Реализация метода распознавания именованных сущностей в сфере информационных технологий для анализа задач по разработке программного обеспечения

Л. А. Куценок[1], Ю. А. Кораблев[2]
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)
[1]kutsenokla@gmail.com, [2]juri.korablev@gmail.com

Аннотация. **В данной статье рассматривается реализация 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, которые могут быть использованы для извлечения информации об именованных сущностях из текстовых данных на русском языке. Эта статья предлагает подходы к реализации анализа задач разработки, что способствует улучшению планирования, управления проектами и сокращению времени разработки программного обеспечения.**

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

I. Введение

В современном мире разработка программного обеспечения играет ключевую роль во многих отраслях промышленности и общественной жизни. От небольших стартапов до крупных корпораций, все организации сталкиваются с необходимостью разрабатывать программные продукты для оптимизации своих бизнес-процессов, улучшения взаимодействия с клиентами и предоставления новых продуктов и услуг на рынке. Однако разработка программного обеспечения – это сложный и трудоемкий процесс, который включает в себя множество задач, начиная от анализа требований и планирования проекта, и заканчивая тестированием и внедрением готового продукта.

Интенсивность и динамика процесса разработки программного обеспечения приводит к появлению большого объема текстовых данных, которые описывают задачи, запросы на изменения, обнаруженные ошибки, комментарии и другие аспекты разработки ПО. Эти данные содержат важную информацию о проекте, но , ввиду большого объема данных, их ручной анализ может быть трудоемким и времязатратным процессом. В связи с этим возникает потребность в автоматизированных методах анализа данных, которые позволят извлекать информацию из текста и предоставлять сведения для принятия управленческих решений.

Целью данной работы является разработка и применение модели распознавания именованных сущностей (Named Entity Recognition, сокр. NER) для анализа текстовых данных, связанных с разработкой программного обеспечения. В предыдущих работах была рассмотрена разработка сервиса для анализа трудозатрат сотрудников IT-компаний [1], однако полученных из систем управления проектами данных оказалось недостаточно для атрибуции задач и их точного анализа.

Для реализации поставленной цели исследования требуется решить следующие задачи:

- сбор и предварительная обработка данных: необходимо провести подготовку текстовых данных из различных источников, связанных с разработкой программного обеспечения, и проведем предварительную обработку данных, чтобы подготовить их для обучения модели NER;
- выбор и адаптация модели: необходимо провести анализ существующих методов и подходов к NER для выбора подходящей модели и ее адаптации;
- обучение и оценка модели: необходимо провести обучение выбранной модели на подготовленных данных и провести оценку ее производительности с помощью соответствующих метрик.

II. Реализация модели

A. *Сбор и предварительная подготовка данных*

Ввиду отсутствия на момент проведения исследования соответствующего набора данных на русском языке, в качестве набора данных для разработки модели был выбран датасет JOSSE (The JIRA Open-Source Software Effort) [2]. Данный набор данных был создан на основе общедоступной информации из экземпляров системы управления проектами Atlassian Jira на английском языке компаний Apache, JBoss, и Spring за 2019 и 2020 годы. Датасет предоставляет для анализа более 21 тысяч сущностей задач по 40 полям, среди которых:

- идентификаторы: 6 (такие как Issue key, Project id и другие);
- текстовые поля: 7 (такие как. Summary, Description, Comments и другие);
- числовые поля: 10 (такие как Votes, Time Spent, Original Estimate и другие);

- поля даты и времени: 4 (такие как Created, Due date, Updated и другие);
- перечисления: 13 (такие как Project type, Issue type, Component и другие).

В рамках данного исследования рассматриваются только текстовые данные, следовательно, потенциально подходящих полей в наборе данных JOSSE всего 7. Рассмотрим их подробнее в табл. 1.

