477, 49, 70,...
1	the 2017 i stanbul cup also known as the teb b...	[9, 4096, 5216, 1356, 2437, 58, 6323, 1, 3791,...
2	andreea mitu and i pek soylu were the defendin...	[6323, 1, 1267, 17, 3742, 14, 131, 4039, 1693,...
3	william colin winfrey may 9 1916 april 14 1994...	[6147, 32696, 5296, 19665, 10484, 6491, 143560...
4	glycoside hydrolase family 1 is a family of gl...	[4931, 13014, 48084, 1420, 11189, 14771, 12992...

```
test_df.head()
```

	section_text	unique_links
0	the 2013 blossom cup was a professional tennis...	[9813, 58, 13148, 17, 3895, 150, 32568, 36, 49...
1	3c bz 4 benzyloxy 3 5 dimethoxyamphetamine is ...	[75, 251, 74, 12365, 114, 103, 198, 29402, 522...
2	the 2005 06 season was as monaco fc s 49th sea...	[111, 50845, 70142, 71907, 12517, 23993, 3684,...
3	allium howellii is a north american species of...	[11659, 23044, 11274, 24556, 95, 14452, 0, 979...
4	commitment is an album led by american saxopho...	[1448, 3808, 8520, 1404, 34736, 61209, 8998, 2...

Next steps: Generate code with test_df | View recommended plots | New interactive sheet

```
val_df.head()
```

	section_text	unique_links
0	dna code construction refers to the applicatio...	[239246, 60516, 212828, 3185, 88807, 5301, 235...
1	novel constructions of such codes include usin...	[15432, 17066, 12069, 18275, 23501, 14593, 110...
2	for any pair of length n displaystyle mathit n...	[205007]
3	a generalized hadamard matrix h h n c m displa...	[]
4	also the rows of such an exponent matrix satis...	[]

Next steps: Generate code with val_df | View recommended plots | New interactive sheet

```
def save_to_txt(df, filename):
    with open(f'{filename}', 'w', encoding='utf-8') as f:
        for _, row in df.iterrows():
            if not row['unique_links']:
                continue
            label_ids = ",".join(map(str, row['unique_links']))
            text = row['section_text'].strip().replace('\n', ' ')
            f.write(f"{label_ids}\t{text}\n")
```

```
save_to_txt(train_df, 'training-data.txt')
save_to_txt(test_df, 'testing-data.txt')
save_to_txt(val_df, 'validation-data.txt')
```

```
!python3 -m pip install libpecos
```

```
Successfully installed datasets-3.5.1 dill-0.3.8 fsspec-2025.3.0 libpecos-1.2.8 multiprocess-0.70.16 numpy-1.26.4 pyic
```

```
!pip install scipy==1.9.3
```

```
Collecting scipy==1.9.3
  Downloading scipy-1.9.3-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (58 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 58.4/58.4 kB 3.2 MB/s eta 0:00:00
Collecting numpy<1.26.0,>=1.18.5 (from scipy==1.9.3)
  Downloading numpy-1.25.2-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (5.6 kB)
Downloading scipy-1.9.3-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (33.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 33.4/33.4 MB 57.7 MB/s eta 0:00:00
Downloading numpy-1.25.2-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (18.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 18.2/18.2 MB 65.0 MB/s eta 0:00:00
Installing collected packages: numpy, scipy
  Attempting uninstall: numpy
    Found existing installation: numpy 1.26.4
    Uninstalling numpy-1.26.4:
      Successfully uninstalled numpy-1.26.4
  Attempting uninstall: scipy
    Found existing installation: scipy 1.13.1
    Uninstalling scipy-1.13.1:
      Successfully uninstalled scipy-1.13.1
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviou
cvxpy 1.6.5 requires scipy>=1.11.0, but you have scipy 1.9.3 which is incompatible.
imbalanced-learn 0.13.0 requires scipy<2,>=1.10.1, but you have scipy 1.9.3 which is incompatible.
jaxlib 0.5.1 requires scipy>=1.11.1, but you have scipy 1.9.3 which is incompatible.
scikit-image 0.25.2 requires scipy>=1.11.4, but you have scipy 1.9.3 which is incompatible.
tensorflow 2.18.0 requires numpy<2.1.0,>=1.26.0, but you have numpy 1.25.2 which is incompatible.
thinc 8.3.6 requires numpy<3.0.0,>=2.0.0, but you have numpy 1.25.2 which is incompatible.
blosc2 3.3.2 requires numpy>=1.26, but you have numpy 1.25.2 which is incompatible.
albumentations 2.0.6 requires scipy>=1.10.0, but you have scipy 1.9.3 which is incompatible.
jax 0.5.2 requires scipy>=1.11.1, but you have scipy 1.9.3 which is incompatible.
Successfully installed numpy-1.25.2 scipy-1.9.3
WARNING: The following packages were previously imported in this runtime:
  [numpy]
You must restart the runtime in order to use newly installed versions.
