Natural Language Processing

21MID0181

Samay Raj Adhikari

-Task Computation-

Under: Prof. Sharmila Banu K

Title: Magnum Opus Task for computing the distance between two documents.

Problem Statement:

In this project, I aim to determine the similarity between two documents by analysing their textual content. The goal is to establish whether the given documents share a significant amount of content and to quantify their similarity using cosine similarity.

Now, for the computation of this task, I have decided to do it on a Research paper on the base topic, Flood detection.

Now going through the contents of the paper I found out paper1 uses:

Flood Detection Using Machine Learning,

while paper2 uses:

A Real-Time Flood Detection System Based on Machine Learning Algorithms with Emphasis on Deep Learning.

Each paper will have its own approach to the solution and to verify how similar these two documents might be, I have used the cosine similarity method.

Further into my document we will be able to see how the two approaches to flood detection papers are similar or dissimilar.

Approach:

To accomplish this, I implemented a two-step process:

1. **Text Extraction & Highlighting Common Words**
    - Extract text from two given PDF documents.
    - Preprocess the text by converting it to lowercase, removing non-alphabet characters, and filtering out stop words.
    - Identify and highlight words that appear in both documents.

2. **Computing Document Similarity**
    - Convert the cleaned text into a vector space using TF-IDF (Term Frequency-Inverse Document Frequency).
    - Compute the cosine similarity between the two document vectors to measure how closely related they are.
    - Identify unique words in each document that do not appear in the other.

Implementation:

Step 1: Document Processing & Highlighting Common Words

- I extracted the first 5000 words from each document and displayed them while highlighting the common words in red.
- This step visually demonstrates the overlapping content between the two texts.

Step 2: Document Similarity Computation

- I converted both documents into a TF-IDF vector representation.
- Using SciPy's cosine function, I calculated the cosine similarity between the two document vectors.
- The cosine similarity score determines how much the documents resemble each other:
    - **1.0** → The documents are identical.
    - **0.0** → The documents have no similarity.
    - **Between 0 and 1** → Partial similarity based on shared content.

Results:

For each document, I extracted key statistics:

- **Total Words** in each document.
- **Unique Words** present in each document but not in the other.
- **Common Words** that exist in both.

Finally, I presented the cosine similarity score, which quantifies the degree of similarity between the two documents. A higher score indicates greater similarity, while a lower score suggests that the documents have fewer shared elements.

Key Takeaways:

- **Cosine similarity** provides a reliable way to measure textual similarity.
- **TF-IDF transformation** helps normalize the text and gives weight to important words.
- **Highlighting common words** gives a visual representation of shared content.
- **Extracting unique words** helps differentiate between the two documents.

With the help of this project, document comparison tasks such as plagiarism detection, content similarity analysis, and textual overlap studies in research papers or articles will become easier and will users know how similar it is to a paper or article that has already been published. This will further restrict people from stealing ideas or using one's document to produce their own.

The properly functionality and coding for this tool is given below.

The coding has been done in "Jupyter Notebook" using a python3 kernel.

In [20]:

```
import pdfplumber
import re
import nltk
from nltk.corpus import stopwords
from termcolor import colored

# Ensure necessary NLTK resources are available
nltk.download('stopwords')

# Define stop words
stop_words = set(stopwords.words('english'))

# Function to extract text from a PDF file
def extract_pdf_text(file_path):
    text = ""
    try:
        with pdfplumber.open(file_path) as pdf:
            for page in pdf.pages:
                text += page.extract_text()
        return text
    except Exception as e:
        print(f"Error: {e}")
        return None

# Function for text preprocessing
def preprocess_text(text):
    text = text.lower()  # Convert to lowercase
    text = re.sub(r'[^a-z\s]', '', text)  # Remove non-alphabet characters
    words = text.split()
    words = [word for word in words if word not in stop_words]  # Remove stopwords
    return ' '.join(words)

# Define file paths (update with your actual PDF file paths)
file_paths = [
    "C:/Users/samay raj/Downloads/flood_detection_paper_1.pdf",
    "C:/Users/samay raj/Pictures/flood_detection_paper_2.pdf"
]

# Extract text from PDFs
```

```
articles = [extract_pdf_text(file_path) for file_path in file_paths]

# Check if reading was successful
if None in articles or any(len(article) < 100 for article in articles):
    print("Error: Could not extract text from articles.")
    exit()

# Preprocess both articles (limiting to first 5000 words)
text_limit = 5000
preprocessed_articles = [preprocess_text(article) for article in articles]

# Limit to the first 5000 words
doc1_words = preprocessed_articles[0].split()[:text_limit]
doc2_words = preprocessed_articles[1].split()[:text_limit]

# Find common words
common_words = set(doc1_words) & set(doc2_words)

# Highlight common words in red in both documents
highlighted_doc1 = ' '.join([colored(word, 'red') if word in common_words else word for word in doc1_words])
highlighted_doc2 = ' '.join([colored(word, 'red') if word in common_words else word for word in doc2_words])

# Display highlighted documents
print("\nDocument 1 - Highlighted Text (Common Words in Red):")
print(highlighted_doc1[:5000])  # Show first 5000 characters

print("\nDocument 2 - Highlighted Text (Common Words in Red):")
print(highlighted_doc2[:5000])  # Show first 5000 characters
```

```
[nltk_data] Downloading package stopwords to C:\Users\samay
[nltk_data]     raj\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
Document 1 - Highlighted Text (Common Words in Red):
flood detection using machine learning krishna gopal sharma rashmeet kaur b harsh dalal c raghav laddha tejinder thi
nd e lovely professional university phagwara punjab krishnagmailcom b rashmeetgmailcom c mrharshdalalgmailcom rvladd
hagmailcom e tejinderlpucoin abstract mankind experienced extreme disasters natural manmade throughout past purpose
research paper analyze one natural disasters floods adverse effect forecast occurrence floods thus take corrective a
ctions within time one common natural disasters occur near water resources particularly near river areas floods rega
rded one prone natural disasters gis technology used visualize flood extent analyze consequences risks associated di
saster paper analyses detects floods using gis dataset machine learning model help gis datasets models used determin
e probability flooding particular area gis dataset used sentinel research helpful method analyze occurrence floods h
```

