Information Retrieval Assignment 1

Group ID: 26

Group Members Name with Student ID:

1. KARTHIKEYAN J - 2024AA05372

2. JANGALE SAVEDANA SUBHASH PRATIBHA - 2024AA05187

3. GANAPATHY SUBRAMANIAN S - 2024AA05188

4. ANANDAN A - 2024AA05269

Problem Statement

Designing a Text Search and Query Correction System using Levenshtein Edit Distance algorithm for Medical Documents

1. Import and download the required libraries

In [4]:

```
import re
import os
from collections import defaultdict
import nltk
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize
import PyPDF2
import pandas as pd
from docx import Document  # Must be imported!
import os
from IPython.display import display, Markdown

# Download NLTK resources
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
```

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

Out[4]: True

Global variables and NLP setup

In [6]:

```python
# Global variables
inverted_index = defaultdict(set)
all_terms = set()
documents = []
doc_metadata = []

# NLP setup
stop_words = set(stopwords.words('english'))
lemmatizer = WordNetLemmatizer()
```

1. Data Preprocessing

```
a) Load documents from the directory provided.
b) Preprocess each document. Will display terms, unique terms
and sample terms from each document.Remove all punctuation,
numbers, and special    characters from the dataset.
```

2. create non-positional inverted index in descending order.

In [8]:

```python
#  User defined functions to remove all punctuation, numbers, and special char
#  Apply lemmatization techniques to convert words to their base or root forms
def preprocess_text(text):
    """Full preprocessing with intermediate steps"""
    print("\n=== ORIGINAL TEXT (SAMPLE) ===")
    print(text[:200] + "...\n" if len(text) > 200 else text)

    # 1. Clean text
    cleaned = re.sub(r'[^a-zA-Z0-9\s]', '', text.lower())
    print("=== AFTER CLEANING ===")
    print(cleaned[:200] + "...\n" if len(cleaned) > 200 else cleaned)

    # 2. Tokenization
    tokens = word_tokenize(cleaned)
    print(f"TOKENS ({len(tokens)}):", tokens[:30], "...\n")

    # 3. Stopword removal
    filtered = [w for w in tokens if w not in stop_words and len(w) > 2]
    print(f"AFTER STOPWORD REMOVAL ({len(filtered)}):", filtered[:30], "...\n"

    # 4. Lemmatization
    lemmatized = [lemmatizer.lemmatize(w) for w in filtered]
    print(f"FINAL PROCESSED TERMS ({len(lemmatized)}):", lemmatized[:30], "...

    return lemmatized
```

In [9]:

```
# User defined functions to read different types of files from a directory.
def read_txt(file_path):
    """Read text file"""
    encodings = ['utf-8', 'latin-1', 'windows-1252']
    for encoding in encodings:
        with open(file_path, 'r', encoding=encoding) as f:
                return f.read()

def read_pdf(file_path):
    """Read PDF file"""
    text = ""
    with open(file_path, 'rb') as file:
            reader = PyPDF2.PdfReader(file)
            for page in reader.pages:
                text += page.extract_text()
    return text

def read_csv(file_path):
    """Read CSV file"""
    encodings = ['utf-8', 'latin-1', 'windows-1252']
    for encoding in encodings:
       df = pd.read_csv(file_path, encoding=encoding)
       return ' '.join(df.select_dtypes(include=['object']).astype(str).values

def read_excel(file_path):
    """Read Excel file"""
    df = pd.read_excel(file_path)
    return ' '.join(df.select_dtypes(include=['object']).astype(str).values.fl

def read_docx(file_path):
    """Read Word DOCX file"""
    doc = Document(file_path)
    return '\n'.join([para.text for para in doc.paragraphs])
```

In [10]:

```
def load_documents(directory):
    """Load documents from directory and build index"""
    global documents, doc_metadata, inverted_index, all_terms
    document_metadata = []

    if not os.path.exists(directory):
        raise FileNotFoundError(f"Directory not found: {directory}")
```

```
print(f"Loading documents from: {directory}")

for root, _, files in os.walk(directory):
    for file in files:
        file_path = os.path.join(root, file)
        try:
            if file.endswith('.txt'):
                text = read_txt(file_path)
            elif file.endswith('.pdf'):
                text = read_pdf(file_path)
            elif file.endswith('.csv'):
                text = read_csv(file_path)
            elif file.endswith(('.xls', '.xlsx')):
                text = read_excel(file_path)
            elif file.endswith('.docx'):
                text = read_docx(file_path)
            else:
                continue

            if text.strip():
                doc_id = len(documents)
                documents.append(text)
                doc_metadata.append({
                    'file_name': file,
                    'file_path': file_path
                })

                print(f"\n===============Loading: {file}==================
                # Add to index
                terms = preprocess_text(text)  # preprocessing each
                for term in terms:
                    inverted_index[term].add(doc_id) # inverted index crea
                    all_terms.add(term)

                 # Store metadata - PROPERLY INDENTED
                document_metadata.append({
                    'doc_id': doc_id,
                    'filename': file,
                    'filetype': os.path.splitext(file)[1],
                    'terms': len(terms),
                    'unique_terms': len(set(terms))
                })

                # Display file processing info
                print(f"\n - {file} ({document_metadata[-1]['filetype']})"
                print(f"  - Total Terms: {document_metadata[-1]['terms']}"
                print(f"  - Unique terms: {document_metadata[-1]['unique_t
                print(f"  - Sample unique terms: {list(set(terms))[:5]}...

                print(f"\n Loaded: {file}")
        except Exception as e:
            print(f"Error processing {file}: {str(e)}")
    print(f"\nTOTAL SUMMARY")
    print(f"\nTotal documents loaded: {len(documents)}")
    print(f"Unique terms in index: {len(all_terms)}")

  #2)   # **********************Show most frequent terms - Sorted index creati
    top_terms = sorted(inverted_index.items(),
                      key=lambda x: len(x[1]),
                      reverse=True)[:5]

    print("\n*************Sorted index*****************")
    print("\nTop 5 terms:")
    for term, doc_ids in top_terms:
        print(f"  {term}: appears in {(doc_ids)} documents")

    return inverted_index, document_metadata
```