```

RESTART SESSION

Modified PECOS model that was copied from their GitHub

```
class CustomPECOS:
    def __init__(self, preprocessor=None, xlinear_model=None, output_items=None):
        self.preprocessor = preprocessor
        self.xlinear_model = xlinear_model
        self.output_items = output_items

    @classmethod
    def train(cls, input_text_path, output_text_path):
        """Train a CustomPECOS model

        Args:
            input_text_path (str): Text input file name.
            output_text_path (str): The file path for output text items.
            vectorizer_config (str): Json_format string for vectorizer config (default None). e.g. {"type": "tfidf", "kwargs

        Returns:
            A CustomPECOS object
        """
        parsed_result = Preprocessor.load_data_from_file(input_text_path, output_text_path)
        Y = parsed_result["label_matrix"]
        corpus = parsed_result["corpus"]
        vectorizer_config = {
          "type": "tfidf",
          "kwargs": {
              "min_df": 40,
              "max_features": 100000,
              "dtype": "float32",
              "stop_words": "english",
              "base_vect_configs": [
                  {
                      "ngram_range": [1, 2],
                      "max_df_ratio": 0.90,
                      "analyzer": "word",
                      "sublinear_tf": True,
                      "smooth_idf": True,
                      "norm": "l2"
                  }
              ]
```

```
        }
    }


        preprocessor = Preprocessor.train(corpus, vectorizer_config)
        X = preprocessor.predict(corpus)

        label_feat = LabelEmbeddingFactory.create(Y, X, method="pifa")

        cluster_chain = Indexer.gen(label_feat, nr_splits=8)

        xlinear_model = XLinearModel.train(X, Y, C=cluster_chain,
                                           negative_sampling_scheme="tfn",
                                           threshold=0.1,verbose=1,threads=-1)

        with open(output_text_path, "r", encoding="utf-8") as f:
            output_items = [q.strip() for q in f]

        return cls(preprocessor, xlinear_model, output_items)

    def predict(self, corpus):
        """Predict labels for given inputs

        Args:
            corpus (list of strings): input strings.
        Returns:
            csr_matrix: predicted label matrix (num_samples x num_labels)
        """
        X = self.preprocessor.predict(corpus)
        Y_pred = self.xlinear_model.predict(X)
        return smat_util.sorted_csr(Y_pred)

    def save(self, model_folder):
        """Save the CustomPECOS model

        Args:
            model_folder (str): folder name to save
        """
        self.preprocessor.save(f"{model_folder}/preprocessor")
        self.xlinear_model.save(f"{model_folder}/xlinear_model")
        with open(f"{model_folder}/output_items.json", "w", encoding="utf-8") as fp:
            json.dump(self.output_items, fp)

    @classmethod
    def load(cls, model_folder):
        """Load the CustomPECOS model

        Args:
            model_folder (str): folder name to load
        Returns:
            CustomPECOS
        """
        preprocessor = Preprocessor.load(f"{model_folder}/preprocessor")
        xlinear_model = XLinearModel.load(f"{model_folder}/xlinear_model")
        with open(f"{model_folder}/output_items.json", "r", encoding="utf-8") as fin:
            output_items = json.load(fin)
        return cls(preprocessor, xlinear_model, output_items)
```

```
input_text_path = "/content/training-data.txt"
output_text_path = "/content/output-labels.txt"
```

```
model_moreGrams = CustomPECOS.train(input_text_path, output_text_path)
```

```
!mkdir model_moreGrams
```

```
model_moreGrams_path = "./model_moreGrams/pecos-CustomPECOS-model"
```

```
model_moreGrams.save(model_moreGrams_path)
```

Prediction

Test Dataset

```
testing_text_path = "/content/testing-data.txt"
```

```
parsed_result = model_moreGrams.preprocessor.load_data_from_file(testing_text_path, output_text_path)
Y_tst = parsed_result["label_matrix"]
corpus = parsed_result["corpus"]
```

```
X = model_moreGrams.preprocessor.predict(corpus)
```

```
def batch_predict(model, X, batch_size=100):
    num_samples = X.shape[0]
    Y_pred = []

    for start in tqdm(range(0, num_samples, batch_size), desc="Predicting", ncols=100):
        end = min(start + batch_size, num_samples)
        batch = X[start:end]

        Y_batch_pred = model.xlinear_model.predict(batch)

        Y_pred.append(Y_batch_pred)

    return np.vstack(Y_pred)
```

```
Y_pred = batch_predict(model_moreGrams, X, batch_size=500)
Y_pred = vstack(Y_pred.ravel())
```

```
Predicting: 100%|██████████████████████████████████████| 3/3 [00:02<00:00,  1.00it/s]
```

```
metrics = smat_util.Metrics.generate(Y_tst, Y_pred, topk=10)
print(metrics)
```

```
prec   = 86.65 84.35 82.11 80.13 78.18 76.58 74.69 72.86 71.08 69.27
recall = 5.32 10.11 14.47 18.55 22.32 25.92 29.14 32.10 34.84 37.27
```

```
def set_css():
  display(HTML('''
  <style>
    pre {
        white-space: pre-wrap;
    }
  </style>
  '''))
get_ipython().events.register('pre_run_cell', set_css)
```

```
num_samples = min(10, len(corpus))
random_indices = random.sample(range(len(corpus)), num_samples)

for i in random_indices:
    text = corpus[i]
    print(f"Text Input: {text}")
    for j in range(Y_pred.indptr[i], Y_pred.indptr[i + 1]):
        pred_label = model_moreGrams.output_items[Y_pred.indices[j]]
        pred_score = Y_pred.data[j]
        print(f"Score {pred_score:.4f}: {pred_label}")
    print("-" * 40)
```

```
county in 2011 the population of zejane was 130 the village is 18 km north west of matulji near the municipality road
leading from vele mune and male mune to opatija and rijeka in a karst valley between two mountain ridges the village
