Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №1

По дисциплине: «Естественно-языковой интерфейс ИС»

Тема: «Разработка информационно-поисковой системы и методы оценки качества ее работы»

Выполнил:

Студент 4 курса

Группы ИИ-21

Романко Н. А.

Проверила:

Булей Е. В.

Брест 2024

Цель: освоить принципы разработки прикладных сервисных программ для решения задачи автоматического лексического и лексико-грамматического анализа текста естественного языка.

Ход работы:

№	Сфера применения	Стратегия поиска	Язык
4	Сеть интернет	Векторная	Русский

- на входе – множество естественно-языковых текстов по которым осуществляется поиск;
- выделение ключевых слов документов осуществляется системой автоматически в соответствие с формулой 1.6;
- система должна позволять пользователю формулировать ЕЯ-запрос;
- на выходе – список документов, релевантных запросу пользователя, в соответствие с моделью поиска, согласно варианту;
- результаты поиска должны содержать: активную ссылку на документ, список слов запроса присутствующих в документе.различных форматов представления входных данных (TXT, RTF, PDF, DOC, DOCX).

Код программы:

Индексация файлов:

```
  def index_files(directory):
      index = {}
      texts = []
      for root, _, files in os.walk(directory):
          for file in files:
              if file.endswith('.pdf'):
                  path = os.path.join(root, file)
                  text = extract_text_from_pdf(path)
                  index[path] = text
                  texts.append(text)
              elif file.endswith('.docx'):
                  path = os.path.join(root, file)
                  text = extract_text_from_docx(path)
                  index[path] = text
                  texts.append(text)

      # Векторизация текстов
      vectorizer = TfidfVectorizer()
      tfidf_matrix = vectorizer.fit_transform(texts)
      return index, tfidf_matrix, vectorizer
```

Поиск совпадений:

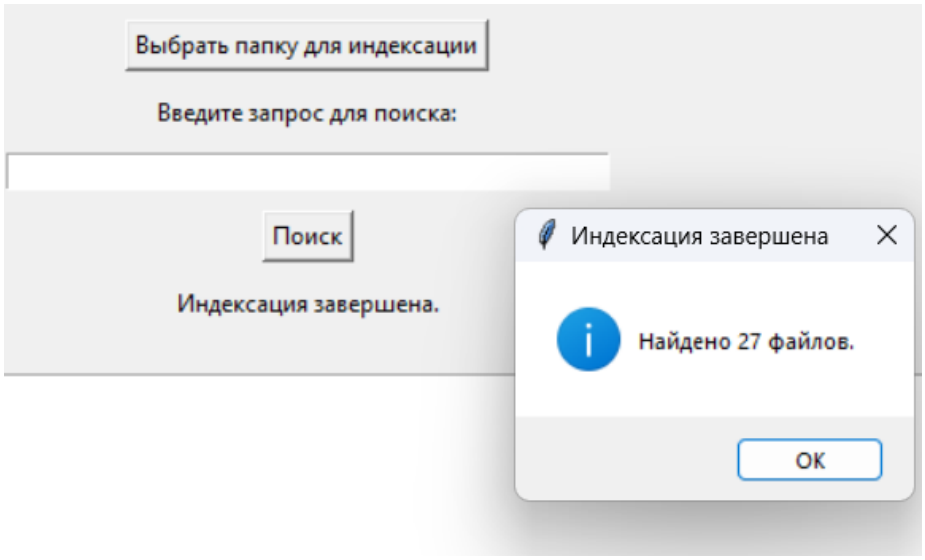
```
def search_index(index, tfidf_matrix, vectorizer, query):
    query_words = query.split()  # Разделяем запрос на слова
    results = {}
    for word in query_words:
        query_vector = vectorizer.transform([word])
        cosine_similarities = np.dot(tfidf_matrix, query_vector.T).toarray()  # косинусное сходство
        for i, path in enumerate(index.keys()):
            if cosine_similarities[i][0] > 0:
                if path not in results:
                    results[path] = {'texts': [], 'count': 0}
                results[path]['texts'].append(index[path])
                results[path]['count'] += index[path].lower().count(word.lower())
    return results
```

Вывод результатов:

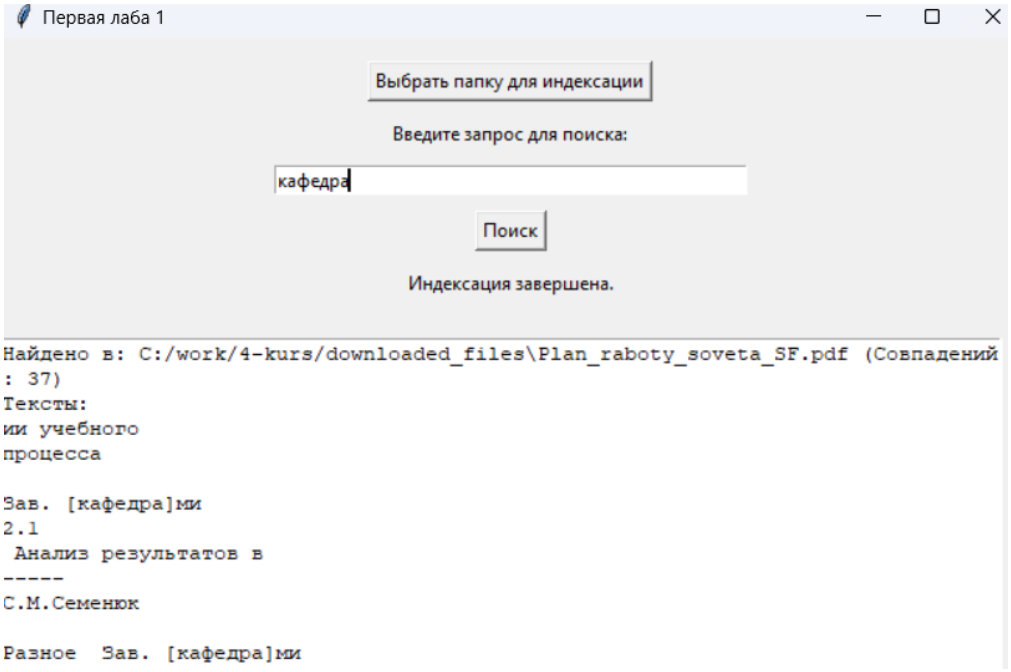
```
def search_files():
    query = query_entry.get()
    if not query:
        messagebox.showwarning("Предупреждение", "Введите запрос для поиска.")
        return
    results = search_index(index, tfidf_matrix, vectorizer, query)
    result_text.delete(1.0, tk.END)

    if results:
        for path, data in results.items():
            matched_texts = []
            for text in data['texts']:
                for word in query.split():
                    start_index = text.lower().find(word.lower())
                    while start_index != -1:
                        start_context = max(0, start_index - 30)  # 30 символов перед
                        end_context = min(len(text), start_index + len(word) + 30)  # 30 символов
после
                        matched_text = text[start_context:end_context].replace(word, f"[{word}]")
                        matched_texts.append(matched_text)
                        start_index = text.lower().find(word.lower(), start_index + 1)
            # Выводим результаты для файла, если есть совпадения
            if matched_texts:
                result_text.insert(tk.END, f"Найдено в: {path} (Совпадений:
{data['count']})\nТексты:\n")
                for matched_text in matched_texts:
                    result_text.insert(tk.END, f"{matched_text}\n")
                    result_text.insert(tk.END, "-----\n")
    else:
        result_text.insert(tk.END, "Результаты не найдены.\n")
```

Результат:

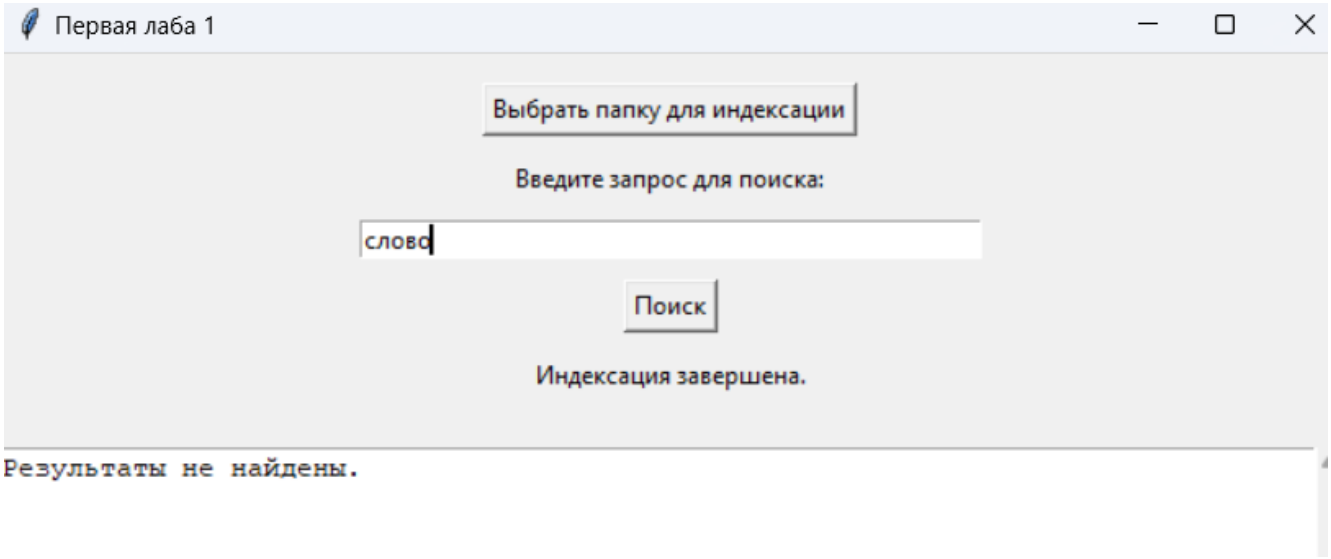
Индексация документов:



Нахождение совпадений:



Не нахождение совпадений:



Вывод: в ходе выполнения лабораторной работы освоил принципы разработки информационно-поисковых систем для поиска файлов с использованием естественного языка.