Разработка системы распознавания намерений и сущностей для систем вопросов и ответов с использованием “no code” платформы "TWIN"

К.А. Аксенов[1], *Л. Сунь*[1], *И.А. Калинин*[2]

[1]*Уральский федеральный университет имени первого Президента России Б.Н. Ельцина*
[2]*ООО «Уралинновация»*

Аннотация: В данной статье разработана новая модель распознавания намерений и сущностей для предметной области обслуживания авиапассажиров, обозначенная как IRERAIR-TWIN, с использованием платформы разработки «без кода» вопросно-ответных систем “TWIN”. Преимущества платформы «без кода» были проанализированы с точки зрения удобства разработки прикладной вопросно-ответной системы и уменьшения объема работы по разработке прикладной модели для узкой предметной области. Результаты показывают, что система “TWIN” предоставляет интуитивно понятный пользовательский веб-интерфейс и более простой подход к разработке семантического модуля системы вопросно-ответной системы, способной решать прикладные задачи для узкой предметной области, не являющиеся чрезмерно сложными. Однако данный подход имеет ограничения для задач глубокого семантического анализа, особенно в сложном контекстном выводе и обработке больших текстовых фрагментов. В заключение статьи подчеркивается, что будущие исследования будут направлены на использование платформ “низкого (малого) кода” (англ. low code) на основе ChatGPT и больших языковых моделей для дальнейшего улучшения интеллектуальных возможностей модели IRERAIR-TWIN. Это расширение направлено на расширение сферы применения сценариев.

Ключевые слова: вопросно-ответные системы, без кода, низкокодовый, распознавание намерений, распознавание именованных сущностей, аннотирование текста, инженерия признаков, предобученная модель, разработка ПО, конечный пользователь.

Введение в разработку моделей без кода

Традиционно разработка интеллектуальных моделей понимания естественного языка требует от разработчиков создания сложных архитектур вопросно-ответных систем и выполнения большого объема работы. Однако с появлением больших языковых моделей разработчикам больше не требуется проектировать и реализовывать сложные алгоритмы обработки естественного языка с нуля, что значительно сокращает объём работы, затрачиваемой на инженерию признаков предметной области. Некоторые исследовательские сообщества упростили использование этих моделей путём создания интерфейсов, позволяющих пользователям напрямую вызывать

большие языковые модели через такие фреймворки, как PyTorch и TensorFlow, для построения моделей для прикладных задач. Эти платформы служат площадками для совместного использования ресурсов разработчиков, включая платформы разработки, такие как "Hugging Face Developer Platform" и "PaddleNLP" Development Platform. Однако проекты, основанные на таких платформах разработки, по-прежнему требуют помощи профессионалов в области анализа данных и разработки компьютерного программного обеспечения.

Из литературы по тонкой настройке моделей [1] видно, что непрофессионалы не могут быстро освоить такие техники, как “тонкая настройка модели для конкретной задачи” (обработка символов, последовательностей, выделение признаков, разметка коротких и длинных текстов), “настройка параметров обучения” (прямое или косвенное задание гиперпараметров), “методы оценки и тестирования моделей”, а также оценить и подготовить (запрограммировать) объем кода, необходимый для качественной работы прикладной системы.

На предыдущем этапе работ [2] мы построили модели распознавания намерений и именованных сущностей на основе структуры двунаправленного энкодера архитектуры Transformer для улучшения интеллектуальных возможностей систем вопросно-ответного типа. При обучении моделей для конкретных задач, использующих такие масштабные языковые модели, возникают не только упомянутые препятствия для непрофессионалов, но и необходимость в использовании дорогостоящих GPU для параллельного ускорения процесса обучения. Кроме того, перед разработкой модели необходимо собрать значительное количество данных для эффективной разметки исходного массива данных (текстов и текстов диалогов) и последующего получения релевантных результатов. Выбор меток и определение стратегии их разметки также требует

профессионального подхода и выбора с целью снижения рисков построения прикладной модели [3]. В большинстве проектов в области обработки естественного языка, с целью повышения точности классификации текста с использованием крупных языковых моделей, инженеры также занимаются морфологическим анализом [4]. Для предварительной обработки данных они могут использовать специализированные инструментарии, такие как spaCy, Jieba, PyMorphy2 и PyMystem3. Кроме того, для решения специфических задач инженерам необходимо разработать шаблоны проектирования функций на основе правил или даже создать базы знаний, чтобы улучшить понимание и точность модели в отношении терминологии и знаний о предметной области. Эти процессы могут представлять трудности для неспециалистов в области ИТ с точки зрения экономических затрат и несвойственных операций.