is known for the cici istro romanians who settled here in the late 15th or early 16th century from 1510 until 1525
when the villages vele mune male mune and zejane were settled by krsto frankopan
Score 0.9998: Time zone
Score 0.9998: Daylight saving time
Score 0.9414: Central European Summer Time
Score 0.9315: UTC+2
Score 0.9194: Central European Time
Score 0.7868: UTC+1
Score 0.4982: Croatia
Score 0.2810: List of sovereign states
Score 0.1846: Eastern European Summer Time
Score 0.1775: Telephone numbering plan
Score 0.1772: Eastern European Time
Score 0.1004: Village
Score 0.0540: Vehicle registration plate
Score 0.0537: Countries_of_the_world
Score 0.0402: UTC+01:00
Score 0.0374: UTC+02:00
Score 0.0361: Hungarian language
Score 0.0287: Counties of Croatia
Score 0.0264: Fire services in the United Kingdom
Score 0.0264: "List of law enforcement agencies in the United Kingdom, Crown Dependencies and British Overseas
Territories"
----------------------------------------
Text Input: the gumbo darter etheostoma thompsoni is a species of freshwater ray finned fish a darter from the
subfamily etheostomatinae part of the family percidae which also contains the perches ruffes and pikeperches it is
found in the neches sabine and calcasieu river drainages in southeastern texas and southwestern louisiana they
inhabit riverbanks where there are exposed roots with accumulated vegetational debris and sand to mixed sand and
gravel substrate with very little silt this species can reach a length of 5 4 cm 2 1 in the gumbo darter was first
formally described in 2012 by royal dallas suttkus henry l bart jr and david a etnier with the type locality given as
the neches river just below the town bluff dam at town bluff tyler county texas the specific name honors the
biologist dr bruce allen thompson 1946 2007
Score 1.0000: Binomial nomenclature
Score 1.0000: Chordate
Score 1.0000: Taxonomy (biology)
Score 1.0000: Eukaryote
Score 1.0000: Animal
Score 0.9998: Family (biology)
Score 0.9995: Actinopterygii
Score 0.9963: Ray-finned_fish
Score 0.9822: Subfamily
Score 0.9720: Perciformes
Score 0.9232: Percidae
Score 0.9019: Perch
Score 0.9015: IUCN Red List
Score 0.8829: Etheostomatinae
Score 0.8806: Gymnocephalus
Score 0.8806: Sander (fish)
Score 0.8579: Conservation_status
Score 0.8232: Specific name (zoology)
Score 0.7886: Type_locality_(biology)
Score 0.7681: Species description
----------------------------------------
Text Input: hiroyuki kinoshita 木下 浩之 kinoshita hiroyuki born october 23 1958 is a japanese actor and voice actor
he was born in saitama sennen no koi story of genji 2001 tokugawa yoshinobu 1998 prince kuni asahiko aoi tokugawa
sandai 2000 emperor go yozei mito komon 2001 dr kogoro matsumiya komyo ga tsuji 2006 shimizu muneharu aibou tokyo
detective duo 2007 ghost in the shell stand alone complex 2002 yamaguchi detective conan 2005 akira sakuma detective
conan 2006 korn bakugan battle brawlers 2007 exedra bokurano ours 2007 sasami detective conan 2007 kakuji dejima
blassreiter 2008 matthew grant bakugan battle brawlers new vestroia 2010 exedra house of five leaves 2010 yagi
heizaemon kindaichi case files r 2014 wang long level e 2011 kyushiro yumeno robotics notes 2012 hiromu hidaka tari
tari 2012 shoichi okita one piece 2013 rock aldnoah zero 2014 volf areash chaika the coffin princess 2014 simon
scania glasslip 2014 ken fukami soul eater not 2014 cafe master go princess precure 2015 tsukasa kaido ajin demi
human 2016 ikuya ogura knight s magic 2017 megalobox 2018 fujimaki kingdom season 3 2021 orudo utsunomiko 1989 kusuri
jin roh the wolf brigade 2000 atsushi henmi detective conan the raven chaser 2009 korn ajin part 1 shodo 2015 ikuya
ogura ace combat zero the belkan war 2006 joshua lucan bristow tales of xillia 2011 jilland borderlands 2 2012
japanese version handsome jack tales of xillia 2 2012 jilland the evil within 2014 japanese version detective
sebastian castellanos nioh 2017 edward kelley starlink battle for atlas 2019 st grand fate grand order 2020 zeus
famicom detective club the missing heir 2021 remake 2021 kanji ayashiro aaron eckhart the dark knight harvey dent
battle los angeles michael nantz erased ben logan sully 2020 the cinema edition jeff skiles wander arthur bretnik the
first lady gerald ford the dark knight harvey dent battle los angeles michael nantz erased ben logan sully 2020 the
cinema edition jeff skiles wander arthur bretnik the first lady gerald ford colin firth bridget jones s diary mark
darcy bridget jones the edge of reason mark darcy nanny mcphee cedric brown mamma mia harry bright mamma mia here we
go again harry bright bridget jones s diary mark darcy bridget jones the edge of reason mark darcy nanny mcphee
cedric brown mamma mia harry bright mamma mia here we go again harry bright the 4400 jordan collier billy campbell
5x2 gilles stephane freiss ambulance fbi agent anson clark keir o donnell american beauty 2003 tbs edition lester
burnham kevin spacey american gods mr world crispin glover antarctic journal lee young min park hee soon arbitrage
det bryer tim roth the a team vance burress agent lynch patrick wilson avalon murphy jerzy gudejko avengers age of
ultron ultron james spader
Score 0.1823: Japan
Score 0.1348: Los Angeles
Score 0.0356: Australia
```

```
Score 0.0154: United Kingdom
Score 0.0068: Canada
Score 0.0063: 576i
Score 0.0044: Netherlands
Score 0.0044: Rock music
Score 0.0037: Anime
Score 0.0034: 16:9
Score 0.0028: HDTV
Score 0.0028: 1080i
Score 0.0027: SDTV
Score 0.0025: South Korea
Score 0.0024: New Zealand
Score 0.0014: Actor
Score 0.0012: Mexico
Score 0.0012: NBC
Score 0.0010: Austria
Score 0.0009: Brazil
----------------------------------------
Text Input: cynips is a genus of gall wasps in the tribe cynipini the oak gall wasps one of the best known is the