In [11]:

```
 # Load documents
directory = "D:/AIML/IR/Assignment/medical_documents/"
print(directory)
path = os.path.abspath(directory)
print(f"\n===============OUTPUT=============================")
inverted_index, document_metadata  =  load_documents(path) #load all the docum
```

```
D:/AIML/IR/Assignment/medical_documents/

===============OUTPUT=============================
Loading documents from: D:\AIML\IR\Assignment\medical_documents

===============Loading: Cardio.pdf=====================================

=== ORIGINAL TEXT (SAMPLE) ===
Cardiologist-Level Arrhythmia Detection with Convolutional Neural Networks
Pranav Rajpurkar�PRANAVSR @CS.STANFORD .EDU
Awni Y. Hannun�AWNI @CS.STANFORD .EDU
Masoumeh Haghpanahi MHAGHPANAHI @IRHYTHMTEC...

=== AFTER CLEANING ===
cardiologistlevel arrhythmia detection with convolutional neural networks
pranav rajpurkarpranavsr csstanford edu
awni y hannunawni csstanford edu
masoumeh haghpanahi mhaghpanahi irhythmtech com
codie...

TOKENS (4517): ['cardiologistlevel', 'arrhythmia', 'detection', 'with', 'convol
utional', 'neural', 'networks', 'pranav', 'rajpurkarpranavsr', 'csstanford', 'e
du', 'awni', 'y', 'hannunawni', 'csstanford', 'edu', 'masoumeh', 'haghpanahi',
'mhaghpanahi', 'irhythmtech', 'com', 'codie', 'bourn', 'cbourn', 'irhythmtech',
'com', 'andrew', 'y', 'ng', 'ang'] ...

AFTER STOPWORD REMOVAL (2853): ['cardiologistlevel', 'arrhythmia', 'detection',
'convolutional', 'neural', 'networks', 'pranav', 'rajpurkarpranavsr', 'csstanfo
rd', 'edu', 'awni', 'hannunawni', 'csstanford', 'edu', 'masoumeh', 'haghpanah
i', 'mhaghpanahi', 'irhythmtech', 'com', 'codie', 'bourn', 'cbourn', 'irhythmte
ch', 'com', 'andrew', 'ang', 'csstanford', 'edu', 'abstract', 'develop'] ...

FINAL PROCESSED TERMS (2853): ['cardiologistlevel', 'arrhythmia', 'detection',
'convolutional', 'neural', 'network', 'pranav', 'rajpurkarpranavsr', 'csstanfor
d', 'edu', 'awni', 'hannunawni', 'csstanford', 'edu', 'masoumeh', 'haghpanahi',
'mhaghpanahi', 'irhythmtech', 'com', 'codie', 'bourn', 'cbourn', 'irhythmtech',
'com', 'andrew', 'ang', 'csstanford', 'edu', 'abstract', 'develop'] ...

 - Cardio.pdf (.pdf)
  - Total Terms: 2853
  - Unique terms: 1264
  - Sample unique terms: ['390', '4827', 'observation', 'challenge', 'proble
m']...

 Loaded: Cardio.pdf

===============Loading: Cardiovascular  Pulmonary.tx
t======================================

=== ORIGINAL TEXT (SAMPLE) ===
Cardiovascular / Pulmonary

Sample Name: Acute Inferior Myocardial Infarction
```

```
Description: Patient presents with a chief complaint of chest pain admitted to
Coronary Care Unit due to acute inferior m...

=== AFTER CLEANING ===
cardiovascular  pulmonary

sample name acute inferior myocardial infarction

description patient presents with a chief complaint of chest pain admitted to c
oronary care unit due to acute inferior myoc...

TOKENS (628): ['cardiovascular', 'pulmonary', 'sample', 'name', 'acute', 'infer
ior', 'myocardial', 'infarction', 'description', 'patient', 'presents', 'with',
'a', 'chief', 'complaint', 'of', 'chest', 'pain', 'admitted', 'to', 'coronary',
'care', 'unit', 'due', 'to', 'acute', 'inferior', 'myocardial', 'infarction',
'medical'] ...

AFTER STOPWORD REMOVAL (397): ['cardiovascular', 'pulmonary', 'sample', 'name',
'acute', 'inferior', 'myocardial', 'infarction', 'description', 'patient', 'pre
sents', 'chief', 'complaint', 'chest', 'pain', 'admitted', 'coronary', 'care',
'unit', 'due', 'acute', 'inferior', 'myocardial', 'infarction', 'medical', 'tra
nscription', 'sample', 'report', 'chief', 'complaint'] ...

FINAL PROCESSED TERMS (397): ['cardiovascular', 'pulmonary', 'sample', 'name',
'acute', 'inferior', 'myocardial', 'infarction', 'description', 'patient', 'pre
sent', 'chief', 'complaint', 'chest', 'pain', 'admitted', 'coronary', 'care',
'unit', 'due', 'acute', 'inferior', 'myocardial', 'infarction', 'medical', 'tra
nscription', 'sample', 'report', 'chief', 'complaint'] ...

 - Cardiovascular  Pulmonary.txt (.txt)
  - Total Terms: 397
  - Unique terms: 266
  - Sample unique terms: ['nitroglycerine', 'cardiologist', 'reviewed', 'platel
et', 'transcription']...

 Loaded: Cardiovascular  Pulmonary.txt

===============Loading: DataAnalyticsinhealthcare.pd
f======================================

=== ORIGINAL TEXT (SAMPLE) ===
See discussions, st ats, and author pr ofiles f or this public ation at : http
s://www .researchgate.ne t/public ation/351792114
Data Analytics in Healthcare Systems – Principles, Challenges, and
Appli...

=== AFTER CLEANING ===
see discussions st ats and author pr ofiles f or this public ation at  httpswww
researchgatene tpublic ation351792114
