МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ «МИСиС»**

*ИНСТИТУТ* ИТАСУ

*КАФЕДРА* ИНЖЕНЕРНОЙ КИБЕРНЕТИКИ

*НАПРАВЛЕНИЕ* 09.04.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

**КУРСОВАЯ РАБОТА**

**на тему:** РАЗРАБОТКА СИСТЕМЫ ДЛЯ ОПРЕДЕЛЕНИЯ ЕСТЕСТВЕННОГО ЯЗЫКА ЭЛЕКТРОННОГО ТЕКСТА

**Студенты: Гайер А.В. – разработчик**

**Сериков А. Ю. – разработчик**

**Тихонова К. С. – аналитик**

**Хабибулин Р. И. – аналитик, тестировщик**

**Группа: МПИ-17-2-2**

**Руководитель: Тарханов И. А.**

**Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Дата защиты: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Москва, 2018

**СОДЕРЖАНИЕ**

1 Цель и задачи системы TextLang 6

2 Прототип системы TextLang 7

3 Требования к системе TextLang 8

3.1 Функциональные требования к системе TextLang 8

3.2 Нефункциональные требования к системе TextLang 11

4 Use Cases 16

5.1 Use Case 1 – Определить естественный язык текста из текстового файла 16

5.2 Use Case 2 – Определить естественный язык текста из текстового поля 17

5.3 Use Case 3 – Сохранить результат работы системы TextLang 17

5.4 Use Case 4 – Восстановить результат работы системы TextLang из текстового файла формата .txt 18

5 Test Cases 19

5.1 Test Case 1 – 19

1. **Цель и задачи системы TextLang**

**Цель системы TextLang:** позволить пользователю быстро определять естественный язык электронного текста.

**Задачи системы TextLang:**

1. необходимо предоставить пользователям возможность быстро определять естественный язык электронного текста;

2. необходимо предоставить пользователям возможность сохранять в файле результат работы приложения;

3. необходимо предоставить пользователям возможность отображать в приложении сохраненный в файле результат.

1. **Прототип системы TextLang**

Тут будет прототип

1. **Требования к системе TextLang**
   1. **Функциональные требования к системе TextLang**
2. **Выбор текстового файла для определения языка:**
3. Система TextLang должна предоставлять возможность выбора текстового файла (*fileDetect*), необходимого для определения языка;
4. Выбор текстового файла (*fileDetect)* производится пользователем системы TextLang путем нажатия кнопки *Select File;*
5. Единовременно пользователь может выбрать только один текстовый файл (*fileDetect)* в открывшемся диалоговом окне;
6. Название выбранного файла (*fileDetect)* должно отображаться рядом с кнопкой *Select File;*
7. Выбирать текстовый файл (*fileDetect)* заново пользователь может неограниченное число раз.
8. **Считывание текста из файла *fileDetect:***
9. Система TextLang должна уметь считывать текст из файла *fileDetect.*
10. **Добавление текста в текстовое поле:**
11. Система TextLang должна предоставлять возможность добавления текста в текстовое поле *textField*;
12. Добавление текста в текстовое поле *textField* осуществляется пользователем системы TextLang;
13. Добавляемый текст может быть написан на любом естественном языке с использованием любых символов;
14. В дальнейшем пользователь может неоднократно изменять текст в текстовом поле *textField*.
15. **Выбор опции (*TextBox* или *File),* показывающей, какой текст будет использован для определения естественного языка:**
16. Система TextLang должна предоставлять возможность выбора только одной из двух опций (*TextBox* или *File);*
17. Выбор опции производится пользователем системы TextLang путем нажатия радиокнопки, которая соответствует опции;
18. В любой момент времени работы системы одна из опций должна быть активна (т.е. радиокнопка, соответствующая опции, должна быть включена).
19. **Определение естественного языка текста:**
20. При нажатии кнопки *Detect Language* система TextLang должна удалять во входном тексте все символы, которые не являются буквами, иероглифами или пробелами (далее – «оставшийся текст»);
21. Система TextLang должна проверять, является ли оставшийся текст пустым;
22. Если оставшийся текст непустой, то после нажатия пользователем кнопки *Detect Language*:
23. Система TextLang должна определять естественный язык текста;
24. Для каждого языка из таблицы 1 (см. нефункциональное требование 8.1) система должна считать степень уверенности в том, что текст написан на данном языке (набор соответствий «*язык-степень уверенности»* будем называть *ScoreValues*);
25. В поле *Prediction* система должна выводить один естественный язык, имеющий максимальное значение из всех степеней уверенности (см. пункт 5.3.2);
26. В поле *Score* система должна выводить значение степени уверенности системы в том, что текст написан на языке из поля *Prediction*;
27. Справа от поля *textField* должна отображаться столбчатая диаграмма (*TOP10)*, в которой:

* на оси *x* отмечены названия 10 естественных языков, имеющих наибольшие значения степеней уверенности среди *ScoreValues* (см. пункт 5.3.2)*;*
* на оси *y* отмечены значения степеней уверенности для языков, представленных на оси *x*.

1. Если оставшийся текст пустой, то после нажатия пользователем кнопки *Detect Language*:
2. Система TextLang должна выводить пользователю сообщение: *"String for detection is empty!"*;
3. Поля *Prediction, Score, textField* не должны измениться;
4. Столбчатая диаграмма (*TOP10*) не должна измениться.
5. Определять язык заново система может неограниченное количество раз.
6. **Сохранение результатов работы системы TextLang:**
7. При нажатии кнопки *Save Result* система TextLang должна сохранять результат ее работы на компьютер пользователя;
8. Результат работы системы должен сохраняться в файл, имеющий следующий формат: dd.*MM.yyyy*\_HH.mm.ss*\_Prediction.txt*, где *Prediction* – это значение поля *Prediction*;
9. В каждом файле с результатом должны содержаться:

* время нажатия кнопки *Detect Language* в формате dd*.MM.yyyy*\_HH.mm.ss (в первой строке файла);
* значение поля *Prediction* (во второй строке файла);
* значение поля *Score* (в третьей строке);
* текст, который пользователь предоставил системе (в четвертой и последующих строках файла).

1. **Выбор текстового файла для восстановления результатов работы системы:**
2. Система TextLang должна предоставлять возможность выбора текстового файла (*fileRecover*), необходимого для восстановления результатов работы системы;
3. Выбор текстового файла (*fileRecover)* производится пользователем системы TextLang путем нажатия кнопки *Display Result From File;*
4. Единовременно пользователь может выбрать только один текстовый файл (*fileRecover)* в открывшемся диалоговом окне;
5. Выбирать текстовый файл (*fileRecover*) заново пользователь может неограниченное число раз.
6. **Считывание текста из файла *fileRecover:***
7. Система TextLang должна уметь считывать текст из файла *fileRecover*.
8. **Восстановление результатов работы системы TextLang из файла:**
9. Система TextLang должна уметь восстанавливать результаты ее работы из файла (*fileRecover*), имеющего следующий вид:

* первая строка файла: строковое значение;
* вторая строка файла: строковое значение;
* третья строка файла: число типа *float*;
* четвертая и последующие строки файла: текст, который пользователь предоставлял системе.