common oak gall wasp cynips quercusfolii which induces characteristic spherical galls about two centimeters wide on
the undersides of oak leaves as of 2008 there are about 39 species in this genus some authors have included antron in
cynips but it was recently resurrected as a distinct genus cynips agama cynips caputmedusae cynips conspicua fuzzy
gall wasp cynips cornifex cynips disticha cynips divisa red pea gall cynips douglasii spined turbaned gall wasp
cynips fusca cynips izzetbaysali cynips longiventris cynips mirabilis speckled gall wasp cynips multipunctata gray
midrib gall wasp cynips quercusechinus urchin gall wasp cynips quercusfolii cynips schlechtendali the wasp formerly
named cynips saltatorius is now named neuroterus saltatorius
Score 1.0000: Taxonomy (biology)
Score 0.9504: Eukaryote
Score 0.9251: Animal
Score 0.8730: Arthropod
Score 0.8720: Insect
Score 0.8550: Hymenoptera
Score 0.5383: Genus
Score 0.2736: Synonym (taxonomy)
Score 0.2318: Type species
Score 0.0895: Gall
Score 0.0195: Chordate
Score 0.0173: Plant
Score 0.0156: Vascular plant
Score 0.0143: Species
Score 0.0130: Tribe (biology)
Score 0.0124: Parasitoid
Score 0.0116: Flowering plant
Score 0.0113: Gall wasp
Score 0.0078: Eudicots
Score 0.0067: Family (biology)
----------------------------------------
Text Input: oleru is a village in bapatla district of the indian state of andhra pradesh it is the located in
bhattiprolu mandal of tenali revenue division it forms a part of andhra pradesh capital region it is situated near
krishna river in the coastal andhra region of the state oleru is situated to the southeast of the mandal headquarters
bhattiprolu at 16 29 32 n 80 00 32 e 16 49222 n 80 00889 e it is spread over an area of 13 78 ha 34 1 acres oleru
gram panchayat is the local self government of the village it is divided into wards and each ward is represented by a
ward member as per the school information report for the academic year 2018 19 the village has a total of 6 schools
these schools include 2 private and 4 mandal parishad schools national highway 216 passes through the village
Score 1.0000: Vehicle registration plate
Score 1.0000: Indian Standard Time
Score 1.0000: UTC+5:30
Score 1.0000: List_of_districts_of_India
Score 1.0000: India
Score 1.0000: Postal Index Number
Score 1.0000: Time zone
Score 1.0000: Mandal
Score 0.9999: Telugu language
Score 0.9999: Andhra Pradesh
Score 0.9968: Gram panchayat
Score 0.9947: Telephone numbering plan
Score 0.9930: Panchayati_raj_(India)
Score 0.9899: States and union territories of India
Score 0.9897: Local_self-government_in_India
Score 0.9859: Indian_state
Score 0.4780: Guntur district
Score 0.4087: District_Councils_of_India
Score 0.2945: Mandal Parishad Primary School
Score 0.2509: Mandal_Parishad
----------------------------------------
Text Input: pokrovsky russian покровский is a rural locality a khutor in vyshnereutchansky selsoviet rural settlement
medvensky district kursk oblast russia population 9 2010 russian census 16 2002 census the khutor is located on the
lyubach river a left tributary of the reut river in the seym basin 50 km 31 mi from the russia ukraine border 41 km
25 mi south west of kursk 14 5 km 9 0 mi south west of the district center the urban type settlement medvenka 8 km 5
0 mi from the selsoviet center verkhny reutets pokrovsky has a warm summer humid continental climate dfb in the
koppen climate classification pokrovsky is located 17 5 km 10 9 mi from the federal route m 2 crimea highway a part
of the european route e105 on the road of intermunicipal significance 38h 185 m2 crimea highway gakhovo 29 5 km 18 3
mi from the nearest railway halt 439 km railway line lgov i kursk the rural locality is situated 50 km 31 mi from
kursk vostochny airport 93 km 58 mi from belgorod international airport and 237 km 147 mi from voronezh peter the
great airport
Score 1.0000: Postal codes in Russia
```

```
Score 0.9998: Russia
Score 0.9993: OKTMO
Score 0.9993: Federal subjects of Russia
Score 0.9990: Districts of Russia
Score 0.9988: Moscow Time
Score 0.9987: Time zone
Score 0.9980: Telephone numbers in Russia
Score 0.9979: 2002 Russian census
Score 0.9976: Selsoviet
Score 0.9973: Kursk
Score 0.9973: Types_of_inhabited_localities_in_Russia
Score 0.9969: 2010 Russian census
Score 0.9965: Russia–Ukraine_border
Score 0.9962: Kursk Oblast
Score 0.9940: Humid continental climate
Score 0.9933: UTC+3
Score 0.9932: Köppen_climate_classification
Score 0.9915: Belgorod International Airport
Score 0.9915: Voronezh International Airport
----------------------------------------
Text Input: howard white smiley johnson september 22 1916 february 19 1945 was a professional american football
offensive lineman in the national football league he played the 1937 1938 and 1939 college football seasons at the
university of georgia before joining the green bay packers for the 1940 and 1941 seasons he joined the united states
marine corps in 1942 and became an officer in addition to seeing combat with the 4th marine division he played for a
service football team in maui hawaii he served with i company 3rd battalion 23rd marines through the battles of
kwajalein saipan earning a silver star and tinian on february 19 1945 1st lieutenant johnson was killed in action by
a mortar shell at the battle of iwo jima and awarded a second silver star posthumously he was one of three former nfl
players to die on iwo jima along with jack chevigny and jack lummus johnson was buried at the national memorial
cemetery of the pacific in honolulu on february 2 1949
Score 0.7108: American football
Score 0.7107: National Football League
Score 0.3709: College football
Score 0.1509: Green Bay Packers
Score 0.1326: United States
Score 0.0432: World War II
Score 0.0244: College Football All-America Team
