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

Карагодин Д. Л.

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** |
| 11 | Английский | DOCX |

**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

import spacy

from spacy.matcher import DependencyMatcher

import docx2txt

# Загрузить модель spaCy для английского языка

nlp = spacy.load("en\_core\_web\_sm")

# Создать сопоставитель для поиска отношений между сущностями

matcher = DependencyMatcher(nlp.vocab)

pattern = [{"RIGHT\_ID": "subj", "RIGHT\_ATTRS": {"POS": "NOUN"}},

           {"LEFT\_ID": "subj", "REL\_OP": ">", "RIGHT\_ID": "obj", "RIGHT\_ATTRS": {"POS": "NOUN"}}]

matcher.add("RELATION", [pattern])

def analyze\_semantics(text):

    # Проанализировать текст

    doc = nlp(text)

    # Найти отношения между сущностями

    matches = matcher(doc)

    # Добавить отношения в объект документа

    for match in matches:

        span = doc[match[1][0]:match[1][1]]

        span.set\_extension("RELATION",default=True,force=True)

        doc.ents += (span,)

            return doc

def display\_semantics(doc):

    # Очистить поле вывода

    text\_box.delete("1.0", tk.END)

    # Добавить отношения в поле вывода

    for ent in doc.ents:

        print(type(ent))

        if ent.\_.get("RELATION"):

            text\_box.insert("end", f"{ent.text} - {ent.root.head.text}\n")

    #text\_box.insert("end", "\n")

def open\_file():

    # Открыть диалоговое окно "Открыть файл"

    file\_path = filedialog.askopenfilename(filetypes=[("DOCX files", "\*.docx")])

    if file\_path:

        # Прочитать текст из файла

        # with open(file\_path, "r") as f:

        #     text = f.read()

        text = docx2txt.process(file\_path)

        # Проанализировать семантику текста

        doc = analyze\_semantics(text)

        # Отобразить семантический анализ

        display\_semantics(doc)

def show\_help():

    # Создать новое окно для руководства

    help\_window = tk.Toplevel()

    help\_window.title("Руководство")

    # Добавить текст руководства

    help\_text = "\*\*Руководство по программе\*\*\n\n" \

                "1. Нажмите кнопку \"Открыть файл\", чтобы выбрать DOCX файл для анализа.\n" \

                "2. Программа проанализирует семантику текста в файле.\n" \

                "3. Результаты анализа будут отображены в поле вывода.\n" \

                "4. Нажмите кнопку \"Помощь\", чтобы открыть это руководство.\n"

    help\_label = tk.Label(help\_window, text=help\_text)

    help\_label.pack()

# Создать основное окно GUI

root = tk.Tk()

root.title("Семантико-семантический анализ spaCy")

# Создать меню

menu\_bar = tk.Menu(root)

root.config(menu=menu\_bar)

# Создать меню "Файл"

file\_menu = tk.Menu(menu\_bar, tearoff=0)

file\_menu.add\_command(label="Открыть файл", command=open\_file)

file\_menu.add\_separator()

file\_menu.add\_command(label="Выход", command=root.destroy)

menu\_bar.add\_cascade(label="Файл", menu=file\_menu)

# Создать меню "Помощь"

help\_menu = tk.Menu(menu\_bar, tearoff=0)

help\_menu.add\_command(label="Помощь", command=show\_help)

menu\_bar.add\_cascade(label="Помощь", menu=help\_menu)

# Создать поле ввода для отображения результатов

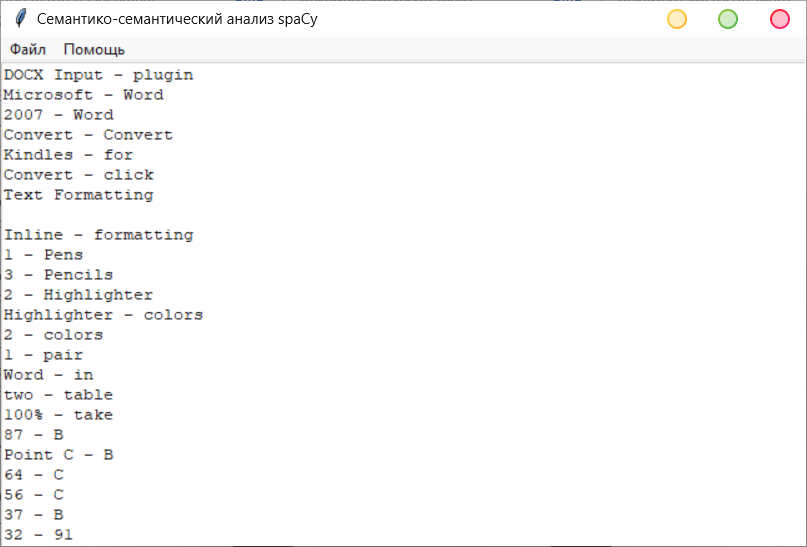
text\_box = tk.Text(root)

text\_box.pack()

# Запустить основное окно GUI

root.mainloop()

# Результат:



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