

УДК 517.977.1

Обзорная статья

<https://doi.org/10.32603/2071-8985-2024-17-1-48-54>**Применение метода распознавания именованных сущностей  
в сфере информационных технологий для анализа задач  
по разработке программного обеспечения****Л. А. Куценок<sup>✉</sup>, Ю. А. Кораблёв**Санкт-Петербургский государственный электротехнический университет  
«ЛЭТИ» им. В. И. Ульянова (Ленина), Санкт-Петербург, Россия<sup>✉</sup>leo@sochi.com

**Аннотация.** Рассматривается применение метода распознавания именованных сущностей (NER) для анализа задач разработки программного обеспечения. Описываются основные методы анализа трудозатрат сотрудников IT-компаний, а также технологии, библиотеки и фреймворки NER – SpaCy, DeepPavlov, Natasha, Pullenti и т. п., которые могут быть использованы для извлечения информации об именованных сущностях из текстовых данных на русском языке. Предлагаются полезные ресурсы и подходы для анализа задач разработки, способствующие улучшению планирования, управления проектами и сокращению времени разработки программного обеспечения.

**Ключевые слова:** распознавание именованных сущностей (NER), анализ задач разработки программного обеспечения, библиотеки NER, технологии обработки естественного языка, извлечение информации, планирование проектов, управление проектами, сокращение времени разработки

**Для цитирования:** Куценок Л. А., Кораблёв Ю. А. Применение метода распознавания именованных сущностей в сфере информационных технологий для анализа задач по разработке программного обеспечения // Изв. СПбГЭТУ «ЛЭТИ». 2024. Т. 17, № 1. С. 48–54. doi: 10.32603/2071-8985-2024-17-1-48-54.

**Конфликт интересов.** Авторы заявляют об отсутствии конфликта интересов.

Review article

**Application of the Named Entity Recognition Method in Information Technology  
to Analysis of Software Development Tasks****L. A. Kutsenok<sup>✉</sup>, Yu. A. Korablev**

Saint Petersburg Electrotechnical University, Saint Petersburg, Russia

<sup>✉</sup>leo@sochi.com

**Abstract.** Explores the application of the Named Entity Recognition method to analyze software development tasks. It describes the main methods of labor-intensive analysis of IT companies, as well as NER technologies, libraries and frameworks such as SpaCy, DeepPavlov, Natasha and Pullenti, which can be used for extracting information about named entities from text data in Russian. This article offers useful resources and approaches for analyzing development tasks, which helps to improve planning, project management and reduce development time.

**Keywords:** Named Entity Recognition (NER), software development task analysis, NER libraries, Natural Language Processing (NLP) technologies, information extraction, project planning, project management, reduction of development time

**For citation:** Kutsenok L. A., Korablev Yu. A. Application of the Named Entity Recognition Method in Information Technology to Analysis of Software Development Tasks // LETI Transactions on Electrical Engineering & Computer Science. 2024. Vol. 17, no. 1. P. 48–54. doi: 10.32603/2071-8985-2024-17-1-48-54.

**Conflict of interest.** The authors declare no conflicts of interest.

**Введение.** В современном мире разработка программного обеспечения стала неотъемлемой частью информационных технологий, играющих важную роль во многих сферах деятельности. Успешное выполнение проектов по разработке программного обеспечения требует точного анализа задач, планирования ресурсов и эффективного управления. При этом эти задачи часто связаны с извлечением и анализом различных типов информации, включая типы задачи, заинтересованные стороны и организации, характеристики разрабатываемых продуктов и других значимых объекты.

В последние годы методы распознавания именованных сущностей (Named Entity Recognition, NER) получило значительное внимание в области обработки естественного языка. NER – это метод анализа текста, который позволяет автоматически определить и классифицировать именованные сущности в текстовых данных. Например, для текста «Сотрудник Иванов И. И. выполнил задачу по дизайну и разработке главной страницы сайта 10 февраля» технология NER может извлечь следующие ключевые сущности:

- актер: Сотрудник Иванов И. И.;
- тип задачи: дизайн, разработка;
- объект задачи: главная страница сайта;
- дата: 10 февраля.