elp authorities take preventive measures keywords sentinel machine learning sequential model feature extraction dimensionality reduction introduction flooding overflow water land besides powerful natural disaster also adverse effect human beings animals plants organisms earlier days traditional methods used predict floods varied depending time place many cultures people used observe natural signs indicated floods signs included animal behaviour colour movement sky level colour water rivers lakes places people measure rainfall predict floods utilizing rain gauges people use astrology predict floods would look positions stars planets determine floods possible eventually floodwalls levees built along riverbank process modern methods ml ai gis many others come contact paper focuses studies floodprone areas using sentinel images develops machinelearning model detect flooding dataset sentinel satellite split stored two separate repositories first repository actual images another repository containing label data files tiff extension read model designed compare flooded area nonflooded area files json geojson extensions read machine learning algorithms build models basic data retrieved files figwhich illustrates two different repositories managed local machine electronic copy available httpsssrncomabstractfig images folder imagesimagespng fig images labels fig images folder repositories imagesimagespng fig illustrates images assigned repository name indicates sentinel satellite data locations prefix sources date sentinel fig demonstrates image divided different spectral information next step multispectral image created stacking bands together fig represents dataset hierarchy used whole study fig dataset hierarchy literature review amir mosavi pinar ozturk kwok wing chau paper proposed machine learning ml methods contributed prediction system significantly given us costeffective solutions paper predict methods short longterm floods also stated main trends proving best quality machine learning models flood prediction according data decompos

Document 2 - Highlighted Text (Common Words in Red):

see discussions stats author profiles publication httpswwwresearchgatenetpublication realtime flood detection system based machine learning algorithms emphasis deep learning article international journal engineering trends technology may doi ijettvip citations reads authors including abdirahman osman hashi mohamed abdi abdi simad university simad university publications citations publications citations see profile see profile content following page uploaded abdirahman osman hashi june user requested enhancement downloaded fileinternational journal engineering trends technology volume issue may issn doiijettvip seventh sense research group realtime flood detection system based machine learning algorithms emphasis deep learning abdirahman osman hashi abdullahi ahmed abdirahman mohamed abdirahman elmi siti zaiton mohd hashi octavio ernesto romo rodriguez faculty member simad university department computing mogadishu somalia department artificial intelligence big data faculty computing university malaysia kelantan kelantan malaysia department computer science faculty informatics istanbul teknik niversitesi istanbul turkey wadanigmailcom aaayaresimadeduso mabdirahmansimadeduso abstract flood expressed water overflowing onto keywords machine learning naive bayes random forest ground usually dry increase water artificial intelligence convolutional neural network data significant impact human life also declared mining natural language processing one usual natural phenomena causing severe introduction financial damage goods properties well well known natural disasters cannot avoided affecting human lives however preventing floods however prealarming systems proper management would useful inhabitants order get sufficient mitigate severity impact time evacuate areas might susceptible meteorological departments developed countries floods happen regarding issue floods floodmonitoring cells may appropriately numerous scholars proposed different solutions instance equipped intelligent scalable flood alarming developing prediction models building proper system hand countries may infrastructure nevertheless economical department including country somalia consequence perspective proposed solutions inefficient people living floodaffected susceptible areas people countries like somalia instance hence dealing aftermath floods every year main objective present research paper propose somalia dangerous unexpected floods

```
occurred novel robust model realtime flood detection baladweyn town hiran region last year system based machinelearn
ingalgorithms deep reported people displaced learning random forest naive bayes j consequence also riverflooding far
convolutional neural networks detect water level impact estimated people somalia measure floods possible humanitaria
n around people displaced consequences occur experimental results fled houses consequence heavy rains proposed metho
d solution forth mentioned happened ethiopia received across country problems conduct research easily affected espec
ially southern regions also hiran simulating novel way detects water levels using region among according unhcrled hy
brid model based arduino gsm modems based protection analysis randomforest algorithm outperformed increased rainwate
r since beginning may machine learning models regarding accuracy stated sharp rise water levels jubba shabelle compa red
alternative classification methods rivers result might lead severe flooding accuracy contrast central southern r egions
somalia according achieved using naive bayes j respectively unhcrled flood magnitude occurred hand using deep learning
approach achieved baladweyn last▯ [
```