Score 0.0174: Association football
Score 0.0170: Offensive_lineman
Score 0.0124: Washington_Redskins
Score 0.0113: Guard_(American_football)
Score 0.0104: Canadian Football League
Score 0.0067: NFL draft
Score 0.0061: University of Southern California
Score 0.0057: Offensive_tackle
Score 0.0056: College football national championships in NCAA Division I FBS
Score 0.0055: Starting lineup
Score 0.0055: Fumble
Score 0.0052: Chicago Bears
Score 0.0049: Pro Bowl
----------------------------------------
Text Input: vermont route 12 vt 12 is a 101 627 mile long 163 553 km north south state highway in vermont that runs
from weathersfield to morrisville route 12 is one of the vermont roads on which moose are most often encountered they
are common from worcester to elmore route 12 begins at the new hampshire state line on the connecticut river in the
town of weathersfield it continues north along the west bank of the connecticut river overlapped with u s route 5
until hartland it then heads northwest to woodstock and then north through montpelier to end at vermont route 15a in
morrisville vermont route 12 runs parallel to interstate 89 from the woodstock hartford vicinity to montpelier
vermont route 12a is a state highway in central vermont united states it provides an alternate route to vt 12 between
randolph and northfield via braintree granville and roxbury the road currently used by vermont route 12a was
originally designated new england interstate route 12a as part of the new england interstate route system and existed
as such until it was replaced by a different system in 1926
Score 0.9774: United States
Score 0.6507: Vermont
Score 0.1684: State highway
Score 0.1191: Connecticut
Score 0.0879: Connecticut River
Score 0.0837: New Hampshire
Score 0.0607: List of counties in Vermont
Score 0.0584: Area code 802
Score 0.0358: Concurrency (road)
Score 0.0340: List_of_towns_in_Vermont
Score 0.0249: Massachusetts
Score 0.0238: U.S. state
Score 0.0230: Eastern Time Zone
Score 0.0226: Pennsylvania
Score 0.0181: New York (state)
Score 0.0170: UTC-4
Score 0.0111: Tennessee
Score 0.0082: en
Score 0.0066: Maine
Score 0.0054: State_highway_(US)
----------------------------------------
Text Input: the north american saxophone alliance nasa is an organization for saxophone players from around north
america following the lead of their colleagues in france who created the association of french saxophonists in 1971
the north american saxophone alliance was established in 1976 under the leadership of frederick hemke since this time
nasa has offered state regional and international conferences attracting many important saxophonists to present
```

```
performances lectures and master classes as well as found competitions for the next generation of classical and jazz
saxophonists nasa is the largest saxophone organization in the western hemisphere dedicated to the establishment of
the saxophone as a medium of serious musical expression members are required to pay dues which vary depending on age
nasa hosts regional conferences for each of its 10 regions information below it also hosts a biennial international
conference 2023 the university of southern mississippi host dannel espinoza 2020 arizona state university host
christopher creviston 2018 university of cincinnati host james bunte 2016 texas tech university host david dees 2014
university of illinois at urbana champaign hosts debra richtmeyer j michael holmes 2012 arizona state university host
timothy mcallister 2010 university of georgia host kenneth stephen fischer 2008 university of south carolina host
clifford leaman 2006 university of iowa host kenneth tse 2004 university of north carolina host steve stusek 2002
university of north texas host eric nestler 2000 university of arizona host kelland thomas 1998 northwestern
university hosts frederick hemke jonathan helton 1996 university of florida host jonathan helton 1994 west virginia
university host david hastings curtis johnson nasa is divided into eleven regions dividing canada the united states
of america and surrounding territories region 1 washington oregon idaho montana wyoming alaska region 2 california
nevada utah arizona colorado new mexico hawaii region 3 north dakota south dakota nebraska minnesota iowa region 4
kansas oklahoma missouri texas arkansas region 5 wisconsin illinois indiana ohio michigan region 6 louisiana
mississippi alabama georgia florida puerto rico region 7 kentucky tennessee virginia north carolina south carolina
maryland delaware washington d c region 8 new york pennsylvania new jersey west virginia connecticut massachusetts
rhode island vermont new hampshire maine region 9 british columbia alberta saskatchewan manitoba yukon northwest
territories region 10 ontario quebec newfoundland new brunswick nova scotia prince edward island the saxophone
symposium is the official peer reviewed journal of nasa issn 0271 3705
Score 0.9738: South Carolina
Score 0.9581: Arizona
Score 0.9555: Texas
Score 0.9533: North Carolina
Score 0.9148: Minnesota
Score 0.8954: Florida
Score 0.8774: Virginia
Score 0.8744: Alabama
Score 0.8630: Georgia (U.S. state)
Score 0.8617: Pennsylvania
Score 0.8179: Illinois
Score 0.7911: Colorado
Score 0.7896: Washington (state)
Score 0.7736: Oregon
Score 0.7726: Kentucky
Score 0.7215: Montana
```

Tag pridictions for data with little to no tags

```
validating_text_path = "/content/validation-data.txt"
```

```
parsed_result = model_moreGrams.preprocessor.load_data_from_file(validating_text_path, output_text_path)
Y_tst = parsed_result["label_matrix"]
corpus = parsed_result["corpus"]
```

```
X = model_moreGrams.preprocessor.predict(corpus)
```

```
Y_pred = batch_predict(model_moreGrams, X, batch_size=500)
Y_pred = vstack(Y_pred.ravel())
```

```