data analytics in healthcare systems  principles challenges and
applications
chapt...

TOKENS (8391): ['see', 'discussions', 'st', 'ats', 'and', 'author', 'pr', 'ofil
```

```
es', 'f', 'or', 'this', 'public', 'ation', 'at', 'httpswww', 'researchgatene',
'tpublic', 'ation351792114', 'data', 'analytics', 'in', 'healthcare', 'system
s', 'principles', 'challenges', 'and', 'applications', 'chapt', 'er', 'may']
...

AFTER STOPWORD REMOVAL (5366): ['see', 'discussions', 'ats', 'author', 'ofile
s', 'public', 'ation', 'httpswww', 'researchgatene', 'tpublic', 'ation35179211
4', 'data', 'analytics', 'healthcare', 'systems', 'principles', 'challenges',
'applications', 'chapt', 'may', '2021', 'doi', '10120197810031852461', 'citatio
ns', '2reads', '10647', 'author', 'sug', 'anthi', 'galg'] ...

FINAL PROCESSED TERMS (5366): ['see', 'discussion', 'at', 'author', 'ofiles',
'public', 'ation', 'httpswww', 'researchgatene', 'tpublic', 'ation351792114',
'data', 'analytics', 'healthcare', 'system', 'principle', 'challenge', 'applica
tion', 'chapt', 'may', '2021', 'doi', '10120197810031852461', 'citation', '2rea
ds', '10647', 'author', 'sug', 'anthi', 'galg'] ...

 - DataAnalyticsinhealthcare.pdf (.pdf)
  - Total Terms: 5366
  - Unique terms: 1843
  - Sample unique terms: ['uplo', 'equipment', 'recommender', 'challenge', 'mer
ging']...

 Loaded: DataAnalyticsinhealthcare.pdf

===============Loading: gender-differences-arteries.pd
f======================================

=== ORIGINAL TEXT (SAMPLE) ===
Adrien Desjardins2
1R o y a lF r e eH o s p i t a l ,L o n d o n ,U n i t e dK i n g d o m ;2Unive
rsity College
London, London, United Kingdom
BACKGROUND In situ fenestration (ISF) is an attractive op...

=== AFTER CLEANING ===
adrien desjardins2
1r o y a lf r e eh o s p i t a l l o n d o n u n i t e dk i n g d o m 2universi
ty college
london london united kingdom
background in situ fenestration isf is an attractive option to...

TOKENS (1133): ['adrien', 'desjardins2', '1r', 'o', 'y', 'a', 'lf', 'r', 'e',
'eh', 'o', 's', 'p', 'i', 't', 'a', 'l', 'l', 'o', 'n', 'd', 'o', 'n', 'u',
'n', 'i', 't', 'e', 'dk', 'i'] ...

AFTER STOPWORD REMOVAL (666): ['adrien', 'desjardins2', '2university', 'colleg
e', 'london', 'london', 'united', 'kingdom', 'background', 'situ', 'fenestratio
n', 'isf', 'attractive', 'option', 'preserve', 'aortic', 'branch', 'patency',
'fenestrated', 'endovascular', 'aorticrepair', 'fevar', 'complex', 'aortic', 'a
neurysms', 'although', 'prefenestrated', 'grafts', 'suitable', 'common'] ...

FINAL PROCESSED TERMS (666): ['adrien', 'desjardins2', '2university', 'colleg
e', 'london', 'london', 'united', 'kingdom', 'background', 'situ', 'fenestratio
```

```
n', 'isf', 'attractive', 'option', 'preserve', 'aortic', 'branch', 'patency',
'fenestrated', 'endovascular', 'aorticrepair', 'fevar', 'complex', 'aortic', 'a
neurysm', 'although', 'prefenestrated', 'graft', 'suitable', 'common'] ...

 - gender-differences-arteries.pdf (.pdf)
  - Total Terms: 666
  - Unique terms: 467
  - Sample unique terms: ['arte', 'siroli', 'clinical', 'still', 'formulatio
n']...

 Loaded: gender-differences-arteries.pdf

===============Loading: in-hospital-mortality-trends-by-health-category.cs
v======================================

=== ORIGINAL TEXT (SAMPLE) ===
05/2018 Anxiety Ambulatory Surgery 09/2018 Anxiety Ambulatory Surgery 10/2018 A
nxiety Ambulatory Surgery 01/2019 Anxiety Ambulatory Surgery 06/2019 Anxiety Am
bulatory Surgery 02/2020 Anxiety Ambulator...

=== AFTER CLEANING ===
052018 anxiety ambulatory surgery 092018 anxiety ambulatory surgery 102018 anxi
ety ambulatory surgery 012019 anxiety ambulatory surgery 062019 anxiety ambulat
ory surgery 022020 anxiety ambulatory surg...

TOKENS (10907): ['052018', 'anxiety', 'ambulatory', 'surgery', '092018', 'anxie
ty', 'ambulatory', 'surgery', '102018', 'anxiety', 'ambulatory', 'surgery', '01
2019', 'anxiety', 'ambulatory', 'surgery', '062019', 'anxiety', 'ambulatory',
'surgery', '022020', 'anxiety', 'ambulatory', 'surgery', '032020', 'anxiety',
'ambulatory', 'surgery', '042020', 'anxiety'] ...

AFTER STOPWORD REMOVAL (10907): ['052018', 'anxiety', 'ambulatory', 'surgery',
'092018', 'anxiety', 'ambulatory', 'surgery', '102018', 'anxiety', 'ambulator
y', 'surgery', '012019', 'anxiety', 'ambulatory', 'surgery', '062019', 'anxiet
y', 'ambulatory', 'surgery', '022020', 'anxiety', 'ambulatory', 'surgery', '032
020', 'anxiety', 'ambulatory', 'surgery', '042020', 'anxiety'] ...

FINAL PROCESSED TERMS (10907): ['052018', 'anxiety', 'ambulatory', 'surgery',
'092018', 'anxiety', 'ambulatory', 'surgery', '102018', 'anxiety', 'ambulator
y', 'surgery', '012019', 'anxiety', 'ambulatory', 'surgery', '062019', 'anxiet