In [28]:

```python
import pdfplumber
import re
import nltk
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
from scipy.spatial.distance import cosine

# Ensure necessary NLTK resources are available
nltk.download('stopwords')

# Define stop words
stop_words = set(stopwords.words('english'))

# Function to extract text from a PDF file
def extract_pdf_text(file_path):
    text = ""
    try:
        with pdfplumber.open(file_path) as pdf:
            for page in pdf.pages:
                text += page.extract_text()
        return text
    except Exception as e:
        print(f"Error: {e}")
        return None

# Function for text preprocessing
def preprocess_text(text):
    text = text.lower()  # Convert to lowercase
    text = re.sub(r'[^a-z\s]', '', text)  # Remove non-alphabet characters
    words = text.split()
    words = [word for word in words if word not in stop_words]  # Remove stopwords
    return ' '.join(words)

# Define file paths (update with your actual PDF file paths)
file_paths = [
    "C:/Users/samay raj/Downloads/flood_detection_paper_1.pdf",
    "C:/Users/samay raj/Pictures/flood_detection_paper_2.pdf"
]

# Extract text from PDFs
articles = [extract_pdf_text(file_path) for file_path in file_paths]

# Check if reading was successful
if None in articles or any(len(article) < 100 for article in articles):
    print("Error: Could not extract text from articles.")
    exit()

# Preprocess both articles
preprocessed_articles = [preprocess_text(article) for article in articles]

# Convert to vector space using TF-IDF for both documents
vectorizer = TfidfVectorizer()
vectors = vectorizer.fit_transform(preprocessed_articles).toarray()

# Compute cosine similarity between the two documents
similarity_score = 1 - cosine(vectors[0], vectors[1])

# Find all common and unique words
words1 = preprocessed_articles[0].split()
words2 = preprocessed_articles[1].split()

common_words = set(words1) & set(words2)  # Intersection of words
unique_words1 = set(words1) - set(words2)  # Words unique to Document 1
unique_words2 = set(words2) - set(words1)   # Words unique to Document 2

# Display Results for Full Document Comparison
print("\n--- Results for Document 1 ---")
print(f"Total Words in Document 1: {len(words1)}")
print(f"Unique Words in Document 1: {len(unique_words1)}")
print(f"All Unique Words in Document 1: {sorted(list(unique_words1))}")
print("\n--- Results for Document 2 ---")
print(f"Total Words in Document 2: {len(words2)}")
print(f"Unique Words in Document 2: {len(unique_words2)}")
print(f"All Unique Words in Document 2: {sorted(list(unique_words2))}")

# Display Common Words
print(f"\n--- Common Words ---")
print(f"Total Common Words: {len(common_words)}")
print(f"All Common Words: {sorted(list(common_words))}")

# Display Cosine Similarity Score
print(f"\nCosine Similarity Score between Document 1 and Document 2: {similarity_score:.4f}")
```

```
[nltk_data] Downloading package stopwords to C:\Users\samay
[nltk_data]     raj\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

