Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа № 1

По дисциплине ЕЯИИС

Тема: «Разработка информационно-поисковой системы и методы оценки качества ее работы»

Выполнил:

Студент 4 курса

Группы ИИ-16

Пилипук З.А.

Проверил:

Слинко Е.В.

Брест, 2021

**Цель работы** освоить на практике основные принципы реализации информационно-поисковых систем и методы оценки качества их работы.

**Код программы:**

#pragma once

#include <string>

#include <map>

#include <vector>

class SearchEngine

{

public:

SearchEngine();

std::map<double, std::string> search(const std::string& request);

private:

void setupSearchConfig(const std::string& request);

double B;

size\_t fileCount;

std::string dataFolder\_;

std::vector<std::pair<std::string, int>> wordCount;

std::map<double, std::string> result\_;

};

#include "SearchEngine.h"

#include <fstream>

#include <filesystem>

#include <iostream>

SearchEngine::SearchEngine() : B(0), fileCount(0)

{

std::ifstream file("config.txt");

std::getline(file, dataFolder\_);

file.close();

}

std::map<double, std::string> SearchEngine::search(const std::string& request)

{

fileCount = 0;

B = 0;

result\_.clear();

wordCount.clear();

setupSearchConfig(request);

for (const auto& x : wordCount)

{

result\_.insert({ B \* x.second, x.first });

}

return result\_;

}

void SearchEngine::setupSearchConfig(const std::string& request)

{

for (const auto& dirPath : std::filesystem::recursive\_directory\_iterator(dataFolder\_))

{

++fileCount;

bool wordExists = false;

size\_t wordCountFile = 0;

std::ifstream file(dirPath);

while (!file.eof())

{

std::string word;

file >> word;

if (word.size() && !isalpha(word[word.size() - 1]))

{

word.pop\_back();

}

if (word == request)

{

wordExists = true;

++wordCountFile;

}

}

file.close();

if (wordExists)

{

wordCount.push\_back({ dirPath.path().filename().string(), wordCountFile });

}

}

B = log10((double)wordCount.size() / (double)fileCount);

}

#include <iostream>

#include <string>

#include <Windows.h>

#include "SearchEngine.h"

int main()

{

std::cout << "Search for: ";

std::string request;

std::getline(std::cin, request);

SearchEngine engine;

auto res = engine.search(request);

std::cout << std::endl;

int id = 1;

for (const auto& x : res)

{

std::cout << id++ << ". " << x.second << std::endl;

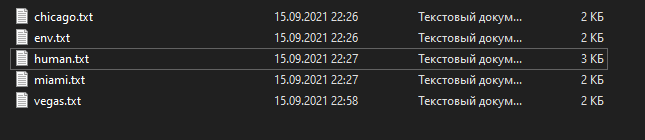
}

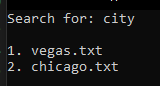
return 0;

}

**Тестирование:**

Файловая система содержит файлы:

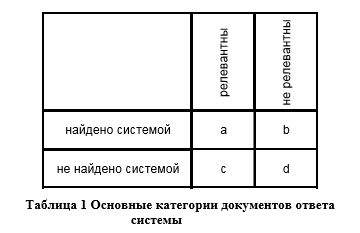


При запросе “city”  


Система отдает найденные файлы в порядке важности:

1. vegas.txt
2. chicago.txt

Во всех файлах так или иначе упоминается “city”. Далее результат vegas.txt и Chicago.txt – считаем за правильную работу системы. Так же “miami.txt” так же связан с темой поиска, и это файл можно считать за тот, который мы хотели бы получить.



Исходя из данных выше:

a = 2, b = 2, c = 1, d = 2;

Метрики:

* 1. Полнота: r = a/(a+c);  
     r = 2/3 = 0.66
  2. Точность: p = a/(a+b)

P = 2/4 = 0.5

* 1. Аккуратность: accuracy = (a+d) / (a+b+c+d)

Accuracy = 4/7 = 0.57

* 1. Ошибка: error = (b+c)/(a+b+c+d)

Error = 3/7 = 0.42

* 1. F-measure: F = 2/((1/p) + (1/r))

F = 2/(1/0.5 + 1/0.66) = 0.57

На небольшом количестве файлов и простых запросах система показывает себя хорошо, нужно расширять базу данных файловой системы.

**Вывод:** освоил на практике основные принципы реализации информационно-поисковых систем и методы оценки качества их работы.