

Лексический анализ электронных писем с использованием библиотек Python

Голованова Маргарита Антоновна

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В статье рассматривается реализация лексического анализа текстов электронных писем с использованием языка программирования Python. Основное внимание уделяется применению библиотек обработки естественного языка для автоматического выделения лексем и распознавания именованных сущностей, таких как дата, время и место событий. Описывается практический пример анализа нескольких входящих электронных писем разной тематики в среде разработки PyCharm. Показаны возможности библиотек NLTK и Natasha для обработки текстовой информации на русском языке. Полученные результаты демонстрируют эффективность использования Python при решении прикладных задач анализа электронной корреспонденции.

Ключевые слова: Python, лексический анализ, обработка текста, электронные письма, NLP, Natasha.

Lexical analysis of email messages using Python libraries

Golovanova Margarita Antonovna

Sholom-Aleichem Priamursky State University

Student

Abstract

The article discusses the implementation of lexical analysis of email texts using the Python programming language. Special attention is paid to the use of natural language processing libraries for automatic tokenization and comprehensive named entity recognition, including extraction of dates, times, persons, organizations, and locations. Practical examples of processing multiple incoming email messages of different types in the PyCharm development environment are presented. The capabilities of the NLTK and, especially, the Natasha library for Russian-language text analysis are demonstrated. The results confirm the effectiveness of Python tools for solving applied problems of email text processing.

Keywords: Python, lexical analysis, text processing, email messages, NLP, Natasha.

1. Введение

1.1. Актуальность

В настоящее время электронная почта остается одним из основных способов передачи деловой и учебной информации. Значительная часть сообщений содержит сведения о событиях, встречах и мероприятиях, включая указание времени, даты и места. Ручная обработка таких данных является трудоемкой и подверженной ошибкам.

Использование методов обработки естественного языка позволяет автоматизировать анализ текстов электронных писем и извлекать из них структурированную информацию. Лексический анализ является базовым этапом такой обработки и играет ключевую роль в системах интеллектуального анализа текстов.

1.2 Обзор исследований

В научных публикациях рассматриваются различные подходы к автоматической обработке текстов на естественном языке, включая использование библиотек Python для токенизации, морфологического анализа, извлечения информации и распознавания именованных сущностей. Эти методы являются основой для лексического анализа электронной корреспонденции. В работе А.Г. Кромских анализируются подходы к использованию библиотеки Natasha для извлечения структурированной информации из текстов на русском языке, включая распознавание дат, времени и географических объектов, что непосредственно относится к обработке электронных писем [1]. Ж. С. Эгембердиева, Б. О. Кочконбаева и А.Ж. Алдосова в своём исследовании рассматривают обработку естественного языка с использованием Python, включая токенизацию, морфологический анализ и применение библиотек NLP для практических задач анализа текста [2]. А.Н. Исмуканова и Д.Н. Лавров демонстрируют применение библиотеки NLTK для сбора статистики встречаемости слов в задаче классификации научных текстов, что позволяет автоматизировать предварительный анализ и обработку текстовой информации [3]. А.В. Антонов рассматривает возможности использования NLTK для проведения сентимент-анализа, что позволяет выявлять эмоциональную окраску текстов и использовать эту информацию при анализе корреспонденции [4]. Е.А. Иванушкина и Н.М. Махина показывают применение библиотеки NLTK для автоматизации извлечения терминов и формирования глоссариев, что облегчает структурирование информации из больших текстовых массивов и подготовку данных для последующего анализа [5]. Анализ представленных исследований показывает, что современные методы NLP и использование библиотек Python (NLTK, Natasha и др.) активно развиваются и позволяют эффективно обрабатывать текстовые данные на русском языке. Вместе с тем задачи комплексного анализа электронных писем, включая автоматическое распознавание ключевых сущностей, требуют дальнейшего применения этих подходов в практических сценариях.

1.3 Цель исследования

Целью исследования является разработка и апробация программного решения на языке Python для лексического анализа электронных писем и автоматического распознавания временных и пространственных характеристик текста.

2. Материалы и методы

В ходе исследования использовалась интегрированная среда разработки PyCharm и язык программирования Python версии 3.10.