```
--- Results for Document 1 ---
Total Words in Document 1: 1122
Unique Words in Document 1: 373
All Unique Words in Document 1: ['abundance', 'actions', 'adopted', 'adverse', 'afan', 'affect', 'ai', 'aida', 'aler
ts', 'allawi', 'along', 'altitude', 'analyses', 'android', 'andtestlabels', 'animal', 'animals', 'asar', 'assigned',
'associated', 'astrology', 'axisaltitudestabilization', 'bad', 'bae', 'bands', 'basically', 'basis', 'batch', 'bbc',
'behaviour', 'behind', 'beings', 'bit', 'blog', 'bodies', 'books', 'boosts', 'build', 'calculated', 'calculating',
'canals', 'chang', 'changes', 'cities', 'cleaned', 'colour', 'columns', 'comparison', 'compiled', 'confusion', 'cons
idered', 'constructing', 'contact', 'continues', 'convd', 'convolution', 'copy', 'corrective', 'correspond', 'costef
fective', 'cover', 'created', 'creation', 'cultivation', 'cultures', 'cycle', 'dalal', 'dayrepeatcycle', 'decomposit
ion', 'decreases', 'deghyo', 'dense', 'depends', 'describe', 'detailed', 'develops', 'difficult', 'dimensions', 'dis
aster', 'divide', 'divided', 'dropout', 'earlier', 'efficient', 'eg', 'ehteram', 'eliminated', 'entirely', 'epoch',
'epochs', 'eventually', 'exact', 'examine', 'experienced', 'experts', 'exporting', 'extension', 'extensions', 'exten
t', 'extracted', 'extraction', 'falls', 'farahani', 'faridah', 'farzin', 'figconvdlayersummaryandparam', 'figtestlos
s', 'figwhich', 'fijohn', 'file', 'files', 'fitted', 'flatten', 'flooded', 'floodprone', 'floodwalls', 'floodwater',
'fluctuates', 'focus', 'focuses', 'folder', 'folders', 'forests', 'gain', 'gauges', 'generation', 'geojson', 'gis',
'gokmen', 'google', 'googleblogcom', 'gopal', 'government', 'gradient', 'graph', 'graphafter', 'graphs', 'haitham',
'harsh', 'hierarchy', 'higher', 'highlevel', 'hojat', 'hongbo', 'hsu', 'httpsssrncomabstract', 'httpsssrncomabstract
fig', 'httpsssrncomabstractphysicsbased', 'httpsssrncomabstractsince', 'hybridization', 'hydronets', 'identify', 'i
e', 'illustrated', 'illustrates', 'imagesimagespng', 'improvements', 'include', 'increases', 'indicate', 'indicate
d', 'indicates', 'interconnected', 'involved', 'iterates', 'jabbari', 'jianzhong', 'json', 'jsondata', 'karami', 'ka
ur', 'keraslayers', 'kerasmodel', 'kg', 'km', 'knowing', 'krishna', 'krishnagmailcom', 'krupa', 'kruti', 'kunverji',
'kuolin', 'kwok', 'label', 'laddha', 'lakes', 'land', 'launch', 'launcher', 'layer', 'lb', 'levees', 'lichiu', 'lite
rature', 'little', 'loaddata', 'local', 'longterm', 'look', 'looking', 'lovely', 'machinelearning', 'managed', 'mank
```

```
ind', 'manmade', 'mass', 'maxpoold', 'mi', 'millions', 'minutesorbitalperiod', 'mmmftftft', 'modelling', 'mohammad',
'mohammed', 'moisture', 'moment', 'morphological', 'movement', 'mrharshdalalgmailcom', 'much', 'multispectral', 'nam
e', 'nasim', 'nazanin', 'nd', 'near', 'nearpolar', 'news', 'nonflooded', 'nonrelated', 'nontrainable', 'occurrence',
'optimization', 'orbit', 'organisms', 'others', 'otherwise', 'othman', 'outcome', 'overflow', 'overwhelming', 'ozgu
r', 'ozturk', 'particularly', 'per', 'period', 'phagwara', 'pinar', 'places', 'planets', 'plants', 'plot', 'plots',
'powerful', 'predicts', 'prefix', 'preventive', 'probability', 'professional', 'prone', 'proving', 'publicly', 'punj
ab', 'quality', 'raghav', 'rain', 'rashmeet', 'rashmeetgmailcom', 'rate', 'rather', 'raw', 'read', 'reduction', 'reg
arded', 'repeated', 'repositories', 'repository', 'represents', 'requires', 'reservoirs', 'resources', 'responsibl
e', 'returns', 'revolutionspercycle', 'rice', 'risks', 'riverbank', 'rvladdhagmailcom', 'saeed', 'sample', 'sample
s', 'satellite', 'say', 'searches', 'seen', 'senflood', 'sentinel', 'shah', 'sharma', 'show', 'shown', 'significantl
y', 'signifies', 'signs', 'singh', 'sky', 'soil', 'sources', 'soyuz', 'specificationsofthesentinelsatellites', 'spec
tral', 'split', 'st', 'stacking', 'stars', 'stop', 'stored', 'strategies', 'streams', 'structures', 'struggling', 's
tudies', 'suddenly', 'suggestions', 'summary', 'sunsynchronous', 'surroundings', 'targeting', 'tayfur', 'tejinder',
'tejinderlpucoin', 'tested', 'testfeatures', 'testimages', 'testingdataset', 'testingdatasetbased', 'testingdataset
s', 'testlabels', 'testloss', 'thind', 'thing', 'throughout', 'tiff', 'together', 'topography', 'traditional', 'trai
n', 'trainacc', 'trainaccuracy', 'trainingdata', 'trainingdataset', 'understood', 'unwanted', 'utilizes', 'utilizin
g', 'valacc', 'validationdata', 'validationdataaccuracy', 'valloss', 'varied', 'vegetation', 'vijay', 'ways', 'win
g', 'yaseen', 'yearlifetime', 'yearsforconsumables', 'yinghao', 'yu', 'zaher', 'zero', 'zhang', 'zhou']