Для снижения сложности разработки моделей в последнее время в общественное поле зрения вышли платформы разработки с низким и нулевым кодом (англ. low code / no code). Разработка с низким и нулевым кодом относится к категории разработки конечным пользователем (англ. End-User-Developmen (EUD)) [5]. Она может снизить порог входа в разработку программного обеспечения для непрограммирующих пользователей (предметных специалистов) и повысить эффективность разработки (за счет того, что в разработке участвует предметный специалист, хорошо знакомый со спецификой предметной области). Например, платформы «без кода» позволяют разрабатывать чат-боты [6], классифицировать нейронные изображения [7] и распознавать лекарства [8]. В [9] отмечается, что использование платформ «без кода» на практике может повысить скорость разработки по сравнению с классической разработкой в пять раз. В настоящее время к распространенным платформам «без кода» относятся следующие: “FlutterFlow”, “Google App Maker”, “AppSheet”,

“Bubble”; а также платформы разработки мини-программ “Baidu”, “Zapier”, “Appy Pie” и “TWIN”. Эти платформы предлагают удобные визуальные интерфейсы, избавляя пользователей от непосредственного контакта с обширными профессиональными машинными языками. Они снизили сложность разработки высокоспециализированных моделей принятия решений для предметных областей и прикладных задач. В проекте, представленном в этой статье, в качестве инструмента разработки использовалась платформа «без кода» разработки вопросно-ответных систем TWIN, специализирующаяся на обработке естественного языка, чтобы продемонстрировать специфику процесса разработки без кода.

Применение вопросно-ответной системы TWIN для диагностики сущностей и намерений в услугах поддержки клиентов авиалиний

Для смягчения рисков моделирования и оптимизации рабочей нагрузки по распознаванию намерений и сущностей мы использовали интегрированную систему TWIN для создания модуля распознавания сущностей и намерений в вопросно-ответной системе(BOC).

Система TWIN представляет собой интегрированную модель, охватывающую инженерию признаков данных (выделение признаков, разметку исходных данных с привязкой к признакам), обучение модели и тестирование модели [10]. В этой системе пользователи могут интерактивно добавлять метки намерений и сущностей через визуальный веб-интерфейс. В процессе структурирования и формализации предметной области пользователи могут гибко добавлять или удалять категории сущностей и намерений по своему усмотрению для удовлетворения потребностей проекта. После завершения разметки обучающей выборки и выделения признаков данных (привязки сущностей и намерений к конкретным экземплярам признаков) пользователь-разработчик может начать обучение модели. Затем он может протестировать модель через веб-интерфейс тестирования,

проверяя общую точность классификации и точность распознавания конкретных намерений и сущностей. Пользователь-разработчик может использовать результаты тестирования для оперативной корректировки модели и для предотвращения проблем, таких как переобучение и отклонение параметров модели, внося в нее корректировки. Преимущество такого подхода, реализованного в ВОС «TWIN» заключается в том, что итеративно разрабатывая модель ее можно обучать и тестировать с первых шагов, начиная с небольшого количества образцов: добавляя обучающую выборку и размечая текст в каждой или в отдельной категории намерений, привязывая признаки к конкретным сущностям, можно сразу проводить обучение и тестирование для проверки соответствия добавленных и размеченных данных, согласно классификации намерений, а в случае диагностики ошибок – сразу их устранять.

А. Создание метки намерений

Для построения модели распознавания намерений и сущностей для поддержки ВОС при обслуживании клиентов авиакомпаний, называемой (англ. Intent recognition and entity recognition in air services (IRERAIR)), нами была разработана на основе модели TWIN модель IRERAIR-TWIN. В этом проекте использовали 8 меток намерений (рисунок 1) и 21 метка сущностей.