Цель данной статьи состоит в рассмотрении применения метода распознавания именованных сущностей в сфере информационных технологий для анализа задач в области разработки программного обеспечения (ПО). В статье будут приведены технологии, библиотеки и фреймворки NER, которые могут использоваться для извлечения и классификации именованных сущностей в текстовых данных на русском языке. Также будут рассмотрены процессы, позволяющие обучать модели NER для конкретных задач разработки ПО.

Преимущества использования метода распознавания именованных сущностей в анализе задач разработки программного обеспечения многообразны. Во-первых, NER позволяет автоматически извлекать важную информацию из текстов, что помогает ускорить и улучшить процесс анализа задач. Во-вторых, NER обеспечивает более

точное и структурированное понимание текстовых данных, что способствует более эффективному планированию и управлению проектами разработки ПО. Кроме того, NER позволяет автоматически классифицировать именованные сущности, что может быть полезно для автоматического определения ключевых акторов, местоположений или дат в задачах разработки.

**Методы анализа трудозатрат сотрудников ИТ-компаний.** Анализ задач в разработке программного обеспечения может осуществляться различными подходами. Данные подходы можно выделить в три основные категории: математические методы, экспертные методы и методы, основанные на технологиях машинного обучения.

Экспертные методы анализа задач основаны на знаниях и опыте экспертов в области разработки ПО. Эксперты проводят анализ задач на основе своих знаний, интуиции и профессионального опыта. Они могут использовать различные методы – беседы с разработчиками, экспертные оценки, декомпозицию задач и другие техники для выявления ключевых аспектов и потенциальных проблем. Однако важно отметить, что экспертные методы требуют, в первую очередь, участия эксперта в процессе анализа результатов, а следовательно, не могут быть автоматизированы. Таким образом, методы экспертного анализа могут быть рассмотрены как способ составления вывода на основе полученных из разных источников данных (в том числе рассматриваемых далее).

Математические методы анализа задач включают в себя использование формализованных моделей, алгоритмов и статистических методов. В предыдущих публикациях [1] были рассмотрены некоторые из этих методов и их функционирование в современных системах управления ИТ-проектами, а также реализация соответствующего инструмента. Эти методы обеспечивают рациональный и строго структурированный подход к анализу задач разработки программного обеспечения, но не могут сделать вывод о качестве оценки трудозатрат в ходе анализа по тем или иным направлениям работы сотрудников на проекте, например о принадлежности задачи к определенному роду работ (например, проектирование, дизайн,

разработка, тестирование), типу задачи (например, создается ли новый экран приложения, новая функция или исправляется обнаруженная ошибка) и другим атрибутам задач в системах управления проектами. Некоторые из этих методов:

- диаграмма сгорания задач;
- диаграмма отношения плановых трудозатрат к фактическим;
- таблица установки максимальных трудозатрат по сотрудникам и проектам;
- таблица нормирования труда. Для решения данной проблемы и более точного анализа результатов оценки трудозатрат могут применяться методы машинного обучения и нейронных сетей, способных обрабатывать постановки задач для извлечения дополнительной информации и атрибутов.

К таким способам анализа постановок задач в текстовом виде из систем управления проектами относится метод NER. Он позволяет автоматически извлекать и классифицировать именованные сущности текстовых данных, что дает возможность более эффективного анализа и понимания информации, связанной с разработкой ПО.

NER работает на основе машинного обучения и использует различные методы и модели для распознавания и классификации именованных сущностей в тексте. Основная задача NER – это определение сущностей – имен людей, названий организаций, местоположений, дат и других значимых объектов. Это позволяет автоматически извлекать важную информацию из текстовых данных, связанных с задачами разработки программного обеспечения.

Существует несколько популярных технологий, библиотек и фреймворков NER, которые могут быть применены для анализа задач разработки программного обеспечения на русском языке [2], [3]. Рассмотрим некоторые из них:

- SpaCy [4]: мощная и эффективная библиотека обработки естественного языка, которая поддерживает NER. Она предоставляет готовые модели для распознавания именованных сущностей на русском языке, которые можно использовать для анализа текстовых данных, связанных с разработкой ПО.
- DeepPavlov [5]: библиотека и фреймворк глубокого обучения, который также предоставляет поддержку для NER. Она содержит предварительно обученные модели NER для русского языка, которые можно использовать для извлечения и

классификации именованных сущностей в задачах разработки ПО.