--- Results for Document 2 ---
Total Words in Document 2: 3326
Unique Words in Document 2: 1068
All Unique Words in Document 2: ['aaayaresimadeduso', 'abdi', 'abdirahman', 'abdullahi', 'ability', 'able', 'acces
s', 'accurately', 'achieve', 'achieved', 'achieving', 'acknowledge', 'acknowledgment', 'acquire', 'action', 'activat
ion', 'actor', 'acts', 'adaptive', 'adaptiveneurofuzzy', 'adding', 'additional', 'adequate', 'adjusted', 'adjustmen
t', 'adjustments', 'advance', 'advanced', 'advances', 'affecting', 'aftermath', 'afterward', 'agriculture', 'ahmad',
'ahmed', 'aim', 'aj', 'al', 'alarming', 'aljumeily', 'allow', 'allowed', 'allows', 'alternating', 'alternative', 'al
ternatives', 'among', 'amount', 'amounts', 'analog', 'analogously', 'analysis', 'analyzed', 'analyzing', 'anfis', 'a
nn', 'announcement', 'anns', 'anticipated', 'anywhere', 'appl', 'applications', 'applied', 'apply', 'approaches', 'a
ppropriate', 'appropriately', 'approximation', 'april', 'ar', 'arduino', 'around', 'array', 'arrival', 'arshad', 'ar
ticle', 'artif', 'artificial', 'artificialneural', 'asking', 'aspect', 'assessment', 'attard', 'attribute', 'autho
r', 'authors', 'automatic', 'avoid', 'background', 'baladweyn', 'balance', 'balanced', 'balancing', 'barthelemy', 'b
ase', 'based', 'baydargil', 'bayes', 'becomes', 'behalf', 'beledweyne', 'beledweyneofficial', 'benefit', 'big', 'bio
logical', 'biomedical', 'boards', 'branches', 'briefly', 'brings', 'broadly', 'building', 'bus', 'byncnd', 'came',
'cannot', 'capability', 'capable', 'carried', 'carry', 'case', 'categories', 'categorization', 'categorize', 'cause
d', 'causing', 'cc', 'cells', 'center', 'central', 'certainty', 'cham', 'change', 'characteristics', 'chatterjee',
'chen', 'chip', 'chosen', 'choubin', 'cipullo', 'circumstances', 'citations', 'class', 'classes', 'classification',
'classifications', 'classified', 'classifier', 'classifiers', 'classify', 'classifying', 'classs', 'cleaning', 'clim
ate', 'close', 'clustering', 'cmhc', 'cnn', 'cnns', 'code', 'collecting', 'collection', 'combined', 'comes', 'commun
icate', 'communication', 'community', 'complexity', 'components', 'composed', 'compu', 'computational', 'computation
s', 'computer', 'computers', 'computing', 'concept', 'conclusion', 'condensed', 'condition', 'conditional', 'conduc
```