Для реализации лексического анализа применены следующие библиотеки:

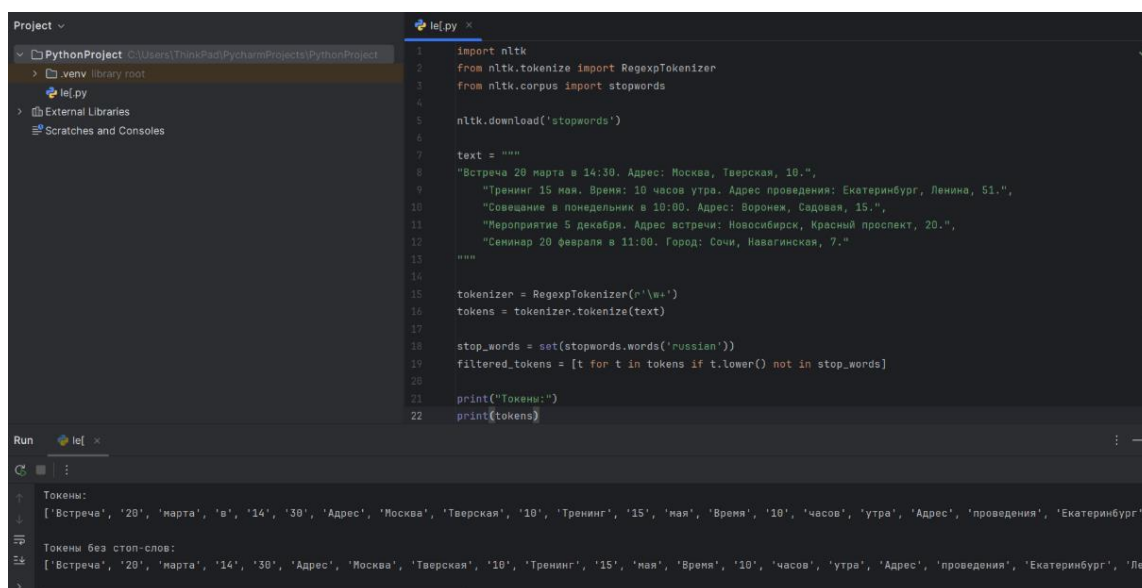
NLTK — для токенизации текста и удаления стоп-слов;

Natasha — для распознавания именованных сущностей (дата, время, место).

Метод исследования основан на практической реализации программного кода и анализе результатов обработки текстов электронных писем.

3. Результаты

На первом этапе подготовлены тексты пяти электронных писем, содержащих информацию о встрече, тренинге, совещании, мероприятии и семинаре. Далее выполнена токенизация текста с использованием библиотеки NLTK, в результате чего получены отдельные слова и предложения (рис. 1).