- Natasha [6]: библиотека на языке Python, которая специализируется на обработке текстов на русском языке. Она содержит инструменты для NER, которые позволяют извлекать информацию об именованных сущностях именах, организациях и географических местах из текстовых данных, связанных с разработкой программного обеспечения.

- Pullenti [7]: библиотека для обработки естественного языка (NLP), разработанная специально для русского языка. Она предоставляет мощные инструменты для выполнения различных задач, связанных с анализом текста, извлечением информации и распознаванием именованных сущностей. Библиотека Pullenti обладает широким функционалом, который включает в себя распознавание именованных сущностей, извлечение дат, адресов, организаций, географических объектов, числовой информации и других важных элементов из текста. Она также поддерживает лемматизацию, морфологический анализ и синтаксический анализ предложений, что позволяет проводить более глубокий анализ текстов. Одна из ключевых особенностей Pullenti – его способность работать с многозначностью русского языка. Библиотека учитывает контекст и предлагает наиболее вероятную интерпретацию значения слова или выражения в данном контексте, что повышает точность и надежность обработки текстовых данных.

Применение NER в анализе задач разработки программного обеспечения может иметь множество преимуществ. Во-первых, автоматическое извлечение и классификация именованных сущностей позволяет сократить время и усилия, затрачиваемые на ручной анализ текстовых данных после использования математических моделей экспертами. Во-вторых, это позволяет получить более точное и структурированное представление информации, что облегчает планирование и управление проектами разработки ПО.

Однако необходимо учитывать и ограничения NER в анализе задач разработки программного обеспечения. В некоторых случаях, особенно при наличии специфической терминологии или нестандартных именованных сущностей, требуется дополнительная настройка и обучение моделей NER для достижения оптимальных результатов на конкретном проекте.

**Процесс обучения NER-моделей для анализа трудозатрат сотрудников ИТ-компаний.** Процесс обучения NER для определенной предметной области включает несколько ключевых шагов [8]. Во-первых, необходимо собрать и подготовить аннотированный набор данных, включающий примеры текстов с указанием именованных сущностей, которые требуется распознать. Эти данные должны быть размечены экспертами, которые определяют границы и классификацию именованных сущностей в тексте. Пример результата данной разметки, реализованной в среде WebAnno [9], представлен на рис. 1.

Далее с использованием подготовленного набора данных проводится процесс обучения модели. Обучение NER может осуществляться с применением различных алгоритмов машинного обучения – скрытых марковских моделей (HMM), условных случайных полей (CRF) или нейронных сетей. Во время обучения модель анализирует текстовые данные и выявляет закономерности, связанные с именованными сущностями, чтобы научиться их правильно распознавать и классифицировать.

Однако для достижения высокой точности и эффективности модели NER требуется не только обучение на аннотированных данных, но и проведение процесса настройки и оптимизации модели. Оно может включать выбор подходящих признаков для распознавания именованных сущностей, настройку параметров модели, а также проведение кросс-валидации и оценку качества модели на отложенных данных.

Кроме того, важным аспектом обучения NER служит использование достаточно объемного и разнообразного набора данных. Большой объем данных позволяет модели обучаться на различных типах текстов и разнообразных именованных сущностях, что повышает ее способность обоб-

щать и правильно распознавать именованные сущности в новых текстах. Данный аспект особенно важен в контексте разработки программного обеспечения и проектного управления, так как объема данных, полученного из систем управления проектами, может быть недостаточно для обучения приемлемо точной модели.

Важно отметить, что обучение NER может быть требовательным к вычислительным ресурсам и времени. Обработка больших объемов данных и настройка сложных моделей может потребовать значительных вычислительных мощностей, поэтому эффективное использование аппаратных ресурсов и оптимизация процесса обучения особенно значимы при тренировке модели NER.

Схематическое представление данного процесса обучения представлено на рис. 2.

