Name: Kushal Kishor Shankhapal

Group A, Assignment 1: Conflation Algorithm

Problem Statement:

Implementation of Conflation Algorithm to generate document representative of a text file

conflation_algorithm.cpp

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <unordered_set>
#include <unordered_map>
#include <algorithm>
#include <cctype>
using namespace std;

// Helper to convert word to lowercase
string toLower(string word) {
    transform(word.begin(), word.end(), word.begin(), ::tolower);
    return word;
}

// Basic stemming function (not as advanced as PorterStemmer)
string simpleStem(string word) {
    if (word.length() > 4) {
        if (word.substr(word.length() - 3) == "ing") word = word.substr(0, word.length()
- 3);
        else if (word.substr(word.length() - 2) == "ed") word = word.substr(0,
word.length() - 2);
        else if (word.back() == 's') word = word.substr(0, word.length() - 1);
    }
    return word;
}

// Tokenize and clean text
vector<string> tokenize(string line) {
    vector<string> words;
    string word;
    for (char ch : line) {
        if (isalnum(ch))
            word += tolower(ch);
        else if (!word.empty()) {
            words.push_back(word);
            word.clear();
        }
    }
    if (!word.empty()) words.push_back(word);
    return words;
}

int main() {
    unordered_set<string> stopwords;
    unordered_map<string, int> freq;

    // Load stopwords
    ifstream stopFile("Group_A_1_Conflation_Algorithm/C++/stopwords.txt");
    string sw;
    while (getline(stopFile, sw)) {
        stopwords.insert(toLower(sw));
    }
    stopFile.close();

    // Process document
    ifstream doc("Group_A_1_Conflation_Algorithm/C++/document.txt");
    string line;
    while (getline(doc, line)) {
        vector<string> words = tokenize(line);
        for (auto& word : words) {
            if (stopwords.find(word) == stopwords.end()) {
                string stemmed = simpleStem(word);
                freq[stemmed]++;
            }
        }
    }
    doc.close();

    // Write output
    ofstream out("Group_A_1_Conflation_Algorithm/C++/output.txt");
    for (auto& [word, count] : freq) {
        if (count >= 1) {
            out << word << ": " << count << endl;
        }
    }
    out.close();

    cout << "Output saved to output.txt" << endl;
    return 0;
}
```

stopwords.txt

```
Of, they've, was, weren, after, won, don't, more, again, such, ...
```

document.txt

```
The advancement of technology has significantly transformed the way people live and
communicate.

With the rise of mobile devices and the internet, information is accessible anytime and
anywhere.

People are now able to work remotely, attend virtual meetings, and stay connected with
friends and family across the globe.

However, concerns about data privacy and cybersecurity are also increasing.

To keep up with this rapid evolution, organizations are investing heavily in digital
infrastructure and workforce training.

Technology continues to impact all sectors — from education to healthcare — making
innovation a constant necessity.
```

output.txt

```
innovation: 1
mak: 1
healthcare: 1
sector: 1
train: 1
education: 1
heavily: 1
rapid: 1
necessity: 1
keep: 1
also: 1
privacy: 1
constant: 1
concern: 1
digital: 1
information: 1
device: 1
people: 2
rise: 1
organization: 1
increas: 1
live: 1
friend: 1
mobile: 1
anytime: 1
transform: 1
attend: 1
invest: 1
way: 1
internet: 1
continue: 1
stay: 1
workforce: 1
communicate: 1
family: 1
significantly: 1
anywhere: 1
impact: 1
advancement: 1
accessible: 1
evolution: 1
however: 1
able: 1
cybersecurity: 1
work: 1
remotely: 1
virtual: 1
data: 1
meeting: 1
infrastructure: 1
technology: 2
globe: 1
connect: 1
acros: 1
```