Predicting: 100%|██████████████████████████████████████| 29/29 [01:01<00:00,  2.11s/it]
```

```
def set_css():
  display(HTML('''
  <style>
    pre {
        white-space: pre-wrap;
    }
  </style>
  '''))
get_ipython().events.register('pre_run_cell', set_css)
```

```
num_samples = min(10, len(corpus))
random_indices = random.sample(range(len(corpus)), num_samples)

for i in random_indices:
    text = corpus[i]
    print(f"Text Input: {text}")
    for j in range(Y_pred.indptr[i], Y_pred.indptr[i + 1]):
        pred_label = model_moreGrams.output_items[Y_pred.indices[j]]
        pred_score = Y_pred.data[j]
        print(f"Score {pred_score:.4f}: {pred_label}")
    print("-" * 40)
```

```
as totally these concerns from senators scientist and government officials inform the thought process behind the
association between poverty health and economy with population throughout the 20th century when americans began to
occupy the island of puerto rico they asserted more than their ideals and beliefs american colonizers asserted
absolute dominance over puerto rico due to the idea of manifest destiny which greatly shifted the dynamics of the
island the u s capitalized on the fact that puerto rico utilized a large fraction of its resources to gain
independence from spain which left the island s economy depleted during this time many puerto ricans lost land while
their natural resources became exploited in the mid 1920s puerto rico s dependency on the production of sugar
devastated the island when the sugar market collapsed additionally the nation wide economic depression in 1927
exacerbated the effects of this collapse as well as the overall stability of the island in 1928 puerto rico suffered
the consequences of a hurricane in san felipe the okeechobee hurricane resulted in over 300 deaths and property
damages ranging from 50 80 million while the agricultural market also suffered in the 1930s puerto rican citizens
began to experience the adverse health effects of tuberculosis malaria diarrhea enteritis hookworm and dietary
deficiencies that were responsible for over 40 percent of deaths this later on gave medical professionals grounds to
support sterilization on the island furthermore these factors resulted in immense and widespread poverty many puerto
ricans faced perpetual hunger and growing unemployment rates in 1930 the median family income was reported to be
approx 250 a year and economically productive families were attributing around 94 of their income toward acquiring
food additionally 27 of the labor force was unemployed the current state of puerto rico confirmed the ideals
americans projected in the midst of the island s annexation about the longevity and potential of puerto rico puerto
ricans were once again viewed as ignorant and devious as they participated in reckless breeding in the midst of this
economic downward spiral this caused many americans and a fraction of puerto ricans to believe that overpopulation
essentially was the cause of the wide variety of problems on the island
Score 0.2906: Puerto Rico
Score 0.1173: United States
Score 0.0686: Spain
Score 0.0484: KEGG
Score 0.0249: Time zone
Score 0.0153: Chemical formula
Score 0.0148: Daylight saving time
Score 0.0145: Canada
Score 0.0143: CAS Registry Number
Score 0.0140: Simplified molecular-input line-entry system
Score 0.0140: JSmol
Score 0.0139: Molar mass
Score 0.0134: ChemSpider
Score 0.0134: International Chemical Identifier
Score 0.0128: Mexico
Score 0.0105: Unique Ingredient Identifier
Score 0.0098: France
Score 0.0090: Standard state
Score 0.0090: PubChem
Score 0.0087: CompTox Chemicals Dashboard
----------
Text Input: autism spectrum disorder asd is a neuro developmental disorder most commonly diagnosed in childhood and
is characterized by deficits in social and communication skills symptoms include social impairments hyper fixations
repetitive behaviors and hypersensitivity asd severity falls on a spectrum which means some individuals may have very
severe symptoms and social impairments and might need substantial assistance while others require less support asd
individuals have been shown to have abnormal reduced intrinsic functional connectivity in their default mode network
dmn as well as disruptions in their frontoparietal network fpn or cen and salience network sn most notably for the sn
asd patients have been shown to have hypoactivity in the anterior insula one of the anchoring points of the sn in the
brain it is thought that these disruptions within networks result in disrupted interactions between networks
resulting in the asd pathology more specifically the reduced activity in the sn leads to deficient signaling to the
fpn and the dmn leading to a disengagement of cognitive systems important for attending to salient external stimuli
or internal mental events
Score 0.7608: PubMed
Score 0.7457: Gene nomenclature
Score 0.7433: Mendelian_Inheritance_in_Man
Score 0.7417: Laboratory mouse
Score 0.7407: GeneCards
Score 0.7406: UniProt
Score 0.7402: Orthologs
Score 0.7398: Entrez
Score 0.7374: Ensembl
Score 0.7299: Base pair
Score 0.7289: HomoloGene
Score 0.7288: Human genome
Score 0.7106: Gene expression
Score 0.7061: Gene_ontology
Score 0.6994: Chromosome
Score 0.6993: Locus (genetics)
Score 0.6743: Mouse Genome Informatics
Score 0.3851: Protein Data Bank
Score 0.3698: GO:0005515
Score 0.2837: Protein-protein_interaction
----------
Text Input: in august 1908 hofmeyr was appointed secretary for the transvaal delegation to the national convention
this placed him in close contact with the leading south african politicians of the day and the debates held at the
national convention strongly influenced and informed his later actions as administrator of south west africa a
consensus had been emerging that the time was right for the merger of the cape colony the colony of natal the orange
