**Assignment 5**

**Code**

#include <iostream>

#include <vector>

#include <set>

#include <string>

using namespace std;

// Function to calculate precision and recall

pair<float, float> calculate\_precision\_recall(const set<string>& answer\_set, const set<string>& relevant\_docs) {

    // Intersection of retrieved documents and relevant documents

    set<string> relevant\_retrieved;

    for (const auto& doc : answer\_set) {

        if (relevant\_docs.find(doc) != relevant\_docs.end()) {

            relevant\_retrieved.insert(doc);

        }

    }

    // Calculate precision and recall

    float precision = answer\_set.size() > 0 ? (float)relevant\_retrieved.size() / answer\_set.size() : 0;

    float recall = relevant\_docs.size() > 0 ? (float)relevant\_retrieved.size() / relevant\_docs.size() : 0;

    return {precision, recall};

}

// Function to calculate harmonic mean (F-measure)

float calculate\_f\_measure(float precision, float recall) {

    return (precision + recall > 0) ? 2 \* (precision \* recall) / (precision + recall) : 0;

}

// Function to calculate E-measure (alpha = 0.5 by default)

float calculate\_e\_measure(float precision, float recall, float alpha = 0.5) {

    return (precision + recall > 0) ? 1.0 - (1.0 / ((alpha / recall) + ((1 - alpha) / precision))) : 0;

}

int main() {

    // Hardcoded input for retrieved documents (answer set)

    set<string> answer\_set\_A = {"d123", "d84", "d56", "d6", "d8", "d9", "d511", "d129", "d187", "d25", "d38", "d48", "d250", "d113", "d3"};

    // Hardcoded input for relevant documents

    set<string> relevant\_docs\_Rq1 = {"d123", "d56", "d9", "d25", "d3"};

    // Calculate precision and recall

    auto [precision, recall] = calculate\_precision\_recall(answer\_set\_A, relevant\_docs\_Rq1);

    // Calculate F-measure

    float f\_measure = calculate\_f\_measure(precision, recall);

    // Calculate E-measure

    float e\_measure = calculate\_e\_measure(precision, recall);

    // Display precision, recall, F-measure, and E-measure

    cout << "\nPrecision: " << precision << endl;

    cout << "Recall: " << recall << endl;

    cout << "F-measure (Harmonic Mean): " << f\_measure << endl;

    cout << "E-measure: " << e\_measure << endl;

    return 0;

}

**OUTPUT**:

Precision: 0.333333

Recall: 1

F-measure (Harmonic Mean): 0.5

E-measure: 0.5