```
t', 'conducted', 'conference', 'connected', 'consequence', 'consideration', 'consolidating', 'constantly', 'constrai
nts', 'contain', 'contamination', 'content', 'context', 'contrast', 'control', 'controlled', 'controller', 'convenie
nt', 'convention', 'converted', 'converter', 'converting', 'converts', 'convolutional', 'corbin', 'core', 'cores',
'correctly', 'correlation', 'cost', 'coulon', 'countries', 'country', 'critical', 'crossplatform', 'ct', 'current',
'damage', 'damages', 'dangerous', 'datadriven', 'dcnn', 'dcnns', 'deal', 'dealing', 'decide', 'decided', 'decision
s', 'decisiontree', 'declared', 'degree', 'deleted', 'demonstrate', 'department', 'departments', 'dependencies', 'de
pletes', 'deployed', 'describes', 'description', 'desire', 'despite', 'detail', 'detected', 'detecting', 'detector',
'devastations', 'develop', 'developers', 'development', 'device', 'devices', 'differences', 'digital', 'dineva', 'di
scussed', 'discussion', 'discussions', 'displaced', 'displayed', 'distribute', 'distribution', 'djordjevic', 'doi',
'doiijettvip', 'done', 'downloaded', 'dramatically', 'drought', 'dry', 'dt', 'duncan', 'dynamic', 'early', 'easily',
'easy', 'economical', 'economy', 'education', 'effects', 'either', 'eliminating', 'elissa', 'elmi', 'embedded', 'emp
hasis', 'enables', 'enabling', 'encoderdecoder', 'eng', 'engineering', 'enhanced', 'enhancement', 'enhancements', 'e
nsemblepredictionsystem', 'ensure', 'environ', 'environment', 'epitomized', 'epss', 'equipment', 'equipped', 'ernest
o', 'error', 'especially', 'essential', 'essentially', 'estimated', 'et', 'ethiopia', 'evacuate', 'evacuated', 'evac
uating', 'even', 'events', 'evidently', 'evolution', 'exceed', 'experiences', 'experiment', 'experimental', 'experim
ents', 'expert', 'explains', 'expressed', 'expression', 'extend', 'extends', 'extremely', 'f', 'face', 'factors', 'f
aculty', 'fairly', 'far', 'fashion', 'fault', 'fed', 'feedforward', 'feng', 'fergus', 'fields', 'figure', 'figures',
'fileinternational', 'filter', 'filters', 'filterwrapper', 'financial', 'find', 'finish', 'firstly', 'firsttime', 'f
is', 'fit', 'five', 'fled', 'flee', 'fleet', 'floodaffected', 'floodcontrol', 'floodmonitoring', 'floodsrelated', 'f
luvial', 'follow', 'followed', 'forecasts', 'forest', 'forth', 'forward', 'found', 'four', 'fourth', 'framework', 'f
ulfilled', 'fully', 'functions', 'funding', 'furthermore', 'future', 'fuzzy', 'fuzzyinference', 'fuzzylogic', 'g',
'gained', 'gathered', 'gathering', 'gave', 'gene', 'generally', 'generalpurpose', 'generating', 'give', 'global', 'g
oing', 'good', 'goods', 'gotten', 'gps', 'grant', 'gratefully', 'gratitude', 'greater', 'ground', 'groundwater', 'gr
oup', 'gsm', 'gsms', 'guide', 'h', 'halbeeg', 'hameed', 'han', 'hand', 'handling', 'happen', 'happened', 'happens',
'hardware', 'hashi', 'hb', 'heavy', 'hence', 'hiran', 'hirshabelle', 'history', 'homes', 'hotspot', 'hourly', 'house
s', 'however', 'hs', 'httpcreativecommonsorglicensesbyncndabdirahman', 'httpsenhalbeegcomhttpsenhalbeegcom', 'httpsw
wwresearchgatenetpublication', 'huge', 'humanitarian', 'hungary', 'hussain', 'hybrid', 'hydrol', 'hydrology', 'ic',
'icsme', 'ictrobot', 'idcnn', 'ide', 'idowu', 'ieee', 'ii', 'iii', 'ijett', 'ijettvip', 'illustrate', 'imbalanced',
'imitate', 'immediate', 'impact', 'implement', 'implementation', 'implemented', 'implementing', 'implications', 'imp
ortant', 'improved', 'inaccuracy', 'incorrect', 'incorrectly', 'increase', 'increasingly', 'inefficient', 'ines', 'i
nevitable', 'inference', 'inform', 'informatics', 'infrastructure', 'inhabitants', 'initial', 'input', 'inputs', 'in
sight', 'insights', 'inspired', 'installed', 'instance', 'instances', 'int', 'integratedcircuit', 'intell', 'intelli
gence', 'intelligent', 'intended', 'international', 'introducing', 'involves', 'io', 'iotbased', 'issn', 'issue', 'i
ssues', 'istanbul', 'iv', 'iwssip', 'j', 'jain', 'java', 'jk', 'job', 'journal', 'js', 'jubba', 'july', 'june', 'jup
yter', 'k', 'kabir', 'kalmansvm', 'keedwell', 'keep', 'kelantan', 'kept', 'key', 'kh', 'khalaf', 'khalighisigarood
i', 'knowledge', 'krishnan', 'l', 'la', 'labeled', 'language', 'languages', 'laptoppc', 'large', 'later', 'layers',
'lead', 'learn', 'learningbased', 'leaves', 'lemmatization', 'less', 'let', 'levels', 'li', 'liang', 'license', 'lif
e', 'like', 'likely', 'linear', 'link', 'liu', 'lives', 'living', 'logic', 'long', 'loop', 'loops', 'lost', 'lotfi',
'low', 'lower', 'lowest', 'mabdirahmansimadeduso', 'machinelearningalgorithms', 'made', 'magnitude', 'maintaining',
'maintenance', 'make', 'makes', 'making', 'malaysia', 'malekian', 'manag', 'management', 'manipulate', 'mapping', 'm
aterials', 'mathematical', 'mean', 'meaning', 'means', 'meanwhile', 'member', 'mention', 'mentioned', 'message', 'me
```