river colony the transvaal colony and perhaps southern rhodesia now zimbabwe northern rhodesia now zambia basutholand
now lesotho swaziland protectorate now eswatini and bechuanaland now botswana in 1908 jan smuts then colonial
secretary and education secretary in the transvaal government wrote to john x merriman then prime minister of the
cape colony about the need for a speedy union during recent months he said a dangerous movement had been growing in
the transvaal a movement for separatism similar to that which had existed before the boer war while smuts and
merriman agreed on many things merriman was concerned about the native franchise he predicted that the cape would not
```

```
want to give it up and that the transvaal and others would not want to adopt it in february 1908 merriman suggested
to smuts that their difficulty might be resolved by leaving the question of the native franchise to the provinces
themselves in may 1908 delegations from the colonies met in pretoria smuts moved for a resolution stating that the
best interests and permanent prosperity of south africa could only be secured by an early union of the four colonies
under the crown of great britain and that delegates from each of the cape transvaal orange river colony and natal be
sent to a national south african convention the national convention first met in october 1908 behind closed doors and
open windows in the sultry heat of a durban summer prominent members of the transvaal delegation included general
louis botha then prime minister of the transvaal and later the first prime minister of the union of south africa
general smuts colonial secretary of the transvaal and later the second prime minister of the union of south africa
and the prime minister that appointed gysbert reitz hofmeyr as administrator of south west africa in 1920 and sir
james percy fitzpatrick a mining financier and author of the classic children s book jock of the bushveld who smuts
had convinced that union would be the fulfilment of the conciliation policy
Score 0.8245: South Africa
Score 0.0563: Zimbabwe
Score 0.0215: Eswatini
Score 0.0210: Botswana
Score 0.0182: Lesotho
Score 0.0165: South_African_Standard_Time
Score 0.0115: Namibia
Score 0.0100: Soviet Union
Score 0.0096: United Nations
Score 0.0089: Coloureds
Score 0.0088: Zambia
Score 0.0085: White_South_African
Score 0.0082: Bantu peoples of South Africa
Score 0.0081: Indian_South_African
Score 0.0079: Asian_South_African
Score 0.0074: Afrikaans
Score 0.0073: UTC+2
Score 0.0060: Post-office_box
Score 0.0057: KwaZulu-Natal
Score 0.0053: Mozambique
----------------------------------------
Text Input: the coefficients a 1 2 2 x 1 x 3 2 textstyle a_ 1 2 frac 2 x frac 1 x frac 3 2 and a 2 2 x 2 x 2 x 2 2 x
3 2 textstyle a_ 2 frac 2 x frac 2x 2 x 2 2x frac 3 2 are rational solutions of the riccati equation a a 2 2 1 x 4 x
2 0 textstyle a a 2 left 2 frac 1 x right frac 4 x 2 0 they yield the fundamental system example 2 an equation with a
type l 3 2 displaystyle mathcal l _ 3 2 decomposition is the coefficient of the first order factor is the rational
solution of a a 2 6 x 2 0 textstyle a a 2 frac 6 x 2 0 upon integration the fundamental system y 1 x 3 textstyle y_ 1
x 3 and y 2 1 x 2 textstyle y_ 2 frac 1 x 2 for c 0 displaystyle c 0 and c displaystyle c to infty respectively is
obtained these results show that factorization provides an algorithmic scheme for solving reducible linear ode s
whenever an equation of order 2 factorizes according to one of the types defined above the elements of a fundamental
system are explicitly known i e factorization is equivalent to solving it a similar scheme may be set up for linear
ode s of any order although the number of alternatives grows considerably with the order for order n 3 displaystyle n
3 the answer is given in full detail in if an equation is irreducible it may occur that its galois group is
nontrivial then algebraic solutions may exist if the galois group is trivial it may be possible to express the
solutions in terms of special function like e g bessel or legendre functions see or in order to generalize loewy s
result to linear pdes it is necessary to apply the more general setting of differential algebra therefore a few basic
concepts that are required for this purpose are given next a field f displaystyle mathcal f is called a differential
field if it is equipped with a derivation operator an operator δ displaystyle delta on a field f displaystyle mathcal
f is called a derivation operator if δ a b δ a δ b displaystyle delta a b delta a delta b and δ a b δ a b a δ b
displaystyle delta ab delta a b a delta b for all elements a b f displaystyle a b in mathcal f a field with a single
derivation operator is called an ordinary differential field if there is a finite set containing several commuting
derivation operators the field is called a partial differential field
Score 0.0790: Chemical formula
Score 0.0757: Simplified molecular-input line-entry system
Score 0.0754: JSmol
Score 0.0754: CAS Registry Number
Score 0.0748: Molar mass
Score 0.0718: International Chemical Identifier
Score 0.0635: ChemSpider
Score 0.0595: Standard state
Score 0.0528: PubChem
Score 0.0407: Unique Ingredient Identifier
Score 0.0350: CompTox Chemicals Dashboard
Score 0.0321: Chembox_validation
Score 0.0187: Protein Data Bank
Score 0.0135: Density
Score 0.0128: ECHA_InfoCard
Score 0.0107: Chemical nomenclature
Score 0.0105: English
Score 0.0089: PubMed
Score 0.0076: Epoch (astronomy)
Score 0.0066: Taxonomy (biology)
----------------------------------------
Text Input: what mek quake lacks in intelligence which is a lot he makes up for in cruelty and destructive force
however he is also something of a coward preferring softer targets that he can inflict more pain upon while chanting