Важно отметить, что исследования в области применения NER в сфере разработки ПО показывают возможность адаптации современных моделей к рассматриваемой сфере на примере анализа интернет-ресурсов разработчиков для поиска ответов на вопросы, связанные с программированием и разработкой программного обеспечения [10], так и в других смежных сферах [11], а следовательно, целесообразно рассматривать обучение моделей NER в контексте анализа трудозатрат сотрудников на реализацию задач.

**Обзор технологий извлечения именованных сущностей.** Рассмотрим перечисленные ранее технологии, а также их особенности.

Библиотека SpaCy относится к популярным инструментам обработки естественного языка и включает в себя модуль распознавания именованных сущностей EntityRecognizer. SpaCy предоставляет готовые модели для NER на различных языках, включая русский (модель ru\_core\_news\_(sm, md,lg) [12], обученная на

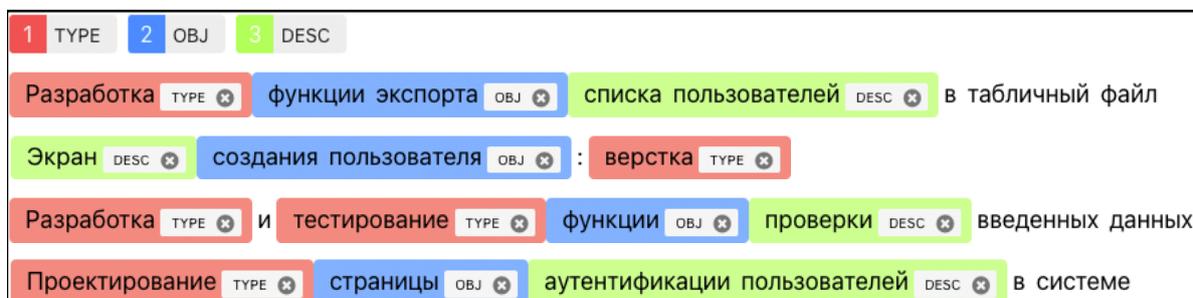


Рис. 1. Пример результата разметки темы задачи для извлечения именованных сущностей:

TYPE – тип задачи, OBJ – объект задачи, DESC – описание объекта задачи

Fig. 1. Example of task topic annotation for named entity recognition: TYPE – task type,

OBJ – task object, DESC – task object description

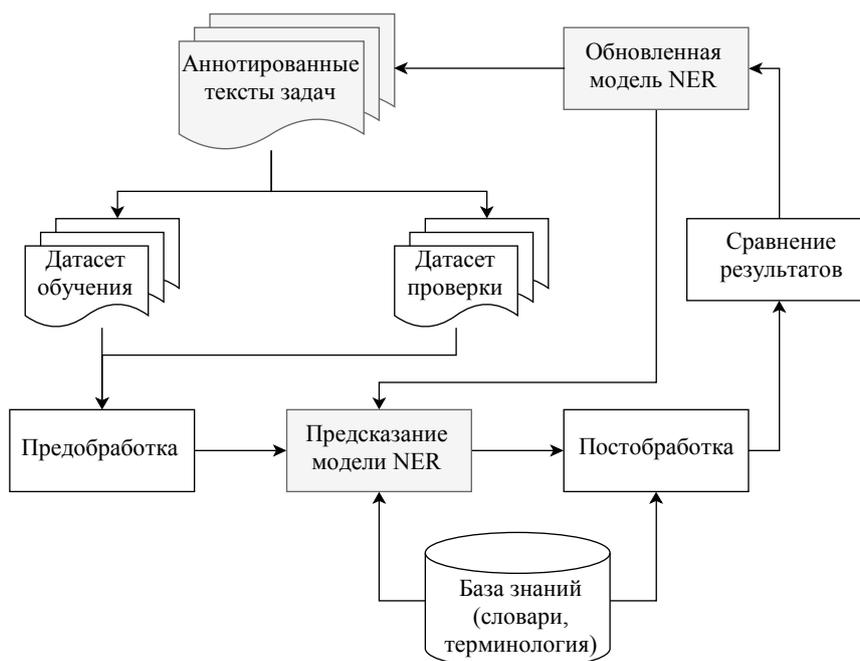


Рис. 2. Схематическое описание процесса обучения NER-модели  
Fig. 2. Typical process flow for NER training

текстах новостных ресурсов). SpaCy обеспечивает не только извлечение и классификацию именованных сущностей, но и предоставляет контекстную информацию о них – такую, как тип и связи между сущностями. Благодаря этому появляется возможность получить более полное представление о взаимодействии именованных сущностей в тексте и использовать эту информацию для более глубокого анализа задач разработки программного обеспечения.