1. Если формат файла (*fileRecover*) соответствует формату из пункта 9.1:
2. Система TextLang должна отображать вторую строку из файла *(fileRecover)* в поле *Prediction*;
3. Система TextLang должна отображать третью строку из файла *(fileRecover)* в поле *Score*;
4. Система TextLang должна отображать четвертую и последующие строки из файла (*fileRecover)* в текстовом поле *textField*;
5. Столбчатая диаграмма (*TOP10*) не должна отображаться.
6. Если формат файла (*fileRecover)* не соответствует формату из пункта 9.1:
7. Система TextLang должна выдавать пользователю сообщение: *"Checkpoint is corrupted! Unable to load"*;
8. Поля *Prediction, Score, textField* не изменяются;
9. Столбчатая диаграмма (*TOP10*) не изменяется.
   1. **Нефункциональные требования к системе TextLang**
10. **Требования к языку программирования:**
11. Приложение TextLang должно быть написано на языке программирования Python.
12. **Требования к используемым библиотекам:**
13. Приложение TextLang должно быть разработано с использованием библиотеки TensorFlow.
14. **Требования к модели, используемой для определения естественного языка текста:**
15. Для определения естественного языка текста должна быть использована сверточная нейронная сеть (*Сonvolutional Neural Network*);
16. Данные о структуре нейронной сети должны храниться в файле *cnn.py*.
17. **Требования к приложению:**
18. Приложение должно работать на компьютерах, на которых установлен Python 3.5;
19. Приложение должно работать через браузеры Google Chrome и Mozilla Firefox;
20. Приложение должно выдавать результат (естественный язык) не более чем за 2 секунды;
21. Точность системы (*Accuracy*) должна быть не меньше 80%;
22. Точность (*Accuracy*) должна считаться для тестовой сбалансированной выборки, состоящей как минимум из 3000 примеров, по следующей формуле:
23. Приложение должно поддерживать все символы UNICODE стандарта ISO/IEC 10646.
24. **Требования к размеру текстового поля *textField*:**
25. В приложении TextLang текстовое поле *textField* должно содержать не более 1 млн символов.
26. **Требования к формату загружаемых файлов *fileDetect* и *fileRecover*:**
27. Система TextLang должна уметь загружать только файлы формата *.txt*.
28. **Требования к размеру загружаемых файлов *fileDetect* и *fileRecover*:**
29. Размер файлов, загружаемых в систему TextLang, не должен превышать 2MB.
30. **Требования к поддержке естественных языков:**
31. Система TextLang должна поддерживать определение 65 естественных языков, перечисленных в таблице 1.

Таблица 1 – Языки, поддерживаемые системой TextLang

|  |  |  |
| --- | --- | --- |
| Номер языка | Код языка | Естественный язык |
| 1 | ar | Arabic |
| 2 | az | Azerbaijani |
| 3 | bg | Bulgarian |
| 4 | bn | Bengali |
| 5 | bo | Tibetan |
| 6 | cs | Czech |
| 7 | da | Danish |
| 8 | de | German |
| 9 | el | Greek |
| 10 | en | English |
| 11 | es | Spanish |
| 12 | fa | Persian |
| 13 | fi | Finnish |
| 14 | fil | Filipino |
| 15 | fr | French |
| 16 | gu | Gujarati |
| 17 | he | Hebrew |
| 18 | hi | Hindi |
| 19 | ht | Haitian |
| 20 | hu | Hungarian |
| 21 | hy | Armenian |
| 22 | id | Indonesian |
| 23 | is | Icelandic |
| 24 | it | Italian |
| 25 | ja | Japanese |
| 26 | ka | Georgian |
| 27 | km | Khmer |
| 28 | kn | Kannada |
| 29 | ko | Korean |
| 30 | ku | Kurdish |
| 31 | lt | Lithuanian |
| 32 | mg | Malagasy |
| 33 | ml | Malayalam |
| 34 | mn | Mongolian |
| 35 | ms | Malay |
| 36 | my | Burmese |
| 37 | nb | Norwegian Bokmal |
| 38 | ne | Nepali |
| 39 | nl | Dutch |
| 40 | nn | Norwegian Nynorsk |
| 41 | pl | Polish |
| 42 | ps | Pashto |
| 43 | pt | Portuguese |
| 44 | ro | Romanian |
| 45 | ru | Russian |
| 46 | si | Sinhalese |
| 47 | sk | Slovak |
| 48 | sl | Slovenian |
| 49 | so | Somali |
| 50 | sq | Albanian |
| 51 | sv | Swedish |
| 52 | sw | Swahili |
| 53 | ta | Tamil |
| 54 | te | Telugu |
| 55 | tg | Tajik |
| 56 | th | Thai |
| 57 | tl | Tagalog |
| 58 | tr | Turkish |
| 59 | ug | Uyghur |
| 60 | uk | Ukrainian |
| 61 | ur | Urdu |
| 62 | uz | Uzbek |
| 63 | vi | Vietnamese |
| 64 | zh-cn | Simplified Chinese |
| 65 | zh-tw | Traditional Chinese |

1. **Требования к отображению файла с сохраненными результатами работы системы:**
2. Файл с сохраненными результатами работы системы должен корректно отображаться через текстовый редактор Notepad++ (т.е. в соответствии с форматом, описанным в пункте 6.3 функциональных требований).
3. **Use Cases**
4. **Use Case 1 – Определить естественный язык текста из текстового файла**