```
ssages', 'meteorological', 'metrics', 'microcontroller', 'might', 'mine', 'mining', 'minutes', 'mitigate', 'mk', 'mo
bile', 'modeling', 'modem', 'modems', 'modifications', 'module', 'mogadishu', 'mohamed', 'mohd', 'monitor', 'monitor
ed', 'monitoring', 'monthly', 'moreover', 'mostly', 'movable', 'mt', 'multilayer', 'multiple', 'n', 'naive', 'nave',
'neal', 'nearly', 'necessity', 'need', 'negative', 'negatively', 'neuro', 'neurofuzzy', 'nevertheless', 'new', 'nitr
ate', 'niversitesi', 'nlp', 'none', 'nonetheless', 'nonlinear', 'nonstandard', 'normal', 'normally', 'notable', 'not
ebook', 'notice', 'notification', 'notion', 'novel', 'numbers', 'numerical', 'numerous', 'objective', 'objects', 'ob
serving', 'obtain', 'obtained', 'ocha', 'octavio', 'offering', 'offers', 'often', 'ogie', 'ones', 'onto', 'open', 'o
perates', 'operation', 'operator', 'opportunity', 'opposite', 'optimized', 'option', 'order', 'ordinary', 'origina
l', 'osman', 'outperformed', 'overall', 'overfitted', 'overfitting', 'overflowing', 'overview', 'page', 'parallel',
'park', 'part', 'partners', 'parts', 'party', 'paths', 'patidar', 'pender', 'peopledisplacedbyfloodsin', 'perceive',
'perceptron', 'perez', 'perform', 'performed', 'performing', 'periods', 'perspective', 'phase', 'phases', 'phenomen
a', 'phenomenon', 'phone', 'pic', 'picfa', 'plan', 'platform', 'play', 'plays', 'pooling', 'popular', 'popularity',
'populated', 'population', 'portion', 'positioning', 'positive', 'potential', 'pp', 'prabhakar', 'pradhan', 'prealar
ming', 'precipitation', 'precision', 'predicative', 'predictions', 'preferred', 'presence', 'present', 'presenting',
'presents', 'preserving', 'prevalent', 'preventing', 'prevention', 'prevents', 'previously', 'private', 'probabilist
ic', 'problems', 'procedure', 'proceedings', 'processes', 'processing', 'produce', 'proficient', 'profile', 'profile
s', 'programing', 'programming', 'programs', 'prominent', 'proper', 'properties', 'proportion', 'propose', 'protecti
on', 'proved', 'provide', 'provides', 'providing', 'public', 'publication', 'publications', 'purposes', 'put', 'pyth
on', 'q', 'qualitatively', 'r', 'rabczuk', 'rahmati', 'rains', 'rainwater', 'raise', 'ramprasad', 'randomforest', 'r
ange', 'rapid', 'rapidly', 'rd', 'reached', 'readings', 'reallife', 'realtime', 'reason', 'recall', 'received', 'rec
eiving', 'recognized', 'reduce', 'redundant', 'regarding', 'region', 'regression', 'relationships', 'reliable', 'rel
ied', 'remarks', 'remove', 'removing', 'representation', 'requested', 'require', 'required', 'researchers', 'researc
hes', 'reservoir', 'resist', 'resour', 'respect', 'respectively', 'response', 'resulted', 'reviewing', 'rise', 'rive
rflooding', 'riverside', 'robotics', 'robust', 'rodriguez', 'role', 'romo', 'root', 'run', 'safety', 'said', 'sajedi
hosseini', 'sankaranarayanan', 'satish', 'savic', 'scalable', 'scale', 'scenario', 'scholarly', 'scholars', 'sci',
'science', 'scientific', 'scientifically', 'scientists', 'scope', 'second', 'secondly', 'section', 'sections', 'se
e', 'seems', 'select', 'selected', 'selection', 'selflearning', 'semiarid', 'send', 'sending', 'sense', 'sensor', 's
ensors', 'sentences', 'sentiment', 'september', 'sequences', 'serdaroglu', 'series', 'service', 'set', 'setting', 's
eventh', 'severe', 'severity', 'shabelle', 'share', 'sharing', 'sharp', 'sheila', 'shin', 'shortly', 'showed', 'shri
nkage', 'sight', 'signals', 'sim', 'simad', 'simcard', 'similar', 'similarly', 'simonovic', 'simple', 'simplicity',
'simplify', 'simulate', 'simulating', 'since', 'singapore', 'single', 'siti', 'situation', 'size', 'sms', 'soft', 's
oftware', 'solution', 'somalia', 'sort', 'source', 'southern', 'special', 'specialized', 'springer', 'squared', 'sta
ble', 'stage', 'stand', 'state', 'statistical', 'statistics', 'stats', 'stemming', 'steps', 'stimulationofflood', 's
trategy', 'streamflow', 'strong', 'structural', 'structured', 'subfield', 'subsampling', 'subscription', 'subsequenc
e', 'subsequent', 'subsequently', 'succeeds', 'successful', 'successfully', 'sufficient', 'suitable', 'summarize',
'supervised', 'suppliers', 'support', 'supporting', 'suppression', 'surpassed', 'surveillance', 'susceptible', 'sust
ainable', 'switzerland', 'systematic', 'table', 'tables', 'tackle', 'taken', 'takes', 'taking', 'tar', 'target', 'ta
sk', 'tasks', 'team', 'technique', 'techniques', 'technologies', 'teknik', 'term', 'terms', 'text', 'th', 'thick',
'thinternational', 'third', 'tiaeiaf', 'tihany', 'title', 'tiwari', 'tokenized', 'tokenswords', 'tolerance', 'tool',
'tools', 'tough', 'town', 'tracking', 'trained', 'training', 'transfer', 'transmitted', 'true', 'ttl', 'turkey', 'un
derstand', 'unexpected', 'unhcrled', 'unit', 'units', 'unnecessary', 'unpublished', 'unrealistic', 'unsuitable', 'up
```