y', 'ambulatory', 'surgery', '022020', 'anxiety', 'ambulatory', 'surgery', '032
020', 'anxiety', 'ambulatory', 'surgery', '042020', 'anxiety'] ...

 - in-hospital-mortality-trends-by-health-category.csv (.csv)
  - Total Terms: 10907
  - Unique terms: 123
  - Sample unique terms: ['92019', '112018', '102021', '62020', '62021']...

 Loaded: in-hospital-mortality-trends-by-health-category.csv

===============Loading: Medical Specialty.tx
t======================================

=== ORIGINAL TEXT (SAMPLE) ===
Medical Specialty:
Cardiovascular / Pulmonary

Sample Name: Abnormal Echocardiogram

Description: Abnormal echocardiogram findings and followup. Shortness of breat
h, congestive heart failure, and valv...

=== AFTER CLEANING ===
medical specialty
cardiovascular  pulmonary

sample name abnormal echocardiogram

description abnormal echocardiogram findings and followup shortness of breath c
ongestive heart failure and valvular in...

TOKENS (567): ['medical', 'specialty', 'cardiovascular', 'pulmonary', 'sample',
'name', 'abnormal', 'echocardiogram', 'description', 'abnormal', 'echocardiogra
m', 'findings', 'and', 'followup', 'shortness', 'of', 'breath', 'congestive',
'heart', 'failure', 'and', 'valvular', 'insufficiency', 'the', 'patient', 'comp
lains', 'of', 'shortness', 'of', 'breath'] ...

AFTER STOPWORD REMOVAL (379): ['medical', 'specialty', 'cardiovascular', 'pulmo
nary', 'sample', 'name', 'abnormal', 'echocardiogram', 'description', 'abnorma
l', 'echocardiogram', 'findings', 'followup', 'shortness', 'breath', 'congestiv
e', 'heart', 'failure', 'valvular', 'insufficiency', 'patient', 'complains', 's
hortness', 'breath', 'worsening', 'patient', 'underwent', 'echocardiogram', 'sh
ows', 'severe'] ...

FINAL PROCESSED TERMS (379): ['medical', 'specialty', 'cardiovascular', 'pulmon
ary', 'sample', 'name', 'abnormal', 'echocardiogram', 'description', 'abnorma
l', 'echocardiogram', 'finding', 'followup', 'shortness', 'breath', 'congestiv
e', 'heart', 'failure', 'valvular', 'insufficiency', 'patient', 'complains', 's
hortness', 'breath', 'worsening', 'patient', 'underwent', 'echocardiogram', 'sh
ow', 'severe'] ...

 - Medical Specialty.txt (.txt)
  - Total Terms: 379
  - Unique terms: 237
  - Sample unique terms: ['ventricular', 'systolic', 'reviewed', 'reason', 'atr
aumatic']...

 Loaded: Medical Specialty.txt

===============Loading: Medical Specialty_Gastro.pd
f======================================

=== ORIGINAL TEXT (SAMPLE) ===
Medical Specialty:
Gastroenterology

Sample Name:  Colonoscopy & Polypectomy - 3
```

```
Description:  Total colonoscopy with biopsy and snare polypectomy.
(Medical Transcription Sample Report)
PR...

=== AFTER CLEANING ===
medical specialty
gastroenterology

sample name  colonoscopy  polypectomy  3

description  total colonoscopy with biopsy and snare polypectomy
medical transcription sample report
preoperati...

TOKENS (336): ['medical', 'specialty', 'gastroenterology', 'sample', 'name', 'c
olonoscopy', 'polypectomy', '3', 'description', 'total', 'colonoscopy', 'with',
'biopsy', 'and', 'snare', 'polypectomy', 'medical', 'transcription', 'sample',
'report', 'preoperative', 'diagnosis', 'alternating', 'hard', 'and', 'soft', 's
tools', 'postoperative', 'diagnosis', 'sigmoid'] ...

AFTER STOPWORD REMOVAL (209): ['medical', 'specialty', 'gastroenterology', 'sam
ple', 'name', 'colonoscopy', 'polypectomy', 'description', 'total', 'colonoscop
y', 'biopsy', 'snare', 'polypectomy', 'medical', 'transcription', 'sample', 're
port', 'preoperative', 'diagnosis', 'alternating', 'hard', 'soft', 'stools', 'p
ostoperative', 'diagnosis', 'sigmoid', 'diverticulosis', 'sessile', 'polyp', 's
igmoid'] ...

FINAL PROCESSED TERMS (209): ['medical', 'specialty', 'gastroenterology', 'samp
le', 'name', 'colonoscopy', 'polypectomy', 'description', 'total', 'colonoscop
y', 'biopsy', 'snare', 'polypectomy', 'medical', 'transcription', 'sample', 're
port', 'preoperative', 'diagnosis', 'alternating', 'hard', 'soft', 'stool', 'po
stoperative', 'diagnosis', 'sigmoid', 'diverticulosis', 'sessile', 'polyp', 'si
gmoid'] ...

 - Medical Specialty_Gastro.pdf (.pdf)
  - Total Terms: 209
  - Unique terms: 132
  - Sample unique terms: ['reaching', 'ileo', 'transcription', 'approximately',
'assessment']...

 Loaded: Medical Specialty_Gastro.pdf

===============Loading: Medical_history.doc
x=====================================

=== ORIGINAL TEXT (SAMPLE) ===
Medical Specialty:
Surgery

Sample Name: Arthroscopy & Chondroplasty

Description: Diagnostic arthroscopy with partial chondroplasty of patella, late
ral retinacular release, and open tibial tubercle t...
```

```
=== AFTER CLEANING ===
