Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №5

По дисциплине: «Естественно-языковой интерфейс ИС»

# Тема: «Разработка автоматизированной системы семантико-синтаксического анализа текста естественного языка»

Выполнил:

Студент 3 курса

Группы ИИ-21

Романко Н. А.

Проверила:

Якимук А. В.

Брест 2024

**Цель:** освоить принципы разработки прикладных сервисных программ для решения задачи автоматического семантико-синтаксического анализа текста естественного языка.

**Ход работы:**

|  |  |  |
| --- | --- | --- |
| **Номер**  **варианта** | **Язык текста** | **Формат входного документа** |
| 2 | Русский | RTF |

**Методические указания:**

Требуется спроектировать и программно реализовать структуры хранения данных, алгоритмы их обработки, необходимые в рамках следующих базовых требований к разрабатываемому приложению:

* входные данные – текст заданного естественного языка;
* выходные данные – структуры, полученные при проведении автоматического семантико-синтаксического анализа предложений входного текста согласно варианту задания;
* взаимодействие с пользователем посредством графического интерфейса (интерфейс должен быть интуитивно понятным и дружественным пользователю);
* наличие системы средств помощи пользователю;
* обеспечение возможности построения, сохранения, просмотра, редактирования, документирования автоматически получаемого результата либо заданной его части;
* поддержка различных форматов представления входных данных (TXT, RTF, PDF, HTML, DOC, DOCX).

**Структурно-функциональная схема:**



**Листинг:**

import tkinter as tk

from tkinter import filedialog, messagebox

import spacy

from striprtf.striprtf import rtf\_to\_text

nlp = spacy.load("ru\_core\_news\_sm")

# Словарь для перевода меток частей речи

POS\_TRANSLATIONS = {

…

}

# Словарь для перевода меток зависимостей

DEP\_TRANSLATIONS = {

…

}

def analyze\_text(text):

doc = nlp(text)

structures = []

for sent in doc.sents:

structure = {

'text': sent.text,

'tokens': [{'text': token.text, 'lemma': token.lemma\_, 'pos': translate\_pos(token.pos\_), 'dep': translate\_dep(token.dep\_)} for token in sent]

}

structures.append(structure)

return structures

def extract\_text\_from\_rtf(filename):

try:

with open(filename, 'rb') as file:

rtf\_content = file.read()

return rtf\_to\_text(rtf\_content.decode('utf-8'))

except Exception as e:

messagebox.showerror("Ошибка", f"Не удалось извлечь текст из файла: {e}")

return None

def translate\_pos(pos):

return POS\_TRANSLATIONS.get(pos, pos)

def translate\_dep(dep):

return DEP\_TRANSLATIONS.get(dep, dep)

def open\_file():

filename = filedialog.askopenfilename(filetypes=[("RTF Files", "\*.rtf")])

if filename:

text = extract\_text\_from\_rtf(filename)

if text:

structures = analyze\_text(text)

if structures:

result\_text.delete(1.0, tk.END)

for structure in structures:

result\_text.insert(tk.END, f"Текст: {structure['text']}\n")

result\_text.insert(tk.END, "Семантико-синтаксический анализ:\n")

for token in structure['tokens']:

result\_text.insert(tk.END, f"Токен: {token['text']}, Лемма: {token['lemma']}, Часть речи: {token['pos']}, Зависимость: {token['dep']}\n")

result\_text.insert(tk.END, "\n" + "="\*50 + "\n")

update\_result\_text\_size()

def update\_result\_text\_size():

width = int(root.winfo\_width() \* 0.6)

height = root.winfo\_height()

result\_text.config(width=width, height=height)

root = tk.Tk()

root.title("Тоже что и 4, но с учетом семантики")

frame = tk.Frame(root)

frame.pack(padx=10, pady=10)

open\_button = tk.Button(frame, text="Открыть файл", command=open\_file)

open\_button.pack(side=tk.LEFT)

result\_text = tk.Text(frame, width=50, height=20)

result\_text.pack(side=tk.RIGHT)

root.bind("<Configure>", lambda event: update\_result\_text\_size())

root.mainloop()

# Результат:

Изображение выглядит как текст, снимок экрана, документ, число

Автоматически созданное описание

**Вывод:** в ходе выполнения лабораторной работы освоил принципы разработки прикладных сервисных программ для решения задачи автоматического семантико-синтаксического анализа текста естественного языка.