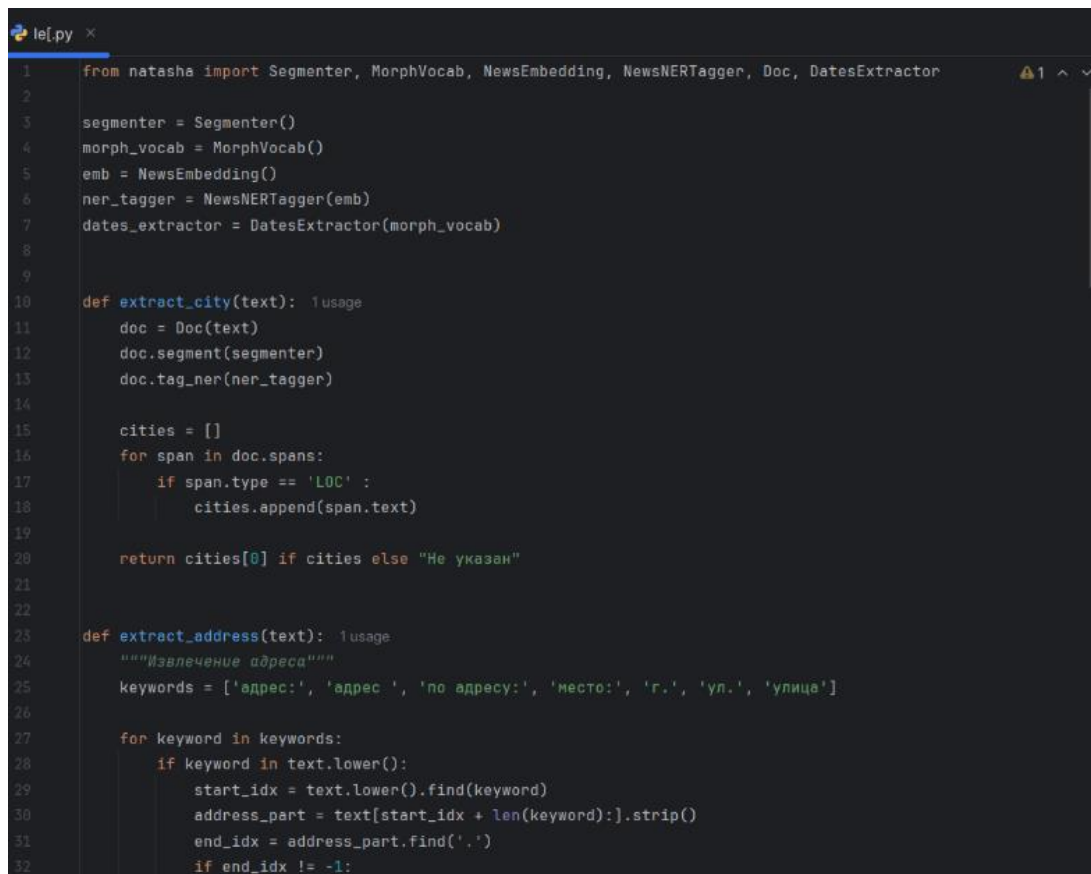
```
1 import nltk
2 from nltk.tokenize import RegexpTokenizer
3 from nltk.corpus import stopwords
4
5 nltk.download('stopwords')
6
7 text = """
8 "Встреча 28 марта в 14:30. Адрес: Москва, Тверская, 18.",
9 "Тренинг 15 мая. Время: 10 часов утра. Адрес проведения: Екатеринбург, Ленина, 51.",
10 "Совещание в понедельник в 10:00. Адрес: Воронеж, Садовая, 15.",
11 "Мероприятие 5 декабря. Адрес встречи: Новосибирск, Красный проспект, 20.",
12 "Семинар 20 февраля в 11:00. Город: Сочи, Навагинская, 7."
13 """
14
15 tokenizer = RegexpTokenizer(r'\w+')
16 tokens = tokenizer.tokenize(text)
17
18 stop_words = set(stopwords.words('russian'))
19 filtered_tokens = [t for t in tokens if t.lower() not in stop_words]
20
21 print("Токены:")
22 print(tokens)
```

Токены:
['Встреча', '28', 'марта', 'в', '14', '30', 'Адрес', 'Москва', 'Тверская', '18', 'Тренинг', '15', 'мая', 'Время', '10', 'часов', 'утра', 'Адрес', 'проведения', 'Екатеринбург', 'Ленина', '51', 'Совещание', 'в', 'понедельник', 'в', '10:00', 'Адрес', 'Воронеж', 'Садовая', '15', 'Мероприятие', '5', 'декабря', 'Адрес', 'встречи', 'Новосибирск', 'Красный', 'проспект', '20', 'Семинар', '20', 'февраля', 'в', '11:00', 'Город', 'Сочи', 'Навагинская', '7']

Токены без стоп-слов:
['Встреча', '28', 'марта', '14', '30', 'Адрес', 'Москва', 'Тверская', '18', 'Тренинг', '15', 'мая', 'Время', '10', 'часов', 'утра', 'Адрес', 'проведения', 'Екатеринбург', 'Ленин', '51', 'Совещание', 'понедельник', '10:00', 'Адрес', 'Воронеж', 'Садовая', '15', 'Мероприятие', '5', 'декабря', 'Адрес', 'встречи', 'Новосибирск', 'Красный', 'проспект', '20', 'Семинар', '20', 'февраля', '11:00', 'Город', 'Сочи', 'Навагинская', '7']