medical specialty
surgery

sample name arthroscopy  chondroplasty

description diagnostic arthroscopy with partial chondroplasty of patella latera
l retinacular release and open tibial tubercle transfe...

TOKENS (716): ['medical', 'specialty', 'surgery', 'sample', 'name', 'arthroscop
y', 'chondroplasty', 'description', 'diagnostic', 'arthroscopy', 'with', 'parti
al', 'chondroplasty', 'of', 'patella', 'lateral', 'retinacular', 'release', 'an
d', 'open', 'tibial', 'tubercle', 'transfer', 'with', 'fixation', 'of', 'two',
'45', 'mm', 'cannulated'] ...

AFTER STOPWORD REMOVAL (417): ['medical', 'specialty', 'surgery', 'sample', 'na
me', 'arthroscopy', 'chondroplasty', 'description', 'diagnostic', 'arthroscop
y', 'partial', 'chondroplasty', 'patella', 'lateral', 'retinacular', 'release',
'open', 'tibial', 'tubercle', 'transfer', 'fixation', 'two', 'cannulated', 'scr
ews', 'gradeiv', 'chondromalacia', 'patella', 'patellofemoral', 'malalignment',
'syndrome'] ...

FINAL PROCESSED TERMS (417): ['medical', 'specialty', 'surgery', 'sample', 'nam
e', 'arthroscopy', 'chondroplasty', 'description', 'diagnostic', 'arthroscopy',
'partial', 'chondroplasty', 'patella', 'lateral', 'retinacular', 'release', 'op
en', 'tibial', 'tubercle', 'transfer', 'fixation', 'two', 'cannulated', 'scre
w', 'gradeiv', 'chondromalacia', 'patella', 'patellofemoral', 'malalignment',
'syndrome'] ...

 - Medical_history.docx (.docx)
  - Total Terms: 417
  - Unique terms: 249
  - Sample unique terms: ['drilled', 'abcd', '325', 'transcription', 'approxima
tely']...

 Loaded: Medical_history.docx

===============Loading: mtsamples.csv====================================

=== ORIGINAL TEXT (SAMPLE) ===
 A 23-year-old white female presents with complaint of allergies.  Allergy / Im
munology  Allergic Rhinitis  SUBJECTIVE:,  This 23-year-old white female presen
ts with complaint of allergies.  She used ...

=== AFTER CLEANING ===
 a 23yearold white female presents with complaint of allergies  allergy  immuno
logy  allergic rhinitis  subjective  this 23yearold white female presents with
complaint of allergies  she used to have a...

TOKENS (68702): ['a', '23yearold', 'white', 'female', 'presents', 'with', 'comp
laint', 'of', 'allergies', 'allergy', 'immunology', 'allergic', 'rhinitis', 'su
bjective', 'this', '23yearold', 'white', 'female', 'presents', 'with', 'complai
nt', 'of', 'allergies', 'she', 'used', 'to', 'have', 'allergies', 'when', 'sh
e'] ...
```

```
AFTER STOPWORD REMOVAL (49840): ['23yearold', 'white', 'female', 'presents', 'c
omplaint', 'allergies', 'allergy', 'immunology', 'allergic', 'rhinitis', 'subje
ctive', '23yearold', 'white', 'female', 'presents', 'complaint', 'allergies',
'used', 'allergies', 'lived', 'seattle', 'thinks', 'worse', 'past', 'tried', 'c
laritin', 'zyrtec', 'worked', 'short', 'time'] ...

FINAL PROCESSED TERMS (49840): ['23yearold', 'white', 'female', 'present', 'com
plaint', 'allergy', 'allergy', 'immunology', 'allergic', 'rhinitis', 'subjectiv
e', '23yearold', 'white', 'female', 'present', 'complaint', 'allergy', 'used',
'allergy', 'lived', 'seattle', 'think', 'worse', 'past', 'tried', 'claritin',
'zyrtec', 'worked', 'short', 'time'] ...

 - mtsamples.csv (.csv)
  - Total Terms: 49840
  - Unique terms: 4646
  - Sample unique terms: ['tsh', 'observation', 'hydrochlorothiazide', 'trigon
e', 'ethibond']...

 Loaded: mtsamples.csv

===============Loading: mtsamples.xlsx=====================================

=== ORIGINAL TEXT (SAMPLE) ===
 A 23-year-old white female presents with complaint of allergies.  Allergy / Im
munology  Allergic Rhinitis  SUBJECTIVE:,  This 23-year-old white female presen
ts with complaint of allergies.  She used ...

=== AFTER CLEANING ===
 a 23yearold white female presents with complaint of allergies  allergy  immuno
logy  allergic rhinitis  subjective  this 23yearold white female presents with
complaint of allergies  she used to have a...

TOKENS (44892): ['a', '23yearold', 'white', 'female', 'presents', 'with', 'comp
laint', 'of', 'allergies', 'allergy', 'immunology', 'allergic', 'rhinitis', 'su
bjective', 'this', '23yearold', 'white', 'female', 'presents', 'with', 'complai
nt', 'of', 'allergies', 'she', 'used', 'to', 'have', 'allergies', 'when', 'sh
e'] ...

AFTER STOPWORD REMOVAL (26030): ['23yearold', 'white', 'female', 'presents', 'c
omplaint', 'allergies', 'allergy', 'immunology', 'allergic', 'rhinitis', 'subje
ctive', '23yearold', 'white', 'female', 'presents', 'complaint', 'allergies',
'used', 'allergies', 'lived', 'seattle', 'thinks', 'worse', 'past', 'tried', 'c
laritin', 'zyrtec', 'worked', 'short', 'time'] ...

FINAL PROCESSED TERMS (26030): ['23yearold', 'white', 'female', 'present', 'com