Библиотека Natasha – это библиотека на языке Python, специализирующаяся на обработке текстов на русском языке. Она предоставляет инструменты для различных задач NLP, включая распознавание именованных сущностей. Natasha включает предварительно обученные модели NER для русского языка, которые можно использовать для анализа текстов, связанных с разработкой ПО. Библиотека Natasha обладает хорошей скоростью работы и простотой использования. Она предоставляет готовые функции для извлечения и классификации именованных сущностей, а также возможность настройки моделей NER в соответствии с конкретными потребностями проекта разработки ПО. Базовая модель Slovnet [13] была обучена на текстах новостных ресурсов.

Фреймворк DeepPavlov [14] – это набор инструментов, разработанных на базе глубокого обучения и обработки естественного языка. Она

предоставляет множество инструментов и моделей для NER, включая модели для русского языка. DeepPavlov позволяет обучать собственные модели NER на основе имеющихся данных или использовать предварительно обученные модели для дообучения, например, анализу текстовых данных, связанных с разработкой программного обеспечения. Фреймворк DeepPavlov предоставляет гибкую настройку моделей NER и возможность интеграции с другими инструментами и библиотеками, что делает его мощным средством анализа задач разработки программного обеспечения. Кроме того, DeepPavlov активно поддерживается сообществом и имеет развитую документацию, что облегчает использование и интеграцию в проекты разработки ПО, а уже обученные базовые модели, например `ner_rus_bert`, позволяют быстро начать тестирование прототипов программных решений.

Набор библиотек Pullenti предназначен для анализа неструктурированной информации и имеет доступные реализации на многих языках программирования – Python, C++, Java, JavaScript, C# и т. п. В состав Pullenti входит библиотека лингвистического анализа, включающего в себя морфологический анализ, извлечение именованных сущностей, семантический анализ, а также ряд процедур обработки текста [15]. Pullenti не включает в себя средства машинного обучения, а использует алгоритмы анализа данных для раз-

личных типов именованных сущностей, таким образом достигая высокой точности на ряде известных типов. Однако по этой же причине адаптация используемых алгоритмов в иную предметную область представляет собой сложную задачу по сравнению с рассмотренными ранее технологиями. Кроме того, Pullenti, в отличие от других рассмотренных технологий, имеет ограничения на коммерческое использование, ввиду чего его применение может стать проблематичным в некоторых ситуациях.

**Заключение.** Применение метода распознавания именованных сущностей при анализе задач разработки программного обеспечения может быть полезным инструментом, который позволит повысить эффективность и точность анализа данных, связанных с проектами разработки ПО. NER позволяет извлекать и классифицировать именованные сущности – типы задач, объекты, свойства и др., из текстовых данных, что облегчает процесс анализа, понимание и управление задачами.

Однако следует учитывать некоторые ограничения и вызовы, связанные с применением NER в анализе задач разработки программного обеспечения. Это необходимость в обширных наборах данных для обучения и настройки моделей, а также возможность некорректного распознавания или классификации именованных сущностей при отсутствии явных контекстуальных указателей. Также важно учитывать конфиденциальность и безопасность данных при использовании NER для анализа текстов.

Для реализации инструмента для анализа трудозатрат сотрудников ИТ-компаний с применением технологии распознавания именованных сущностей потребуются дополнительные исследования, направленные на подготовку моделей для библиотек и фреймворков SpaCy, Natasha, DeepPavlov и Pullenti. По итогам создания данных моделей и сравнения результатов их работы будет возможно сделать выбор в пользу одной из этих технологий для включения ее в разрабатываемый инструмент анализа трудозатрат.