Рисунок 1- Результат токенизации текста электронного письма

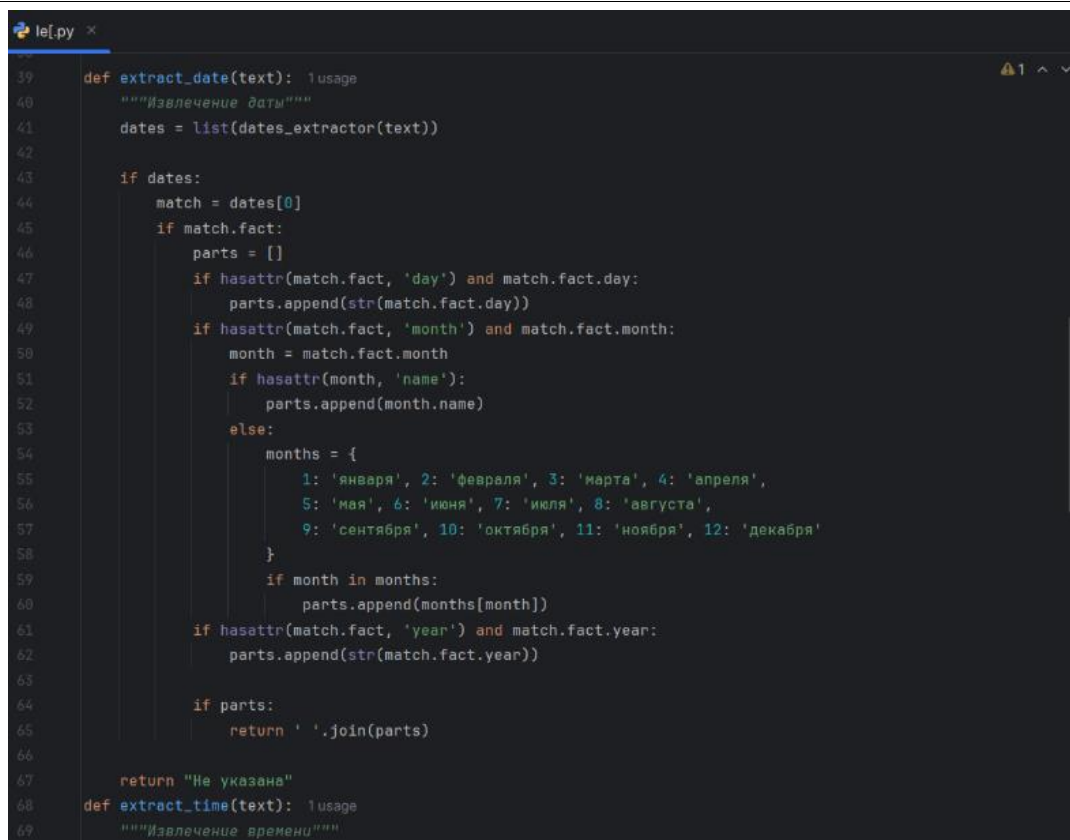
Основной этап исследования посвящен применению библиотеки Natasha для извлечения структурированной информации. Инициализация компонентов Natasha включает загрузку необходимых моделей для сегментации текста, морфологического анализа и распознавания именованных сущностей. Функция `extract_city()` использует NER-тэггер для идентификации географических названий, в то время как `extract_address()` осуществляет поиск по ключевым словам для выделения полных адресов (рис.2).



```
1 from natasha import Segmenter, MorphVocab, NewsEmbedding, NewsNERTagger, Doc, DatesExtractor
2
3 segmenter = Segmenter()
4 morph_vocab = MorphVocab()
5 emb = NewsEmbedding()
6 ner_tagger = NewsNERTagger(emb)
7 dates_extractor = DatesExtractor(morph_vocab)
8
9
10 def extract_city(text): 1usage
11     doc = Doc(text)
12     doc.segment(segmenter)
13     doc.tag_ner(ner_tagger)
14
15     cities = []
16     for span in doc.spans:
17         if span.type == 'LOC':
18             cities.append(span.text)
19
20     return cities[0] if cities else "Не указан"
21
22
23 def extract_address(text): 1usage
24     """Извлечение адреса"""
25     keywords = ['адрес:', 'адрес ', 'по адресу:', 'место:', 'г.', 'ул.', 'улица']
26
27     for keyword in keywords:
28         if keyword in text.lower():
29             start_idx = text.lower().find(keyword)
30             address_part = text[start_idx + len(keyword):].strip()
31             end_idx = address_part.find('.')
32             if end_idx != -1:
```

Рисунок 2- Инициализация компонентов Natasha и функции извлечения городов и адресов