plaint', 'allergy', 'allergy', 'immunology', 'allergic', 'rhinitis', 'subjectiv
e', '23yearold', 'white', 'female', 'present', 'complaint', 'allergy', 'used',
'allergy', 'lived', 'seattle', 'think', 'worse', 'past', 'tried', 'claritin',
'zyrtec', 'worked', 'short', 'time'] ...

 - mtsamples.xlsx (.xlsx)
  - Total Terms: 26030
  - Unique terms: 4647
  - Sample unique terms: ['tsh', 'observation', 'hydrochlorothiazide', 'trigon
e', 'ethibond']...

 Loaded: mtsamples.xlsx

===============Loading: train.txt=====================================

=== ORIGINAL TEXT (SAMPLE) ===
4         Catheterization laboratory events and hospital outcome with direct ang
ioplasty for acute myocardial infarction To assess the safety of direct infarct
angioplasty without antecedent thrombolytic ther...

=== AFTER CLEANING ===
4         catheterization laboratory events and hospital outcome with direct ang
ioplasty for acute myocardial infarction to assess the safety of direct infarct
angioplasty without antecedent thrombolytic ther...

TOKENS (2157): ['4', 'catheterization', 'laboratory', 'events', 'and', 'hospita
l', 'outcome', 'with', 'direct', 'angioplasty', 'for', 'acute', 'myocardial',
'infarction', 'to', 'assess', 'the', 'safety', 'of', 'direct', 'infarct', 'angi
oplasty', 'without', 'antecedent', 'thrombolytic', 'therapy', 'catheterizatio
n', 'laboratory', 'and', 'hospital'] ...

AFTER STOPWORD REMOVAL (1255): ['catheterization', 'laboratory', 'events', 'hos
pital', 'outcome', 'direct', 'angioplasty', 'acute', 'myocardial', 'infarctio
n', 'assess', 'safety', 'direct', 'infarct', 'angioplasty', 'without', 'anteced
ent', 'thrombolytic', 'therapy', 'catheterization', 'laboratory', 'hospital',
'events', 'assessed', 'consecutively', 'treated', 'patients', 'infarctions', 'i
nvolving', 'left'] ...

FINAL PROCESSED TERMS (1255): ['catheterization', 'laboratory', 'event', 'hospi
tal', 'outcome', 'direct', 'angioplasty', 'acute', 'myocardial', 'infarction',
'assess', 'safety', 'direct', 'infarct', 'angioplasty', 'without', 'anteceden
t', 'thrombolytic', 'therapy', 'catheterization', 'laboratory', 'hospital', 'ev
ent', 'assessed', 'consecutively', 'treated', 'patient', 'infarction', 'involvi
ng', 'left'] ...

 - train.txt (.txt)
  - Total Terms: 1255
  - Unique terms: 629
  - Sample unique terms: ['twentynine', 'clinical', 'nature', 'terminal', 'prob
lem']...

 Loaded: train.txt

===============Loading: Train_Data.txt=====================================

=== ORIGINAL TEXT (SAMPLE) ===
###24293578
OBJECTIVE        To investigate the efficacy of 6 weeks of daily low-dose oral
prednisolone in improving pain , mobility , and systemic low-grade inflammation
in the short term and whether the ef...

=== AFTER CLEANING ===
24293578
objective       to investigate the efficacy of 6 weeks of daily lowdose oral p
rednisolone in improving pain  mobility  and systemic lowgrade inflammation in
the short term and whether the effect wo...

TOKENS (5539): ['24293578', 'objective', 'to', 'investigate', 'the', 'efficac
y', 'of', '6', 'weeks', 'of', 'daily', 'lowdose', 'oral', 'prednisolone', 'in',
'improving', 'pain', 'mobility', 'and', 'systemic', 'lowgrade', 'inflammation',
'in', 'the', 'short', 'term', 'and', 'whether', 'the', 'effect'] ...

AFTER STOPWORD REMOVAL (3453): ['24293578', 'objective', 'investigate', 'effica
cy', 'weeks', 'daily', 'lowdose', 'oral', 'prednisolone', 'improving', 'pain',
'mobility', 'systemic', 'lowgrade', 'inflammation', 'short', 'term', 'whether',
'effect', 'would', 'sustained', 'weeks', 'older', 'adults', 'moderate', 'sever
e', 'knee', 'osteoarthritis', 'methods', 'total'] ...

FINAL PROCESSED TERMS (3453): ['24293578', 'objective', 'investigate', 'efficac
y', 'week', 'daily', 'lowdose', 'oral', 'prednisolone', 'improving', 'pain', 'm
obility', 'systemic', 'lowgrade', 'inflammation', 'short', 'term', 'whether',
'effect', 'would', 'sustained', 'week', 'older', 'adult', 'moderate', 'severe',
'knee', 'osteoarthritis', 'method', 'total'] ...

 - Train_Data.txt (.txt)
  - Total Terms: 3453
  - Unique terms: 1344
  - Sample unique terms: ['challenge', 'problem', 'joint', 'formulation', 'mone
tary']...

 Loaded: Train_Data.txt

TOTAL SUMMARY

Total documents loaded: 12
Unique terms in index: 8116

**************Sorted index*****************

Top 5 terms:
  patient: appears in {0, 1, 2, 3, 5, 6, 7, 8, 9, 10, 11} documents
  disease: appears in {0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11} documents
  risk: appears in {1, 2, 3, 5, 6, 7, 8, 9, 10, 11} documents
  heart: appears in {0, 1, 2, 3, 5, 8, 9, 10, 11} documents
  also: appears in {0, 2, 3, 5, 7, 8, 9, 10, 11} documents
```