ТАБЛИЦА I. ТИПЫ ТЕКСТОВЫХ ДАННЫХ ДАТАСЕТА JOSSE

№	Поле	Описание	Есть в % сущностей	Подходит для анализа
1	Summary	Тема задачи	100%	Да
2	Project name	Название проекта, к которому привязана задача	100%	Нет, содержит только название проекта
3	Project URL	Ссылка на проект в системе управления проектами	95%	Нет, является URL-ссылкой в сети Интернет
4	Description	Описание задачи	80%	Да
5	Project description	Описание проекта, к которому привязана задача	79%	Нет, описывает проект задачи, а не саму задачу
6	Comment	Набор комментарие в к задаче	73%	Нет, так как содержит данные о ходе работы над задачей, а не описание самой задачи
7	Environment	Описание среды, в которой выполнялась задача	6%	Нет, описывает среду выполнения задачи, а не саму задачу

По результатам анализа типов текстовых данных датасета JOSSE можно сделать вывод о том, что наиболее подходящим полем для анализа в рамках данной работе являются поле Summary и Description, которые содержат краткое и полное описание задачи соответственно.

Для подготовки набора данных для обучения модели NER был произведена случайная выборка из 1 тысячи задач, содержащих поля Summary и Description из исходного датасета для ручной разметки, поскольку датасет JOSSE не содержит предварительно размеченных меток для именованных сущностей.

С учетом ограниченных человеческих ресурсов на подготовку данных, аннотация полей Summary и Description проводилась по следующему наберу NER-тегов:

- TYPE – тип задачи: разработка, проектирование, внедрение, развертывание и другие;
- OBJ – объект задачи: страница, экран, функция, метод, инструмент и другие;
- DESC – описание объекта задачи: (функция) проверки, (страница) аутентификации пользователей, (метод экспорта) списка пользователей и другие.

Аннотация исходных данных для подготовки модели к обучения проводилась в среде WebAnno, выбранная на основе результатов сравнения инструментов для создания аннотаций, пригодных для применения в NER [3]. Пример результата подготовки аннотаций представлен на рис. 1.

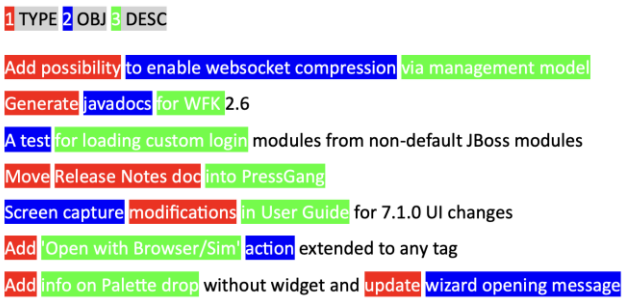

Рис. 1. Пример результата подготовки аннотации для NER-модели

По результатам аннотирования выбранного набора задач был получен датасет, подготовленный к обучению распознаванию именованных сущностей, со следующими характеристиками:

ТАБЛИЦА II. ХАРАКТЕРИСТИКИ ДАТАСЕТА

№	Сущность	Количество экземпляров	Доля в общем объеме датасета
1	TYPE	2315	25%
2	OBJ	2976	32%
3	DESC	3848	42%

Полученный датасет был разделен на тренировочную, проверочную и тестовую выборки в соотношение 80:10:10 соответственно [4]. Схематическое представление процесса обучение модели NER представлено на рис. 2.

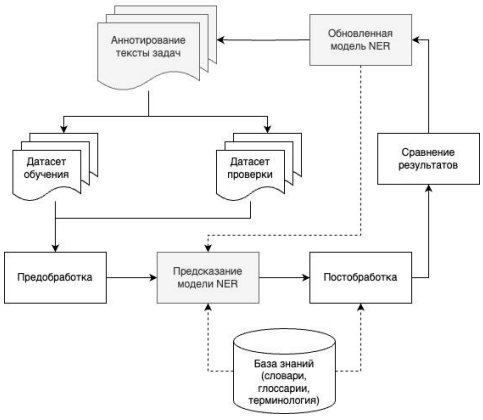

Рис. 2. Схематичесое представление процесса обучения NER-модели

B. Выбор и адаптация модели

Ранее проведенные исследования [5] показали, что одним из инструментов, пригодных для реализации анализа задач разработки программного обеспечение с применением технологии распознавания именованных сущностей является фреймворк DeepPavlov.

