**IR ASSIGNMENT – 2**

**Implement Index Term Selection Using TF-IDF and Compare with Simpler Frequency – Based Methods**

**Source Code:**

**Index.html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>TF vs TF-IDF Comparison</title>

    <style>

        body {

            font-family: 'Arial', sans-serif;

            background-color: #f4f4f4;

            margin: 0;

            padding: 0;

            display: flex;

            flex-direction: column;

            align-items: center;

            justify-content: center;

            height: 100vh;

        }

        h2 {

            color: #333;

            margin-bottom: 10px;

        }

        form {

            background: white;

            padding: 20px;

            border-radius: 8px;

            box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);

            display: flex;

            align-items: center;

            gap: 10px;

        }

        input[type="file"] {

                 display: none; /\* Hide the default file input \*/

        }

        .custom-file-upload {

            background: #007bff;

            color: white;

            border: none;

            padding: 10px 15px;

            border-radius: 5px;

            cursor: pointer;

            font-size: 16px;

            transition: 0.3s;

            display: inline-block;

        }

        .custom-file-upload:hover {

            background: #0056b3;

        }

        button {

            background: #28a745;

            color: white;

            border: none;

            padding: 10px 15px;

            border-radius: 5px;

            cursor: pointer;

            font-size: 16px;

            transition: 0.3s;

        }

        button:hover {

            background: #218838;

        }

        #file-name {

            margin-left: 10px;

            font-size: 16px;

            color: #333;

        }

        table {

            width: 80%;

            margin-top: 20px;

            border-collapse: collapse;

            background: white;

            box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);

            border-radius: 10px;

            overflow: hidden;

        }

        th, td {

            padding: 12px;

            border: 1px solid #ddd;

            text-align: center;

        }

        th {

            background-color: #007bff;

            color: white;

        }

        tr:nth-child(even) {

            background-color: #f9f9f9;

        }

        .error {

            color: red;

            margin-top: 10px;

        }

        @media (max-width: 768px) {

            form {

                flex-direction: column;

                gap: 5px;

                width: 90%;

            }

            table {

                width: 100%;

            }

        }

    </style>

</head>

<body>

    <h2>Upload a Text File for Frequency-based Method vs TF-IDF Analysis</h2>

    <form action="/" method="POST" enctype="multipart/form-data">

        <label for="file-upload" class="custom-file-upload">Choose File</label>

        <input id="file-upload" type="file" name="file" required>

        <span id="file-name">No file chosen</span>

        <button type="submit">Upload & Process</button>

    </form>

    <script>

        document.getElementById("file-upload").addEventListener("change", function() {

            const fileNameDisplay = document.getElementById("file-name");

            if (this.files.length > 0) {

                fileNameDisplay.textContent = this.files[0].name;

            } else {

                fileNameDisplay.textContent = "No file chosen";

            }

        })

    </script>

    {% if error %}

        <p class="error">{{ error }}</p>

    {% endif %}

    {% if top\_freq\_terms and top\_tfidf\_terms %}

        <h3>Top Terms Comparison</h3>

        <table>

            <tr>

                <th>Rank</th>

                <th>Top Frequency Terms </th>

                <th>Frequency</th>

                <th>Top TF-IDF Terms</th>

                <th>TF-IDF Score</th>

            </tr>

            {% for i in range(top\_freq\_terms|length) %}

            <tr>

                <td>{{ i+1 }}</td>

                <td>{{ top\_freq\_terms[i][0] }}</td>

                <td>{{ top\_freq\_terms[i][1] }}</td>

                <td>{{ top\_tfidf\_terms[i][0] }}</td>

                <td>{{ "%.4f"|format(top\_tfidf\_terms[i][1]) }}</td>

            </tr>

            {% endfor %}

        </table>

    {% endif %}

</body>

</html>

**Freq\_tf\_idf.py :**

from flask import Flask, render\_template, request

from sklearn.feature\_extraction.text import CountVectorizer, TfidfVectorizer

import numpy as np

import os

app = Flask(\_\_name\_\_)

UPLOAD\_FOLDER = "uploads"

os.makedirs(UPLOAD\_FOLDER, exist\_ok=True)

app.config["UPLOAD\_FOLDER"] = UPLOAD\_FOLDER

def process\_text(file\_path):

    with open(file\_path, "r", encoding="utf-8") as file:

        documents = file.readlines()

    documents = [line.strip() for line in documents if line.strip()]

    if not documents:

        return [], []

    count\_vectorizer = CountVectorizer(stop\_words='english')

    count\_matrix = count\_vectorizer.fit\_transform(documents)

    freq\_scores = np.array(count\_matrix.sum(axis=0)).flatten()

    terms = count\_vectorizer.get\_feature\_names\_out()

    freq\_dict = dict(zip(terms, freq\_scores))

    top\_freq\_terms = sorted(freq\_dict.items(), key=lambda x: x[1], reverse=True)[:10]

    tfidf\_vectorizer = TfidfVectorizer(stop\_words='english')

    tfidf\_matrix = tfidf\_vectorizer.fit\_transform(documents)

    tfidf\_scores = np.array(tfidf\_matrix.sum(axis=0)).flatten()

    tfidf\_dict = dict(zip(tfidf\_vectorizer.get\_feature\_names\_out(), tfidf\_scores))

    top\_tfidf\_terms = sorted(tfidf\_dict.items(), key=lambda x: x[1], reverse=True)[:10]

    return top\_freq\_terms, top\_tfidf\_terms

@app.route("/", methods=["GET", "POST"])

def index():

    if request.method == "POST":

        if "file" not in request.files:

            return render\_template("index.html", error="No file uploaded")

        file = request.files["file"]

        if file.filename == "":

            return render\_template("index.html", error="No file selected")

        file\_path = os.path.join(app.config["UPLOAD\_FOLDER"], file.filename)

        file.save(file\_path)

        top\_freq\_terms, top\_tfidf\_terms = process\_text(file\_path)

        return render\_template("index.html", top\_freq\_terms=top\_freq\_terms, top\_tfidf\_terms=top\_tfidf\_terms)

    return render\_template("index.html")

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(debug=True)

**Screenshots:**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.