Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №3

По дисциплине: «Естественно-языковой интерфейс ИС»

Тема: «Разработка системы машинного перевода документов»

Выполнил:

Студент 4 курса

Группы ИИ-21

Пучинский А.А.

Проверил:

Булей Е.В.

Брест 2024

**Цель:** освоить на практике основные принципы машинного перевода документов.

**Ход работы:**

|  |  |  |
| --- | --- | --- |
| **№** | **Направление перевода** | **Предметная область** |
| 9 | Англо-русский | Научные статьи по медицине, Критика предметов изобразительного искусства |



* на входе – естественно-языковой текст на входном языке, подлежащий процедуре машинного перевода;
* подсчитать количество слов во входном тексте, количество переведенных слов, определить грамматическую информацию (теги частей речи и их расшифровка, для этого следует использовать функциональность лр. №3 весенний семестр).
* на выходе:
* перевод входного текста на выходной язык;
* упорядоченный по частоте встречаемости в тексте список слов и их переводов на выходной язык с грамматической информацией (вкладка 1) (можно использовать функциональность лр.

№1 весеннего семестра);

* построенное дерево синтаксического разбора выбранного предложения (вкладка 2).
* обеспечить наличие утилиты автоматического пополнения/корректировки полученного словаря (таблицы БД).
* обеспечить сохранение и распечатку результатов перевода и упорядоченных по частоте встречаемости в тексте списков слов и их переводов на выходной язык с грамматической информацией в файл формата txt кодировки Unicode.
* интерфейс системы должен быть простым и доступным для пользователей любого уровня, содержать понятный набор инструментов.

**Код программы:**

import tkinter as tk

from tkinter import filedialog, scrolledtext, messagebox

import fitz  # PyMuPDF для обработки PDF

from deep\_translator import GoogleTranslator  # Используем deep\_translator

import tempfile

import os

import threading

class PDFTranslator:

    def \_\_init\_\_(self, window):

        self.window = window

        self.window.title("PDF Переводчик")

        self.window.configure(bg="#F0F0F0")

        # Кнопка для открытия PDF

        self.open\_button = tk.Button(window, text="Открыть PDF", command=self.load\_pdf, bg="#4A90E2", fg="white", font=("Arial", 12, "bold"))

        self.open\_button.pack(pady=10)

        # Область для отображения текста PDF

        self.text\_area = scrolledtext.ScrolledText(window, wrap=tk.WORD, width=60, height=10, bg="#E8F0FE", fg="#333333", font=("Arial", 10))

        self.text\_area.pack(pady=10)

        # Кнопка для начала перевода

        self.translate\_button = tk.Button(window, text="Перевести", command=self.start\_translation, bg="#4A90E2", fg="white", font=("Arial", 12, "bold"))

        self.translate\_button.pack(pady=5)

        self.translate\_button.config(state=tk.DISABLED)

        # Кнопка для остановки перевода

        self.stop\_button = tk.Button(window, text="Стоп", command=self.stop\_translation, state=tk.DISABLED, bg="#FF5733", fg="white", font=("Arial", 12, "bold"))

        self.stop\_button.pack(pady=5)

        # Область для отображения переведённого текста

        self.translated\_area = scrolledtext.ScrolledText(window, wrap=tk.WORD, width=60, height=10, bg="#E8F0FE", fg="#333333", font=("Arial", 10))

        self.translated\_area.pack(pady=10)

        # Инициализация переводчика с deep\_translator

        self.translator = GoogleTranslator(source='en', target='ru')

        self.temp\_file = None

        self.stop\_translation\_flag = False

    def load\_pdf(self):

        file\_path = filedialog.askopenfilename(filetypes=[("PDF Files", "\*.pdf")])

        if file\_path:

            try:

                document = fitz.open(file\_path)

                self.temp\_file = tempfile.NamedTemporaryFile(delete=False, mode='w+', encoding='utf-8')

                for page in document:

                    page\_text = page.get\_text("text")

                    self.temp\_file.write(page\_text + '\n\n')

                self.temp\_file.flush()

                self.temp\_file.seek(0)

                # Очистка поля и отображение текста PDF

                self.text\_area.delete(1.0, tk.END)

                self.text\_area.insert(tk.END, self.temp\_file.read())

                self.translated\_area.delete(1.0, tk.END)

                self.translate\_button.config(state=tk.NORMAL)

            except Exception as e:

                messagebox.showerror("Ошибка", f"Не удалось открыть PDF: {e}")

    def start\_translation(self):

        """Запуск перевода в отдельном потоке."""

        self.stop\_translation\_flag = False

        translation\_thread = threading.Thread(target=self.perform\_translation)

        translation\_thread.start()

        self.stop\_button.config(state=tk.NORMAL)

        self.translate\_button.config(state=tk.DISABLED)

        self.open\_button.config(state=tk.DISABLED)

    def stop\_translation(self):

        """Остановка перевода."""

        self.stop\_translation\_flag = True

        self.stop\_button.config(state=tk.DISABLED)

    def perform\_translation(self):

        """Процесс перевода текста PDF."""

        if self.temp\_file:

            try:

                self.temp\_file.seek(0)

                self.translated\_area.delete(1.0, tk.END)

                self.translate\_button.config(text="Переводим...")