DeepPavlov [6] – это фреймворк для языка программирования Python, разработанный для работы с обработкой естественного языка (Natural Language Processing, сокр. NLP) и обучением моделей глубокого обучения (deep learning). Он предоставляет различные инструменты и предварительно обученные модели для

различных задач NLP, включая распознавание именованных сущностей.

Модель NER в DeepPavlov использует схему кодирования тегов BIO, где B – начало тега (Beginning), I – содержимое тега (Inside), O – другие, несвязанные теги (Outside). Для соответствия данной схеме сформированный ранее набор данных был автоматически обработан средствами языка программирования Python.

В качестве исходной модели для обучения на сформированном датасете (fine-tuning) была выбрана встроенная в DeepPavlov модель ner_conll2003_bert, основанная на наборе данных CoNLL-2003 [7] и использующая языковую модель BERT (Bidirectional Encoder Representations from Transformers), показывающая наилучшие характеристики для распознавания именованных сущностей в английском языке. Обучение модели NER было проведено на обучающем наборе данных, а проверочная выборка использовалась для настройки гиперпараметров и оценки производительности модели. Процедура обучения заняла приблизительно 1800 эпох, на каждой из которых модель будет обновлялась с помощью оптимизатора и функции потерь, в качестве которых использовались стандартные функции Torch Trainer библиотеки PyTorch [8].

III. Анализ результатов

Оценка результатов работы разработанной модели была проведена на тестовом наборе данных, который ранее не использовался в процессе обучения и настройки модели. Для оценки использовать стандартные метрики оценки, такие как точность (precision), полнота (recall) и F1-мера [9], основанных на результатах работы модели по истинно положительный, ложноположительный, истинно ложным и ложноотрицательным предсказаниям, чтобы оценить качество распознавания именованных сущностей и точность работы модели в целом.

В табл. 3–5 представлены результаты анализа работы модели на тестовом наборе данных для полей Summary и Description соответственно, а также для всей модели в целом. Метрики точности, полноты и F1-меры были рассчитаны для каждого класса (NER-тега) отдельно, метрики для полного текста и для всей модели в целом были рассчитаны как средневзвешенное значение каждого из классов. Расчет метрик производился с помощью библиотеки Scikit-learn [10] языка программирования Python.

ТАБЛИЦА III. Результаты работы модели для поля Summary

№	Сущность	Precision	Recall	F1
1	**Полный текст**	**0.75**	**0.75**	**0,74**
2	TYPE	0.85	0.70	0.76
3	OBJ	0.70	0.78	0.73
4	DESC	0.73	0.76	0.74

ТАБЛИЦА IV. Результаты работы модели для поля Description

№	Сущность	Precision (%)	Recall (%)	F1 (%)
1	**Полный текст**	**0.68**	**0.69**	**0.68**
2	TYPE	0.76	0.65	0.70
3	OBJ	0.67	0.73	0.69
4	DESC	0.65	0.68	0.66

ТАБЛИЦА V. Результаты работы модели в целом

№	Сущность	Precision (%)	Recall (%)	F1 (%)
1	**Полный текст**	**0.71**	**0.71**	**0.70**
2	TYPE	0.80	0.67	0.72
3	OBJ	0.68	0.75	0.70
4	DESC	0.68	0.71	0.68

На основе представленных результатов можно сделать вывод о том, что разработанная модель работает, показывает среднюю точность 74 % при анализе темы задачи и среднюю точность 68 % при анализе полного описания. Данный результат можно обосновать тем, что поле Description, как правило, содержит в себе больше нерелевантных к рассматриваемой задаче данных. Наличие большего числа O-тегов по результатам разметки приводит к меньшей точности работы модели, следовательно, для повышения характеристик ее работы требуется дополнительные трудозатраты на более подробную аннотацию исходного набора данных. Кроме того, сравнительно небольшое количество выбранных для аннотации тегов по сравнению с представленными в модели также оказало негативный результат, что необходимо учесть в будущих исследованиях и доработке созданной модели.