**Цель:** определить естественный язык текста из текстового файла

**Действующее лицо:** пользователь системы

**Активатор:** нажатие кнопки *Detect Language*

**Основной сценарий:**

1. Пользователь нажимает на кнопку *Select File*;
2. Система открывает диалоговое окно, с помощью которого можно выбрать только файл формата *.txt*;
3. Пользователь выбирает файл и диалоговое окно закрывается;
4. Текст из файла считывается системой;
5. Система выводит название выбранного файла рядом с кнопкой *Select File*.
6. Пользователь выбирает опцию выбора текста: *File*;
7. Пользователь нажимает на кнопку *Detect Language*;
8. Система удаляет в тексте символы, которые не являются буквами, иероглифами и пробелами;
9. Система определила, что оставшийся текст не является пустым;
10. Система определяет естественный язык текста и выводит его название в поле *Prediction*;
11. Система выводит в поле *Score* значение степени уверенности системы в том, что текст написан на языке из поля *Prediction*;
12. Справа от поля *textField* отображается столбчатая диаграмма *TOP10* (см. пункт 5.3.5 функциональных требований).

**Альтернативный сценарий:**

1. Система определила, что оставшийся текст является пустым;
2. Система выводит сообщение: "*String for detection is empty!*".
3. **Use Case 2 – Определить естественный язык текста из текстового поля**

**Цель:** определить естественный язык текста из текстового поля

**Действующее лицо:** пользователь системы

**Активатор:** нажатие кнопки *Detect Language*

**Основной сценарий:**

1. Пользователь набирает текст в текстовом поле *textField*;
2. Текст из текстового поля считывается системой;
3. Пользователь выбирает опцию выбора текста: *TextBox*;
4. Пользователь нажимает на кнопку *Detect Language*;
5. Система удаляет в тексте символы, которые не являются буквами, иероглифами и пробелами;
6. Система определила, что оставшийся текст не является пустым;
7. Система определяет естественный язык текста и выводит его название в поле *Prediction*;
8. Система выводит в поле *Score* значение степени уверенности системы в том, что текст написан на языке из поля *Prediction*;
9. Справа от поля *textField* отображается столбчатая диаграмма *TOP10* (см. пункт 5.3.5 функциональных требований).

**Альтернативный сценарий:**

1. Система определила, что оставшийся текст является пустым;
2. Система выводит сообщение: "*String for detection is empty!*".
3. **Use Case 3 – Сохранить результат работы системы TextLang**

**Цель:** сохранить результат определения естественного языка текста

**Действующее лицо:** пользователь системы

**Заинтересованные лица:** разработчики, тестировщики

**Активатор:** нажатие кнопки *Save Result*

**Сценарий:**

1. Система создает выходной текст, первая строка которого представляет собой дату и время нажатия кнопки *Detect Language*, во второй строке записывается значение поля *Prediction*, в третьей строке – значение поля *Score*, а в последующих строках записывается входной текст;
2. Система создает файл, который имеет название следующего формата:   
   dd.*MM.yyyy*\_HH.mm.ss*\_Prediction.txt*, где *Prediction* – это значение поля *Prediction;*
3. Система записывает выходной текст в файл из пункта 2;
4. Система сохраняет файл на компьютер пользователя.
5. **Use Case 4 – Восстановить результат работы системы TextLang из текстового файла формата .txt**

**Цель:** отобразить результат работы системы TextLang, который был сохранен в файл специального вида

**Действующее лицо:** пользователь системы

**Активатор:** нажатие кнопки *Display Result From File*

**Основной сценарий:**

1. Пользователь нажимает на кнопку *Display Result From File*;
2. Система открывает диалоговое окно, с помощью которого можно выбрать только файл формата *.txt*;
3. Пользователь выбирает файл и диалоговое окно закрывается;
4. Текст из файла считывается системой;
5. Система определила, что вид текста корректен;
6. Текст из второй строки файла отображается в поле *Prediction*;
7. Текст из третьей строки файла отображается в поле *Score*;
8. Текст, начинающийся с четвертой строки в выбранном файле, отображается в текстовом поле *textField*.

**Альтернативный сценарий:**

1. Система определила, что вид текста некорректен;
2. Система выводит сообщение: "*Checkpoint is corrupted! Unable to load*".
3. **Test Cases**
4. **Test Case 1 –**

Тут будут тест кейсы