Реализация функций временного анализа (рис. 3) демонстрирует комплексный подход к обработке дат и времени. `DatesExtractor` обеспечивает надежное распознавание различных форматов дат, а алгоритм поиска времени учитывает как цифровые, так и словесные форматы представления



```
39 def extract_date(text): 1 usage
40 """Извлечение даты"""
41 dates = list(dates_extractor(text))
42
43 if dates:
44     match = dates[0]
45     if match.fact:
46         parts = []
47         if hasattr(match.fact, 'day') and match.fact.day:
48             parts.append(str(match.fact.day))
49         if hasattr(match.fact, 'month') and match.fact.month:
50             month = match.fact.month
51             if hasattr(month, 'name'):
52                 parts.append(month.name)
53             else:
54                 months = {
55                     1: 'января', 2: 'февраля', 3: 'марта', 4: 'апреля',
56                     5: 'мая', 6: 'июня', 7: 'июля', 8: 'августа',
57                     9: 'сентября', 10: 'октября', 11: 'ноября', 12: 'декабря'
58                 }
59                 if month in months:
60                     parts.append(months[month])
61         if hasattr(match.fact, 'year') and match.fact.year:
62             parts.append(str(match.fact.year))
63
64         if parts:
65             return ' '.join(parts)
66
67     return "Не указана"
68 def extract_time(text): 1 usage
69 """Извлечение времени"""
```

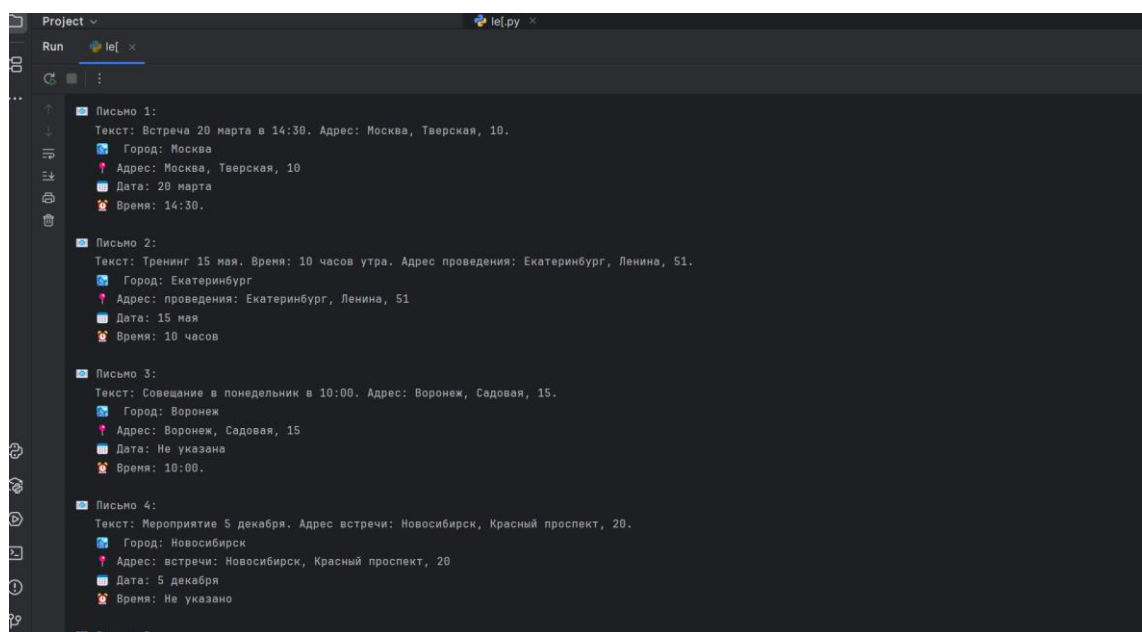
Рисунок 3 - Функции извлечения дат и времени из текста электронных писем

Тестирование системы организовано с использованием пяти разноформатных электронных писем (рис.4). Основной цикл обработки последовательно применяет разработанные функции извлечения сущностей к каждому тестовому примеру, что позволяет наглядно продемонстрировать работу всей системы в комплексе. Подобный подход к тестированию обеспечивает возможность оценки как корректности работы отдельных компонентов, так и системы в целом.

```
96
97 emails = [
98     "Встреча 20 марта в 14:30. Адрес: Москва, Тверская, 10.",
99     "Тренинг 15 мая. Время: 10 часов утра. Адрес проведения: Екатеринбург, Ленина, 51.",
00     "Совещание в понедельник в 10:00. Адрес: Воронеж, Садовая, 15.",
01     "Мероприятие 5 декабря. Адрес встречи: Новосибирск, Красный проспект, 20.",
02     "Семинар 20 февраля в 11:00. Город: Сочи, Навагинская, 7."
03 ]
04
05 print("=" * 80)
06 print("АНАЛИЗ ЭЛЕКТРОННЫХ ПИСЕМ")
07 print("Извлечение: город, адрес, дата, время")
08 print("=" * 80)
09
10 for i, email in enumerate(emails, 1):
11     print(f"\n✉ Письмо {i}:")
12     print(f"    Текст: {email}")
13
14     result = analyze_email(email)
15     print(f"    📍 Город: {result['город']}")
16     print(f"    📍 Адрес: {result['адрес']}")
17     print(f"    📅 Дата: {result['дата']}")
18     print(f"    🕒 Время: {result['время']}")
19
20
```