#### Список литературы

1. The problem of man-hour distribution in modern task managers / L. A. Kutsenok, N. V. Razmochaeva, V. P. Semenov, A. A. Bezrukov // IEEE Conf. of Russian Young Researchers in Electrical and Electronic Engin. (EIConRus). St. Petersburg and Moscow, Russia: IEEE, 2020. С. 381–385. doi: 10.1109/EIConRus49466.2020.9039149.
2. Селиверстов И. П., Воробьев К. А., Петров А. В. Сравнение подходов к поиску именованных сущностей // Сб. тр. конф. «Математические модели техники, технологий и экономики». СПб.: ПОЛИТЕХ-ПРЕСС, 2019. С. 110–113.
3. Mandravickaitė J., Krilavičius T. Testing performance of NER models for Russian // Intern. J. of Design, Analysis and Tools for Integrated Circuits and Systems. 2021. Vol. 10, no. 1. С. 55–58.
4. SpaCy. Industrial-Strength Natural Language Processing. URL: <https://spacy.io/> (дата обращения 20.05.2023).
5. DeepPavlov. An open-source conversational AI framework. URL: <https://deeppavlov.ai/> (дата обращения 20.05.2023).
6. Проект Natasha – набор Python-библиотек для обработки текстов на естественном русском языке. URL: <https://natasha.github.io/> (дата обращения 20.05.2023).
7. Pullenti. URL: <https://www.pullenti.ru/> (дата обращения 20.05.2023).
8. Vasiliev Y. Natural language processing with Python and SpaCy: A practical introduction. No Starch Press, 2020. 216 p.
9. Software-specific named entity recognition in software engineering social content / D. Ye, Z. Xing, C. Y. Foo, Z. Q. Ang, J. Li, N. Kapre // IEEE 23<sup>rd</sup> Intern. Conf. on Software Analysis, Evolution, and Reengineering (SANER). Osaka, Japan: IEEE, 2016. Т. 1. С. 90–101. doi: 10.1109/SANER.2016.10.
10. Leser U., Hakenberg J. What makes a gene name? Named entity recognition in the biomedical literature // Briefings in Bioinformatics. 2005. Т. 6, № 4. С. 357–369.
11. WebAnno: A flexible, web-based and visually supported system for distributed annotations / S. Muhie, I. Gurevych, R. E. de Castilho, C. Biemann // Proc. of the 51<sup>st</sup> Ann. Meeting of the Association for Computational Linguistics: System Demonstrations. Sofia, Bulgaria: Computer Science, 2013. С. 1–6.
12. SpaCy. URL: [https://spacy.io/models/ru#ru\\_core\\_news](https://spacy.io/models/ru#ru_core_news) (дата обращения 10.07.2023).
13. Natasha / slovnet. URL: <https://github.com/natasha/slovnet> (дата обращения 10.07.2023).
14. Mozharova V. A., Loukachevitch N. V. Combining knowledge and CRF-based approach to named entity recognition in Russian // 5<sup>th</sup> Intern. Conf. on Analysis of Images, Social Networks and Texts, AIST 2016. Yekaterinburg, Russia, 2016: Springer International Publishing, 2017. С. 185–195.
15. Система PullEnti – извлечение информации из текстов естественного языка и автоматизированное построение информационных систем / О. В. Золотарёв, М. М. Шарнин, С. В. Клименко, К. И. Кузнецов // Тр. Междунар. науч. конф.: в 2 т. «Ситуационные центры и информационно-аналитические системы класса 4I для задач мониторинга и безопасности» (SCVRT2015–16). Т. 2. Протвино: АНО «Институт физико-технической информатики», 2016. С. 28–35.