                for line in self.temp\_file:

                    if self.stop\_translation\_flag:

                        break

                    if line.strip():

                        try:

                            translated\_line = self.translator.translate(line.strip())  # перевод с английского на русский

                            self.translated\_area.insert(tk.END, translated\_line + '\n\n')

                        except Exception as e:

                            self.translated\_area.insert(tk.END, f"Ошибка перевода: {e}\n\n")

                        self.translated\_area.update\_idletasks()

                self.translate\_button.config(state=tk.NORMAL, text="Перевести")

                self.open\_button.config(state=tk.NORMAL)

                self.stop\_button.config(state=tk.DISABLED)

            except Exception as error:

                messagebox.showerror("Ошибка", f"Не удалось перевести текст: {error}")

                self.translate\_button.config(state=tk.NORMAL)

                self.stop\_button.config(state=tk.DISABLED)

                self.open\_button.config(state=tk.NORMAL)

        else:

            messagebox.showwarning("Внимание", "Сначала откройте PDF файл.")

    def \_\_del\_\_(self):

        """Удаление временного файла при завершении программы."""

        if self.temp\_file:

            try:

                os.remove(self.temp\_file.name)

            except Exception as e:

                print(f"Не удалось удалить временный файл: {e}")

if \_\_name\_\_ == "\_\_main\_\_":

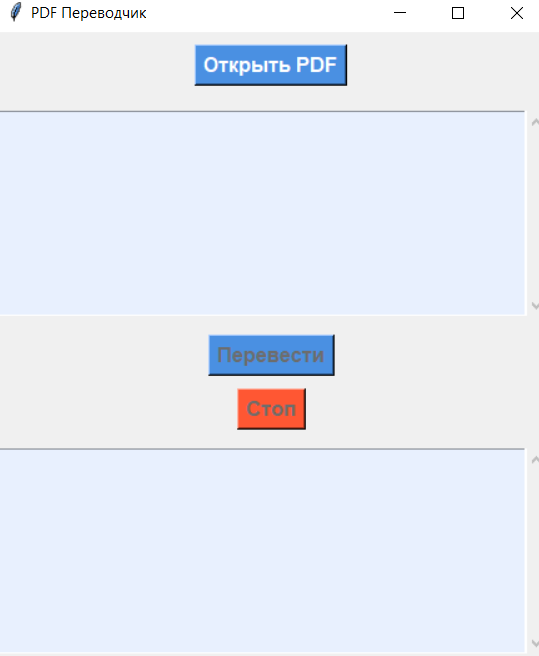
    main\_window = tk.Tk()

    app\_instance = PDFTranslator(main\_window)

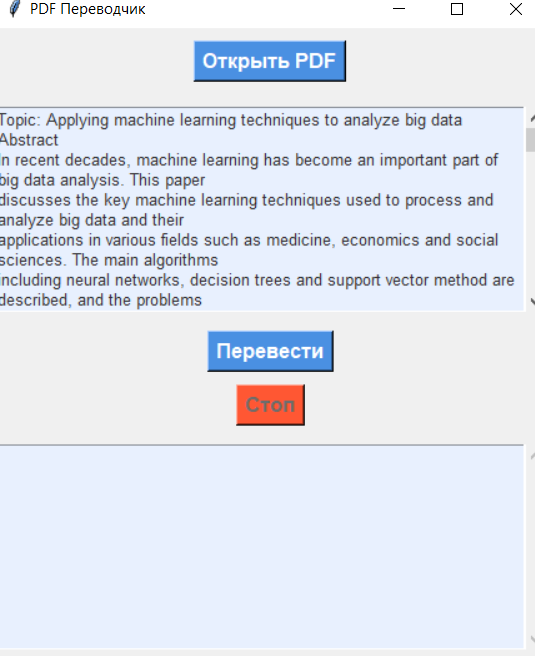
    main\_window.mainloop()

**Результат:**

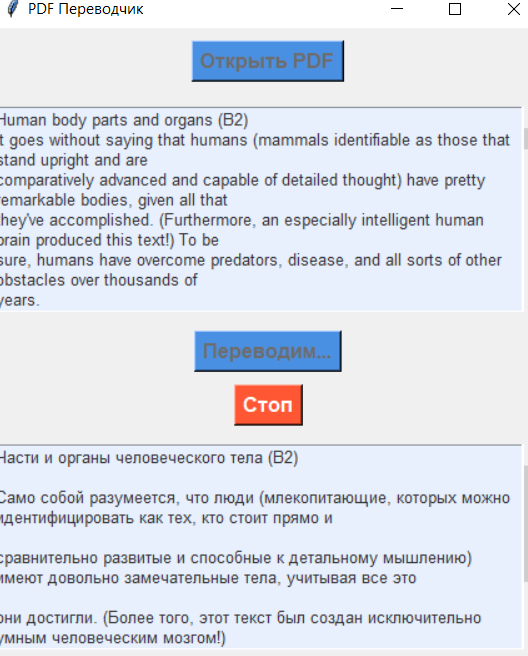
**Исходное главное меню:**



**Выбран файл:**



**Перевод:**



**Вывод:** в ходе выполнения лабораторной работы освоилна практике основные принципы машинного перевода документов.