```
coming', 'upload', 'uploaded', 'urban', 'urls', 'usage', 'useful', 'user', 'usual', 'usually', 'utilization', 'utilize', 'utilized', 'v', 'valuable', 'value', 'values', 'variations', 'variety', 'varkonyikoczy', 'vary', 'vector', 'vectors', 'vendor', 'verify', 'verstaevel', 'victims', 'video', 'videos', 'view', 'vision', 'voltage', 'volume', 'vrkonyikczy', 'vulnerability', 'wadanigmailcom', 'warning', 'watershed', 'wavelet', 'waveletbootstrapann', 'wbann', 'weather', 'weka', 'whilst', 'widely', 'window', 'windows', 'wireless', 'word', 'words', 'wordvec', 'work', 'works', 'write', 'written', 'wu', 'x', 'xia', 'xing', 'xu', 'year', 'years', 'z', 'zadeh', 'zaiton', 'zone', 'zones']

--- Common Words ---
Total Common Words: 263
All Common Words: ['abstract', 'according', 'accuracy', 'accurate', 'across', 'actual', 'addition', 'additionally', 'affected', 'alert', 'algorithm', 'algorithms', 'also', 'although', 'amir', 'analyze', 'another', 'application', 'approach', 'architecture', 'area', 'areas', 'authorities', 'available', 'avoided', 'b', 'basic', 'beginning', 'besides', 'best', 'better', 'blue', 'built', 'c', 'calculation', 'called', 'cause', 'certain', 'changed', 'chau', 'collected', 'color', 'combination', 'come', 'common', 'commonly', 'compare', 'compared', 'compatible', 'complex', 'consequences', 'containing', 'contains', 'contributed', 'corresponding', 'could', 'create', 'data', 'dataset', 'datasets', 'date', 'days', 'decision', 'decreasing', 'deep', 'demonstrates', 'depending', 'design', 'designed', 'detect', 'detection', 'detects', 'determine', 'determined', 'developed', 'developing', 'different', 'dimensionality', 'directly', 'disasters', 'due', 'e', 'effect', 'effective', 'electronic', 'ensemble', 'etc', 'event', 'every', 'example', 'extract', 'extreme', 'fact', 'feature', 'features', 'field', 'fig', 'finally', 'first', 'flood', 'flooding', 'floods', 'following', 'forecast', 'forecasting', 'form', 'function', 'get', 'given', 'gives', 'globe', 'help', 'helpful', 'high', 'highest', 'human', 'hydrological', 'image', 'images', 'improve', 'improvement', 'improving', 'included', 'including', 'increased', 'increasing', 'information', 'introduction', 'inundation', 'keywords', 'kisi', 'know', 'known', 'kw', 'labels', 'last', 'learning', 'level', 'locations', 'loss', 'lot', 'machine', 'main', 'manner', 'many', 'matrix', 'may', 'measure', 'measures', 'method', 'methodology', 'methods', 'ml', 'model', 'models', 'modern', 'mosavi', 'must', 'natural', 'needed', 'network', 'networks', 'neural', 'next', 'noise', 'number', 'observe', 'observed', 'occur', 'occurred', 'one', 'output', 'p', 'paper', 'papers', 'parameters', 'particular', 'past', 'people', 'performance', 'place', 'point', 'positions', 'possible', 'power', 'predict', 'prediction', 'previous', 'process', 'proposed', 'purpose', 'rainfall', 'random', 'reads', 'real', 'recent', 'red', 'reduced', 'references', 'regard', 'regions', 'related', 'relevant', 'removed', 'reported', 'research', 'rest', 'result', 'results', 'retrieved', 'review', 'risk', 'river', 'rivers', 'sent', 'separate', 'sequential', 'sets', 'several', 'shape', 'short', 'showing', 'shows', 'significant', 'slightly', 'solutions', 'stated', 'step', 'study', 'system', 'systems', 'take', 'technology', 'testing', 'therefore', 'things', 'three', 'thus', 'time', 'times', 'total', 'tree', 'trees', 'trends', 'two', 'type', 'university', 'updated', 'us', 'use', 'used', 'uses', 'using', 'various', 'visualize', 'water', 'way', 'well', 'whereas', 'whole', 'within', 'world', 'would']

Cosine Similarity Score between Document 1 and Document 2: 0.4271
```

A **cosine similarity score of 0.4271** indicates that the two documents have a **moderate level of similarity** in terms of their textual content. This means that while the documents share some common terms and themes, they also have **significant differences** in vocabulary and content.

References to the Research papers:

Paper1:
[1] K. G. Sharma, R. Kaur, H. Dalal, R. Laddha, and T. Thind, “Flood Detection Using Machine Learning,” *Proc. KILBY 100 7th Int. Conf. Comput. Sci. (ICCS 2023)*, May 5, 2023. Available:
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4491445

Paper2:
[2] Hashi, A. O., Abdirahman, A. A., Abdi, M. A., & Rodriguez, O. E. R. (2021, May). *A real-time flood detection system based on machine learning algorithms with emphasis on deep learning.* International Journal of Engineering Trends and Technology, 69(5), 249-256.
https://doi.org/10.14445/22315381/IJETT-V69I5P232