Justification:

1. Above function first load all the documents with different file extensions from the given directory

2. Preprocessing is done for each document. a) ORIGINAL TEXT (SAMPLE) shows few line from the document b) AFTER CLEANING shows text after removing special charaters, then converting all text to lowercase c)

TOKENS (stream of text converted into smaller units called tokens) are extracted from each document d) AFTER STOPWORD REMOVAL removes all the stopwords(common English words to be excluded from each document e) FINAL PROCESSED TERMS shows all the words after Lemmatization(reduce words to their base or dictionary form).

3. Sorted index is created after preprocessing step. All the indexed are sorted and top 5 terms are displayed appearing in respective documents

Purpose:

Preprocess text for improving search efficiency with efficient indexing, accuracy, and relevance.

3. Wildcard search and regular search

In [14]:

```
# Search Functions
def wildcard_search(query, inverted_index):
    if not query.endswith('*'):
        return []
    prefix = query[:-1].lower()
    return sorted([term for term in inverted_index.keys()
                  if term.startswith(prefix)])

def regular_search(query, inverted_index, doc_metadata):
    terms = preprocess_text(query)
    if not terms:
        return []

    # Find documents containing ALL terms (AND logic)
    matching_docs = None
    for term in terms:
        if term in inverted_index:
            if matching_docs is None:
                matching_docs = set(inverted_index[term])
            else:
                matching_docs.intersection_update(inverted_index[term])
        else:
            return []  # If any term doesn't exist, return nothing

    return list(matching_docs) if matching_docs else []

def search(query, inverted_index, doc_metadata):
    if query.endswith('*'):
        terms = wildcard_search(query, inverted_index)
        # For wildcard searches, return terms with document counts
        enriched_terms = []
        for term in terms:
            doc_count = len(inverted_index.get(term, []))
            enriched_terms.append({
                'term': term,
                'doc_count': doc_count,
                'example_docs': list(inverted_index.get(term, []))[:3]  # Show
            })
        return {
            'type': 'wildcard',
            'query': query,
            'count': len(terms),
            'results': enriched_terms
        }
    else:
        doc_ids = regular_search(query, inverted_index, doc_metadata)
        results = []
        for doc_id in doc_ids:
            doc = doc_metadata[doc_id]
            results.append({
                'doc_id': doc_id,
                'filename': doc['filename']
            })
        return {
            'type': 'regular',
            'query': query,
            'count': len(results),
            'results': results
        }
```