Намерение	Описание на русском языке
atis_flight	Спрашивайте о рейсах
atis_airfare	Узнать цену
atis_airline	Запрос авиакомпании
atis_ground_service	Спросите, какой наземный транспорт доступен
atis_abbreviation	Спросите о значении аббревиатур
atis_aircraft	Спросите о типе самолета
atis_quantity	Спросите о количестве рейсов и спектре услуг
atis_flight_time	Узнайте расписание рейсов

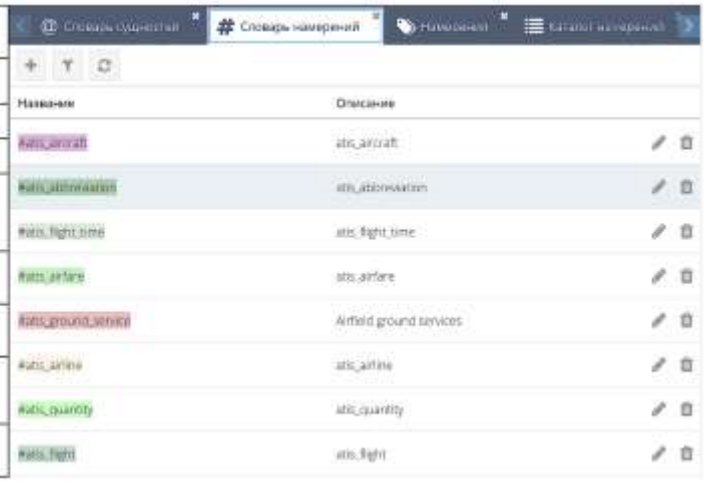

Рис. 1. Метки намерений для модели IRERAIR-TWIN.

B. Создание этикеток сущностей и проверка точности этикеток в режиме реального времени

Процесс разработки агента анализа текста в ВОС “TWIN” соблюдал следующую последовательность: *"Выберите метку намерения → Добавьте данные → Аннотируйте (разметьте) данные → Обучите модель → Проверьте модель → Проверьте соответствие категорий именованных сущностей → Добавьте или уберите необходимые типы именованных сущностей → Добавьте или уберите данные по необходимости → Переобучите → Повторно протестируйте модель...",* при этом этот цикл повторяется до достижения удовлетворительных результатов тестирования. После этого тот же процесс повторяется для следующей метки намерения, чтобы продолжать размечать исходный текст. На рисунке 2 показан детальный процесс разметки текста и привязки слов/словосочетаний к сущностям, а на рисунке 3 представлен список из 21 категории именованных сущностей, использованных при обучении модели IRERAIR-TWIN.

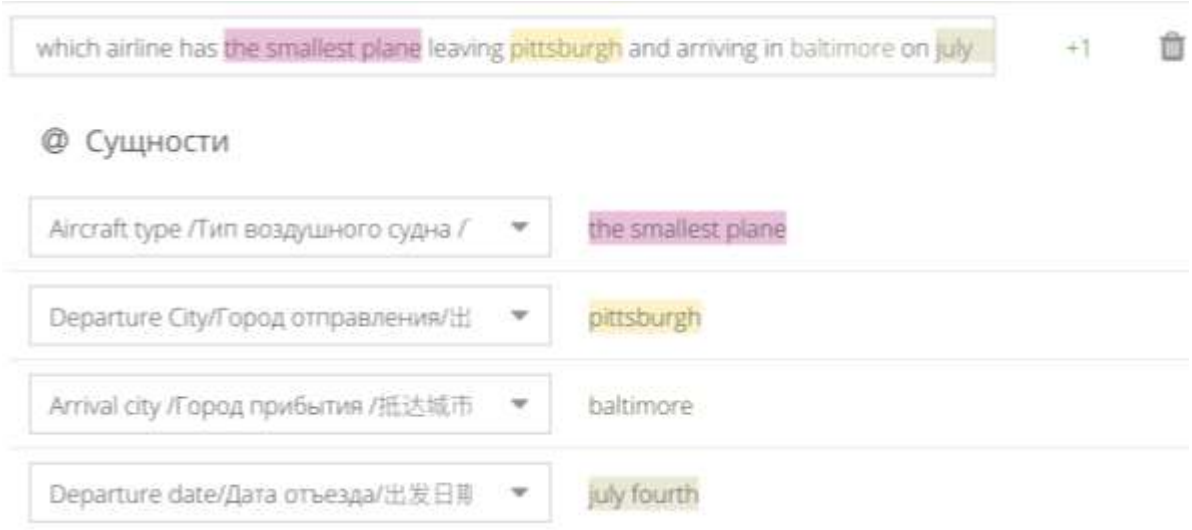

Рис. 2. Разметка текста и привязка слов/словосочетаний к именованным сущностям в ВОС «TWIN».

Этикетки сущностей
@Abbreviated
@AircrattPositionRating
@Aircrafttype
@AirlineName
@AirportBusstops
@AirSchedule
@ArrivalCity
@ArrivalDate
@Arrivaltime
@CitiesCovered
@Cityoftransit
@DepartureAirport
@DepartureCity
@DepartureDate
@Departuretime
@Fares
@Flightnumber
@GroundTransport
@Ticket
@Times
@TypeofAirTickets

Рис. 3. 21 именованная сущность модели IRERAIR-TWIN.