Несмотря на это, достигнутая по результатам исследования общая средняя точность работы модели в 70.46 % при анализе полей задачи Summary и Description делает возможным ее интеграцию в инструменты анализа трудозатрат сотрудников IT-компаний для создания более полного представления о распределении ресурсов сотрудников на основе данных из систем управления проектами.

IV. Заключение

В данной статье была представлена разработка и оценка модели распознавания именованных сущностей (NER) для анализа задач разработки программного обеспечения с использованием фреймворка DeepPavlov. Исследование имело цель создать модель с общей точностью 70.46 %, способную распознавать разнообразные типы именованных сущностей в текстовых данных, связанных с разработкой ПО.

Результаты исследования показали, что разработанная модель NER с использованием DeepPavlov обладает достаточной производительностью и точностью в распознавании именованных сущностей в текстовых данных разработки ПО для ее внедрения в инструменты по анализу трудозатрат. Разработанная модель позволяет автоматизировать часть процесса анализа задач разработки программного обеспечения, что может значительно повысить эффективность и точность этого процесса.

В заключение, исследование демонстрирует возможность применения фреймворка DeepPavlov для разработки моделей NER и его значимость в области анализа текстовых данных в разработке программного обеспечения. Разработанная модель может быть использована в различных задачах исследования и приложениях, где требуется точное распознавание именованных сущностей для более глубокого анализа и понимания текстовых данных. Дальнейшее развитие и оптимизация модели могут открыть новые возможности и применения в будущих исследованиях и проектах разработки программного обеспечения.

Список литературы

[1] Leonid A. Kutsenok, Natalya V. Razmochaeva, Viktor P. Semenov, Artem A. Bezrukov. The Problem of Man-hour Distribution in Modern Task Managers // 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). – IEEE, 2020.

[2] Alhamed Mohammed, Storer Tim. JOSSE: A Software Development Effort Dataset Annotated with Expert Estimates (1.0) [Data set]. 38th IEEE International Conference on Software Maintenance and Evolution (ICSME 2022).

[3] Neves M., Ševa J. An extensive review of tools for manual annotation of documents // Briefings in bioinformatics. 2021. Т. 22. №. 1.

[4] Joseph V. R. Optimal ratio for data splitting // Statistical Analysis and Data Mining: The ASA Data Science Journal. 2022. Т. 15. №. 4.

[5] Куценок Л.А., Кораблев Ю.А. Применение метода распознавания именованных сущностей в сфере информационных технологий для анализа задач по разработке программного обеспечения // Известия СПбГЭТУ «ЛЭТИ». 2023. Т. 16, № 10. (в печати)

[6] Mikhail Burtsev, Alexander Seliverstov, Rafael Airapetyan, Mikhail Arkhipov, Dilyara Baymurzina, Nickolay Bushkov, Olga Gureenkova, Taras Khakhulin, Yuri Kuratov, Denis Kuznetsov, Alexey Litinsky, Varvara Logacheva, Alexey Lymar, Valentin Malykh, Maxim Petrov, Vadim Polulyakh, Leonid Pugachev, Alexey Sorokin, Maria Vikhreva, Marat Zaynutdinov. Deeppavlov: Open-source library for dialogue systems // Proceedings of ACL 2018, System Demonstrations. 2018.

[7] Sang E.F., De Meulder F. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition // arXiv preprint cs/0306050. 2003.

[8] Collobert R., Bengio S., Mariéthoz J. Torch: a modular machine learning software library. – Idiap, 2002. – №. REP_WORK.

[9] Taha A.A., Hanbury A. Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool //BMC medical imaging. 2015. Т. 15. №. 1.

[10] Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel. Scikit-learn: Machine learning in Python // the Journal of machine Learning research. 2011. Т. 12.