**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №5

«Вычисление расстояния Ливенштейна»

|  |  |  |
| --- | --- | --- |
| Выполнила: |  | Проверил: |
| студентка группы ИУ5-33 |  | преподаватель каф. ИУ5 |
| Беспалова У.А. |  | Гапанюк Ю.Е. |
| Подпись и дата:  08.11.2018 |  | Подпись и дата: |

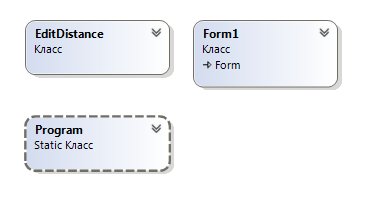
Москва, 2018 г.

1. Содержания задания

Разработать программу, реализующую вычисление расстояния Левенштейна с использованием алгоритма Вагнера-Фишера.

1. Программа должна быть разработана в виде библиотеки классов на языке C#.
2. Использовать самый простой вариант алгоритма без оптимизации.
3. Дополнительно возможно реализовать вычисление расстояния Дамерау-Левенштейна (с учетом перестановок соседних символов).
4. Модифицировать предыдущую лабораторную работу, вместо поиска подстроки используется вычисление расстояния Левенштейна.
5. Предусмотреть отдельное поле ввода для максимального расстояния. Если расстояние Левенштейна между двумя строками больше максимального, то строки считаются несовпадающими и не выводятся в список результатов.

Диаграмма классов



Текст программы

using System;

using System.IO;

using System.Diagnostics;

using System.Collections.Generic;

//using EditDistanceProject;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Liven;

namespace лаба4

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent(); //просто не будем это трогать.

}

/// <summary>

/// Список слов

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

List<string> list = new List<string> ();//создали пустой лист

private void Form1\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

OpenFileDialog fileD = new OpenFileDialog();

fileD.Filter = "текстовые файлы|\*.txt"; //ограничиваемся текстовыми файлами

if (fileD.ShowDialog() == DialogResult.OK)

{

Stopwatch t = new Stopwatch(); //штучка для измерения времени

t.Start(); //начали отсчет

string text = File.ReadAllText(fileD.FileName); //чтение файла в виде строки

string[] textArray = text.Split(new char[] { ' ', '.', ',', '!', '?', '-', ';', ':', '/', '\t', '\n' }, StringSplitOptions.RemoveEmptyEntries); //разделяем строку на слова

foreach (string i in textArray)

{

string str = i.Trim(); //поудаляли пробелы перед и после слова

if (!list.Contains(str)) list.Add(str); //если в списке такого слова нет, добавляем его

}

t.Stop(); //остановили таймер

this.label4.Text = (t.ElapsedTicks\*0.0001).ToString(); //посчитали время

this.label1.Text = list.Count.ToString(); //посчитали количество записанных слов

}

else

{

MessageBox.Show("Нужно выбрать файл");

}

}

private void buttonSeach\_Click(object sender, EventArgs e)

{

string word = this.textBoxFind.Text.Trim(); //введенное слово для поиска

if (!string.IsNullOrWhiteSpace(word) && list.Count > 0) //если слово не пустое

{

int kol;

if (!int.TryParse(this.TextBoxKol.Text.Trim(), out kol))

{

MessageBox.Show("Необходимо указать максимальное расстояние");

return;

}

if (kol<1||kol>5)

{

MessageBox.Show("Масксимальное расстояние должно быть больше 1 и меньше 5");

return;

}

string wordUp = word.ToUpper();//перевели в верхний регистр

List<string> tempList = new List<string>();//пустой массив для найденных слов

Stopwatch t = new Stopwatch();

t.Start(); //начали отсчет

foreach (string i in list)

{

if (EditDistance.Distance(i.ToUpper(), wordUp) <=kol) // если растояние меньше указанного, то

{

tempList.Add(i); //записываем во временный список

}

}

t.Stop();

this.labelTime2.Text = (t.ElapsedTicks \* 0.0001).ToString();

this.listBoxResults.Items.Clear(); //очистили список (чтоб не показывал предыдущие результаты)

this.listBoxResults.BeginUpdate();

foreach (string i in tempList)

{

this.listBoxResults.Items.Add(i);

}

this.listBoxResults.EndUpdate();

}

else

{

MessageBox.Show("Нужно ввести слово для поиска");

}

}

private void label5\_Click(object sender, EventArgs e)

{

}

}

}

Сlass 1

public class EditDistance //посчитать расстояние

{

public static int Distance(string str1, string str2)

{

if ((str1==null)||(str2==null)) return -1; //если обе строки пустые, конец

int len1 = str1.Length;

int len2 = str2.Length;

if ((len1 == 0) && (len2 == 0)) return 0; //если есть нулевые строки, все просто

if (len1 == 0) return len2;

if (len2 == 0) return len1;

string strUp1 = str1.ToUpper();

string strUp2 = str2.ToUpper(); //приводим к верхнему регистру

int[,] matrix = new int[len1 + 1, len2 + 1];//создали матрицу

for (int i = 0; i < len1; i++) matrix[i, 0] = i;

for (int j = 0; j < len2; j++) matrix[0, j] = j; //пронумеровали столбики и строчки

for (int i=1; i<len1+1; i++)

{

for (int j=1; j<len2+1; j++)

{

int sEqual = ((strUp1.Substring(i - 1, 1) == strUp2.Substring(j - 1, 1)) ? 0 : 1);

//если соответствующие символы равны, присваиваем 0, иначе 1

int oI = matrix[i, j - 1] + 1; //insert

int oD = matrix[i - 1, j] + 1; //delete

int oR = matrix[i - 1, j - 1] + sEqual; //replace

matrix[i, j] = Math.Min(Math.Min(oI, oD), oR); //минимум

if ((i>1)&&(j>1)&& //если элемент не первый (не номер столбика)

(strUp1.Substring(i-1,1) == strUp2.Substring(j-2, 1))&& //если при перестановке символы равны

(strUp1.Substring(i - 2, 1) == strUp2.Substring(j - 1, 1)))

{

matrix[i, j] = Math.Min(matrix[i, j], matrix[i - 2, j - 2] + sEqual);

}

}

}

return matrix[len1, len2]; //вернули нижний правый

}

}

Экранные формы с примерами выполнения программы