Рисунок 4 - Тестирование системы анализа на пяти примерах электронных писем

Результаты выполнения программы (рис.5), демонстрируют высокую эффективность разработанной системы в извлечении ключевых сущностей из электронных писем. Проведенный анализ пяти тестовых писем позволил оценить работу системы в реальных условиях и выявить как сильные стороны, так и направления для дальнейшего совершенствования.



```
Project v
Run le[py x
✉ Письмо 1:
Текст: Встреча 20 марта в 14:30. Адрес: Москва, Тверская, 10.
📍 Город: Москва
📍 Адрес: Москва, Тверская, 10
📅 Дата: 20 марта
🕒 Время: 14:30.

✉ Письмо 2:
Текст: Тренинг 15 мая. Время: 10 часов утра. Адрес проведения: Екатеринбург, Ленина, 51.
📍 Город: Екатеринбург
📍 Адрес: проведения: Екатеринбург, Ленина, 51
📅 Дата: 15 мая
🕒 Время: 10 часов

✉ Письмо 3:
Текст: Совещание в понедельник в 10:00. Адрес: Воронеж, Садовая, 15.
📍 Город: Воронеж
📍 Адрес: Воронеж, Садовая, 15
📅 Дата: Не указана
🕒 Время: 10:00.

✉ Письмо 4:
Текст: Мероприятие 5 декабря. Адрес встречи: Новосибирск, Красный проспект, 20.
📍 Город: Новосибирск
📍 Адрес: встречи: Новосибирск, Красный проспект, 20
📅 Дата: 5 декабря
🕒 Время: Не указано

✉ Письмо 5:
```

Рисунок 5 - Результаты извлечения сущностей из тестовых электронных писем

4. Выводы

В результате проведенного исследования реализована программная модель лексического анализа электронных писем с использованием библиотек Python в рамках выполнения работ по дисциплине «Информационные технологии». Показано, что применение NLTK и Natasha позволяет эффективно обрабатывать тексты на русском языке и автоматически извлекать ключевую информацию.

Разработанный подход может быть использован в учебных целях, а также при создании интеллектуальных систем обработки электронной почты и текстовой информации.

Библиографический список

1. Кромских А. Г. Подходы к использованию библиотеки Natasha для извлечения структурированной информации из текстов на русском языке // Актуальные вопросы эксплуатации систем охраны и защищённых телекоммуникационных систем. Сборник материалов Всероссийской научно-практической конференции. Воронеж, 2021. С. 23–25.
2. Эгембердиева Ж. С., Кочконбаева Б. О., Алдосова А. Ж. Обработка естественного языка с использованием языка программирования Python // Инновационный потенциал развития науки в современном мире: достижения и инновации. Сборник материалов X Международной научно-практической конференции. Уфа, 2023. С. 155–165.
3. Исмуканова А. Н., Лавров Д. Н. Библиотека "NLTK" для сбора статистики встречаемости слов в задаче классификации научных текстов // Математическое и компьютерное моделирование. Сборник материалов V Международной научной конференции, посвященной памяти Р. Л. Долганова. 2017. С. 115–117.
4. Антонов А. В. Возможности использования библиотеки NLTK для сентимент-анализа // Теоретические и практические аспекты развития науки в современном мире. Сборник статей международной научной конференции. Санкт-Петербург, 2023. С. 26–28.
5. Иванушкина Е. А., Махина Н. М. Использование библиотеки NLTK для автоматизации извлечения терминов и формирования глоссариев // Актуальные проблемы современной науки: взгляд молодых учёных. Материалы Национальной научно-практической студенческой конференции. Брянск, 2024. С. 54–58.