Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Институт информационных технологий

Кафедра ИС

ОТЧЁТ

по лабораторной работе №8

АЛГОРИТМЫ ОБРАБОТКИ ТЕКСТОВЫХ ДАННЫХ

Выполнил:

ст. гр. ИС/б-21-2-о

Шевелёв К. С.

Проверил:

Минкин С. И.

Севастополь

2024

1. ПОСТАНОВКА ЗАДАЧИ
   1. Построить облако тегов по тексту.
   2. Реализовать алгоритм вычисления разницы между двумя текстами (антиплагиат)
2. ХОД РАБОТЫ
   1. Построим облако тегов по тексту.

Листинг 1 – Код программы

using System;

using System.Collections.Generic;

using System.Linq;

public class TagCloudBuilder

{

public Dictionary<string, int> BuildTagCloud(string text)

{

Dictionary<string, int> tagCloud = new Dictionary<string, int>();

string[] words = text.Split(new[] { ' ', ',', '.', ';', ':', '!', '?' }, StringSplitOptions.RemoveEmptyEntries);

foreach (string word in words)

{

if (IsNounInNominativeCase(word) || word.Length >= 3)

{

string normalizedWord = word.ToLower();

if (tagCloud.ContainsKey(normalizedWord))

{

tagCloud[normalizedWord]++;

}

else

{

tagCloud.Add(normalizedWord, 1);

}

}

}

return tagCloud;

}

private bool IsNounInNominativeCase(string word)

{

return false;

}

}

public class Program

{

public static void Main(string[] args)

{

string text = "Солнце светит, солнце греет. Лето пришло, лето встречает.";

TagCloudBuilder tagCloudBuilder = new TagCloudBuilder();

Dictionary<string, int> tagCloud = tagCloudBuilder.BuildTagCloud(text);

foreach (KeyValuePair<string, int> tag in tagCloud)

{

Console.WriteLine($"Слово: {tag.Key}, Количество: {tag.Value}");

}

}

}

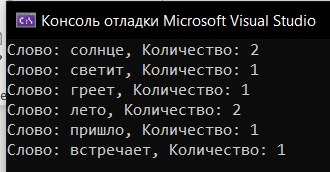


Рисунок 1 – Результат работы программы

* 1. Реализован алгоритм вычисления разницы между двумя текстами (антиплагиат).

Листинг 2 – Код программы

using System;

public class PlagiarismChecker

{

public void CompareTexts(string text1, string text2)

{

int[,] matrix = new int[text1.Length + 1, text2.Length + 1];

for (int i = 0; i <= text1.Length; i++)

{

matrix[i, 0] = i;

}

for (int j = 0; j <= text2.Length; j++)

{

matrix[0, j] = j;

}

for (int i = 1; i <= text1.Length; i++)

{

for (int j = 1; j <= text2.Length; j++)

{

if (text1[i - 1] == text2[j - 1])

{

matrix[i, j] = matrix[i - 1, j - 1];

}

else

{

matrix[i, j] = Math.Min(Math.Min(matrix[i - 1, j] + 1, matrix[i, j - 1] + 1), matrix[i - 1, j - 1] + 1);

}

}

}

int matchesCount = text1.Length + text2.Length - matrix[text1.Length, text2.Length];

double percentage = (double)matchesCount / (double)(text1.Length + text2.Length) \* 100;

Console.WriteLine("Результат сравнения текстов:");

Console.WriteLine($"Общее число совпадений: {matchesCount}");

Console.WriteLine($"Процентное соотношение совпадений: {percentage}%");

}

}

public class Program

{

public static void Main(string[] args)

{

string t1 = "Привет, как твои дела?";

string t2 = "Привет, дела в порядке";

PlagiarismChecker checker = new PlagiarismChecker();

checker.CompareTexts(t1, t2);

}

}

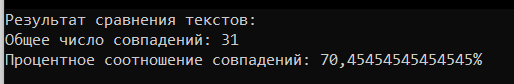


Рисунок 2 – Результат работы программы

ВЫВОД

В ходе выполнения лабораторной работы были изучены подходы к строительству облаков тегов по тексту, а также к реализации проверки на антиплагиат.