### Информация об авторах

**Куценок Леонид Алексеевич** – аспирант кафедры автоматизации и процессов управления СПбГЭТУ «ЛЭТИ».

E-mail: leo@sochi.com

**Кораблёв Юрий Анатольевич** – канд. техн. наук, доцент кафедры автоматизации и процессов управления СПбГЭТУ «ЛЭТИ».

E-mail: juri.korablev@gmail.com

<https://orcid.org/0000-0002-2432-7169>

### References

1. The problem of man-hour distribution in modern task managers / L. A. Kutsenok, N. V. Razmochaeva, V. P. Semenov, A. A. Bezrukov // IEEE Conf. of Russian Young Researchers in Electrical and Electronic Engin. (EIConRus). St. Petersburg and Moscow, Russia: IEEE, 2020. S. 381–385. doi: 10.1109/EIConRus49466.2020.9039149.
2. Seliverstov I. P., Vorob'ev K. A., Petrov A. V. Svravnenie podhodov k poisku imenovannyh sushhnostej // Sb. tr. konf. «Matematicheskie modeli tehniki, tehnologij i jekonomiki». SPb.: POLITEH-PRESS, 2019. S. 110–113. (In Russ.).
3. Mandravickaitė J., Krilavičius T. Testing performance of NER models for Russian // Intern. J. of Design, Analysis and Tools for Integrated Circuits and Systems. 2021. Vol. 10, no. 1. S. 55–58.
4. SpaCy. Industrial-Strength Natural Language Processing. URL: <https://spacy.io/> (data obrashhenija 20.05.2023).
5. DeepPavlov. An open-source conversational AI framework. URL: <https://deeppavlov.ai/> (data obraschenija 20.05.2023).
6. Proekt Natasha – nabor Python-bibliotek dlja obrabotki tekstov na estestvennom russkom jazyke. URL: <https://natasha.github.io/> (data obraschenija 20.05.2023). (In Russ.).
7. Pullenti. URL: <https://www.pullenti.ru/> (data obraschenija 20.05.2023).
8. Vasiliev Y. Natural language processing with Python and SpaCy: A practical introduction. No Starch Press, 2020. 216 p.
9. Software-specific named entity recognition in software engineering social content / D. Ye, Z. Xing, C. Y. Foo, Z. Q. Ang, J. Li, N. Kapre // IEEE 23<sup>rd</sup> Intern. Conf. on Software Analysis, Evolution, and Reengineering (SANER). Osaka, Japan: IEEE, 2016. T. 1. S. 90–101. doi: 10.1109/SANER.2016.10.
10. Leser U., Hakenberg J. What makes a gene name? Named entity recognition in the biomedical literature // Briefings in Bioinformatics. 2005. T. 6. № 4. S. 357–369.
11. WebAnno: A flexible, web-based and visually supported system for distributed annotations / S. Muhie, I. Gurevych, R. E. de Castilho, C. Biemann // Proc. of the 51<sup>st</sup> Annual Meeting of the Association for Computational Linguistics: System Demonstrations. Sofia, Bulgaria: Computer Science, 2013. S. 1–6.
12. SpaCy. URL: [https://spacy.io/models/ru#ru\\_core\\_news](https://spacy.io/models/ru#ru_core_news) (data obraschenija 10.07.2023).
13. Natasha / slovnet. URL: <https://github.com/natasha/slovnet> (data obraschenija 10.07.2023).
14. Mozharova V. A., Loukachevitch N. V. Combining knowledge and CRF-based approach to named entity recognition in Russian // 5<sup>th</sup> Intern. Conf. on Analysis of Images, Social Networks and Texts, AIST 2016. Yekaterinburg, Russia, 2016: Springer International Publishing, 2017. S. 185–195.
15. Sistema PullEnti – izvlechenie informacii iz tekstov estestvennogo jazyka i avtomatizirovannee postroenie informacionnyh sistem / O. V. Zolotar'ov, M. M. Sharnin, S. V. Klimenko, K. I. Kuznecov // Tr. Mezhdunar. nauch. konf.: v 2 t. «Situacionnye centry i informacionno-analiticheskie sistemy klassa 4I dlja zadach monitoringa i bezopasnosti» (SCVRT2015–16). T. 2. Protvino: ANO «Institut fiziko-tehnicheskoy informatiki», 2016. S. 28–35. (In Russ.).

---

### Information about the authors

**Leonid A. Kutsenok** – postgraduate student of the Department of Automation and Control Processes of Saint Petersburg Electrotechnical University.

E-mail: leo@sochi.com

**Yuri A. Korablev** – Cand. Sci. (Eng.), Associate Professor of the Department of Automation and Control Processes of Saint Petersburg Electrotechnical University.

E-mail: juri.korablev@gmail.com

<https://orcid.org/0000-0002-2432-7169>

Статья поступила в редакцию 06.06.2023; принята к публикации после рецензирования 09.11.2023; опубликована онлайн 30.01.2024.

Submitted 06.06.2023; accepted 09.11.2023; published online 30.01.2024.

---