his catchphrase big jobs he has no compunctions about massacring humans as evidenced in the third element when he is
annoyed about having missed most of a battle because of the time it took to remove all the human gristle clogging his
caterpillar tracks in his early stories he openly loathed hammerstein and ro jaws as a result of their time in ro
busters but later this hatred was toned down to a general dislike of everything and everybody except videos of
chainsaw torture despite this mek quake is eager to show his intelligence and importance and in kronicles of khaos he
kept trying to join in at the end of deadlock s statements so he could seem like he was already versed in khaos he
continued to be part of the warriors in their return to mars as they tried to keep the peace during a civil war
developing an interest in conspiracy theories and rap music but it was clear that his increasing mental instability
was putting the rest of the team in danger after the defeat of the shadow warriors mek quake was checked into a robot
asylum by the others while hammerstein seemed sad to see his former enemy depart the others appeared more than eager
```

```
to meet his replacement zippo mek quake was assigned to clean the cell of former volgan general volkhan but when
blackblood called up to gleefully tell him he was going to be replaced an embittered mek quake freed volkhan and his
associates and helped kill the asylum staff he then joined with volkhan and blackblood in an attempt to destroy the
rest of the abc warriors as well as marineris city when the rebellion failed and he was abandoned to his fate by
volkhan s troops mek quake managed to escape destruction with steelhorn s fusion hammer and unwittingly found himself
a celebrity in the union of martian free states for the destruction of the marinus red house after promoting his book
tour mek quake has found employment as howard quartz s bodyguard and was most recently responsible for murdering
tubal caine s adopted son
Score 0.0470: United States
Score 0.0124: Germany
Score 0.0123: France
Score 0.0038: World War II
Score 0.0034: Los Angeles
Score 0.0028: PubMed
Score 0.0024: Protein Data Bank
Score 0.0022: Locus (genetics)
Score 0.0021: Human genome
Score 0.0020: Mouse Genome Informatics
Score 0.0020: Entrez
Score 0.0020: Gene expression
Score 0.0020: Base pair
Score 0.0020: Gene nomenclature
Score 0.0020: Orthologs
Score 0.0020: GeneCards
Score 0.0020: Chromosome
Score 0.0019: Laboratory mouse
Score 0.0019: HomoloGene
Score 0.0019: Ensembl
----------------------------------------
Text Input: temp capt robert william rowland law mc maj hervey major lawrence dso scottish rifles lt col henry gordon
leahy royal garrison artillery maj alfred leamy royal army ordnance corps lt col harold ledward maj john robert lee
frcs lt col roderick livingstone lees dso vd lancashire fusiliers lt victor lefobure essex regiment maj edward james
leggett royal army ordnance corps maj robert anthony linington leggett dso worcestershire regiment capt geoffrey
hamilton leigh south lancashire regiment temp maj h s le rossignol royal jersey militia col robert thomas morland
lethbridge army pay department lt col charles cameron leveson gower cmg royal artillery temp capt george ernest lewis
royal army service corps maj cuthbert hillyer ley royal engineers quartermaster and maj harry sylvanus lickmau ext
reg empl hon maj willie cresswell link royal army ordnance corps capt victor alexander john hope marquess of
linlithgow lothians and border horse maj sir john lister kaye bt royal army service corps bt lt col john little
northumberland regiment maj marchall william litton royal irish fusiliers maj george william david bowen lloyd royal
welsh fusiliers capt thomas lodge royal west surrey regiment maj francis carleton logan logan lancashire fusiliers
maj william logan royal army veterinary corps maj sydney francis mcilree lomer king s royal rifle corps capt gerard
hanslip long suffolk regiment capt henry john leicester longden mbe army school department maj charles frederick
gemley low royal army ordnance corps quartermaster and capt james lindsay low royal engineers temp maj andrew alfred
lowe royal engineers lt col thomas enoch lowe south staffordshire regiment temp maj reginald hugh lucas royal army
service corps col thomas lucas woodwright lucas mbe glamorgan volunteer corps capt dudley owen lumley mbe wiltshire
regiment edith mary lyde rrc matron queen alexandra s imperial military nursing service maj arthur abram lyle london
regiment temp capt oliver lyle highland light infantry maj charles joseph edward addis mcarthur kings own scottish
borderers capt henry montray jones mccance capt frederic ewing mcclellan middlesex regiment temp maj michael
mccormack mc royal west surrey regiment maj john mcdermott indian army temp hon lt col peter macdiurmid royal army
medical corps maj andrew edward macdonald cameron highlanders capt angus g macdonald royal army medical corps 2nd lt
james mcdonald king s own scottish borderers maj john mci mcdougall royal garrison artillery maj donald keith
mcdowell cmg royal army medical corps temp maj samuel johnson mcdowell army pay department temp maj james mcewen
staff for royal engineers service maj albert william crawford mcfall yorkshire light infantry temp capt charles
hamilton mcguinness capt james douglas macindoe mc scots guards capt alexander donald mackeanzie royal engineers maj
colin mansfield mackenzie dso london regiment capt eric francis wallace mackenzie mc royal army medical corps lt col
robert wilson mckergow sussex yeomanry
Score 0.2129: Scotland
Score 0.0757: United States
Score 0.0646: England
Score 0.0431: London
Score 0.0313: Wales
Score 0.0092: Edinburgh
Score 0.0091: History of Scotland
Score 0.0091: List of years in Scotland
```