In [35]:

```
print("\n**************TESTING SEARCHES*********************")
print("\nSearch options:")
print("- Regular search: 'eg: diabetes'")
print("- Wildcard search: 'eg: cardio*'")
print("Type 'exit' to quit\n")

while True:

    query = input("\nEnter Search term: ").strip()

    if query.lower() == 'exit':
        break

    results = search(query, inverted_index, document_metadata)

    if results['type'] == 'wildcard':
        # Wildcard search results
            if results['count'] > 0:
                for term_info in results['results'][:10]:
                    print(f"- {term_info['term']} (in {term_info['doc_count']}
            else:
                print("\nNo matching terms found")  # Wildcard-specific no-res

    else:
            # Regular search results
            if results['count'] > 0:
                for doc in results['results'][:10]:
                    print(f"- Document: {doc['filename']}")
            else:
                print("\nNo direct matches found")  # � Only shows when count
```

```
***************TESTING SEARCHES**********************

Search options:
- Regular search: 'eg: diabetes'
- Wildcard search: 'eg: cardio*'
Type 'exit' to quit

- cardio (in 2 documents)
- cardiogenic (in 1 documents)
- cardiographic (in 1 documents)
- cardiol (in 1 documents)
- cardiolo (in 1 documents)
- cardiologist (in 2 documents)
- cardiologistlevel (in 1 documents)
- cardiology (in 4 documents)
- cardiopulmonary (in 4 documents)
- cardiovascular (in 9 documents)
=== ORIGINAL TEXT (SAMPLE) ===
patient
=== AFTER CLEANING ===
patient
TOKENS (1): ['patient'] ...

AFTER STOPWORD REMOVAL (1): ['patient'] ...

FINAL PROCESSED TERMS (1): ['patient'] ...
- Document: Cardio.pdf
- Document: Cardiovascular  Pulmonary.txt
- Document: DataAnalyticsinhealthcare.pdf
- Document: gender-differences-arteries.pdf
- Document: Medical Specialty.txt
- Document: Medical Specialty_Gastro.pdf
- Document: Medical_history.docx
- Document: mtsamples.csv
- Document: mtsamples.xlsx
- Document: train.txt
```

4. Levenshtein distance logic and suggest terms for misspelled search strings based on distance

In [17]:

```
def levenshtein(s1, s2):
    if len(s1) < len(s2):
        return levenshtein(s2, s1)

    if len(s2) == 0:
        return len(s1)
    prev_row = range(len(s2) + 1)
    for i, c1 in enumerate(s1):
        curr_row = [i + 1]
        for j, c2 in enumerate(s2):
            inserts = prev_row[j + 1] + 1
            deletes = curr_row[j] + 1
            substitute = prev_row[j] + (c1 != c2)
            curr_row.append(min(inserts, deletes, substitute))
        prev_row = curr_row

    return prev_row[-1]

def suggest_terms(misspelled_word, inverted_index, max_suggestions=5):
    # First check for quick matches with common errors
    suggestions = []

    # Calculate distances to all terms in our vocabulary
    distances = []
    for correct_word in inverted_index.keys():
        distance = levenshtein(misspelled_word.lower(), correct_word.lower())
        distances.append((correct_word, distance))

    # Sort by distance (closest first)
    distances.sort(key=lambda x: x[1])

    # Get the top N suggestions with smallest distance
    closest_matches = [word for word, dist in distances[:max_suggestions]]

    return closest_matches
```

Justification:

Above functions are Levenshtein distance calculation. The Levenshtein distance (or edit distance) measures the minimum number of single-character edits (insertions, deletions, or substitutions) required to transform one string into another.

Second function suggest_terms Used to find near matches for misspelled terms using levenshtein function.

In [31]:

```
print("\n **********OUTPUT*********")
print("\nType Mispelled word: 'eg: cardeo'")
print("Type 'exit' to quit")
while True:
            query = input("\nEnter Search term: ").strip()

            if query.lower() == 'exit':
                break  # Exit the loop before processing

            print("\nDid you mean:")
            suggestions = suggest_terms(query, inverted_index)
            print(f"'{query}': {suggestions}")
```

```
**********OUTPUT*********
```

```
Type Mispelled word: 'eg: cardeo'
Type 'exit' to quit
Did you mean:
'dybetes': ['diabetes', 'detec', 'better', 'deep', 'detect']
```

Justification Above functions are created when user wants to search terms.

Purpose of wildcard search: Enables prefix-based searching (e.g., "cardio*" finds "cardiovascular", "cardiology") Supports exploratory searches when users know only the beginning of terms

Purpose of regular search: Performs exact term matching with AND logic Handles preprocessed queries (tokenized, normalized)