C. Составная метка намерения

Как правило, когда пользователи задают вопросы, они могут одновременно передавать несколько запросов намерений. Для удовлетворения потребности в обнаружении нескольких намерений, система Twin поддерживает добавление составных (мульти-) намерений для более точной классификаций. В процессе разметки также использовали составные метки намерений, чтобы обеспечить возможность BOC более точно идентифицировать и обрабатывать составные намерения, примеры составных намерений показаны на рисунке 4.

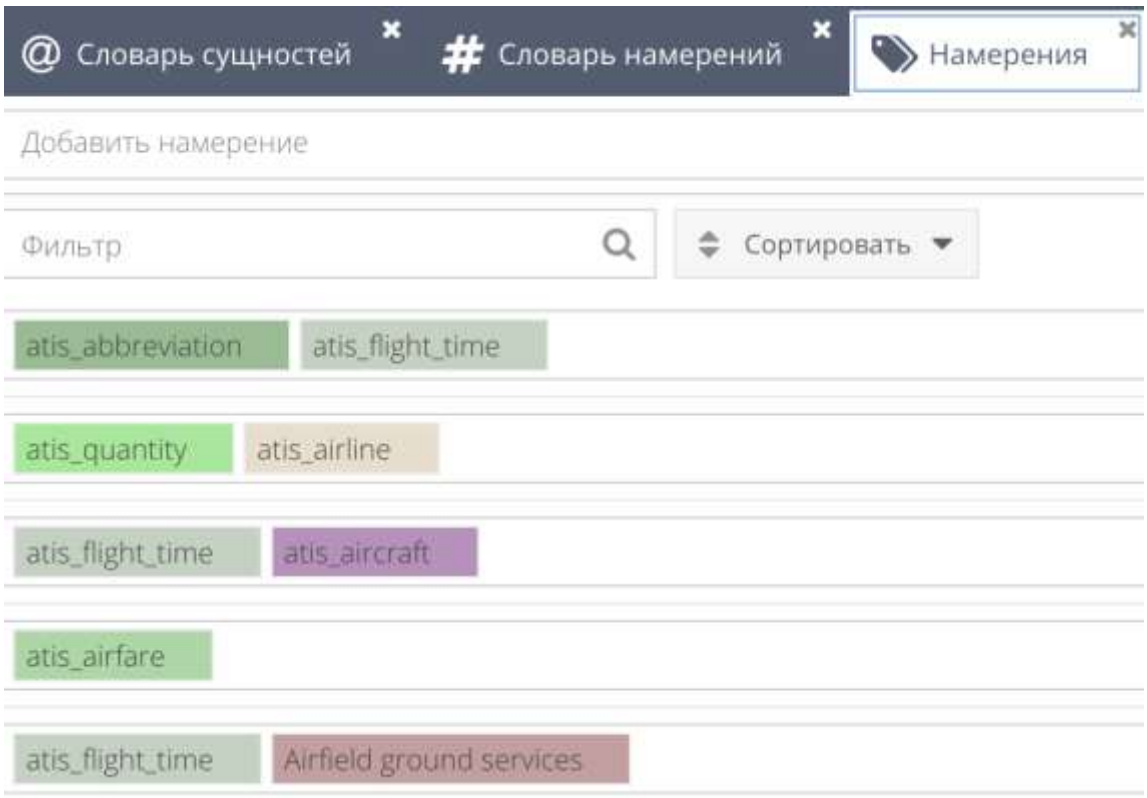

Рис. 4. Описание составных намерений IRERAIR-TWIN

D. Обучение и отладка модели диагностика сущностей и намерений

Для получения релевантного результата диагностики сущностей и намерений в данном проекте был проведен ряд циклов доработки и дообучения модели. После каждого цикла доработки-обучения выполняли тесты, добавляя различные текстовые данные с похожими значениями в рамках одного составного намерения. Параллельно экспериментировали с добавлением фраз и ключевых терминов, а затем оценивали влияние этих фраз на диагностику намерений. На рисунке 5 – показан веб-интерфейсы процесса обучения и результаты модели после обучения на тестах как для атомарных, так и составных намерений.

Из рисунка 10 видно, что Модель А способна точно определять намерения пользователя и успешно извлекать именованные сущности.

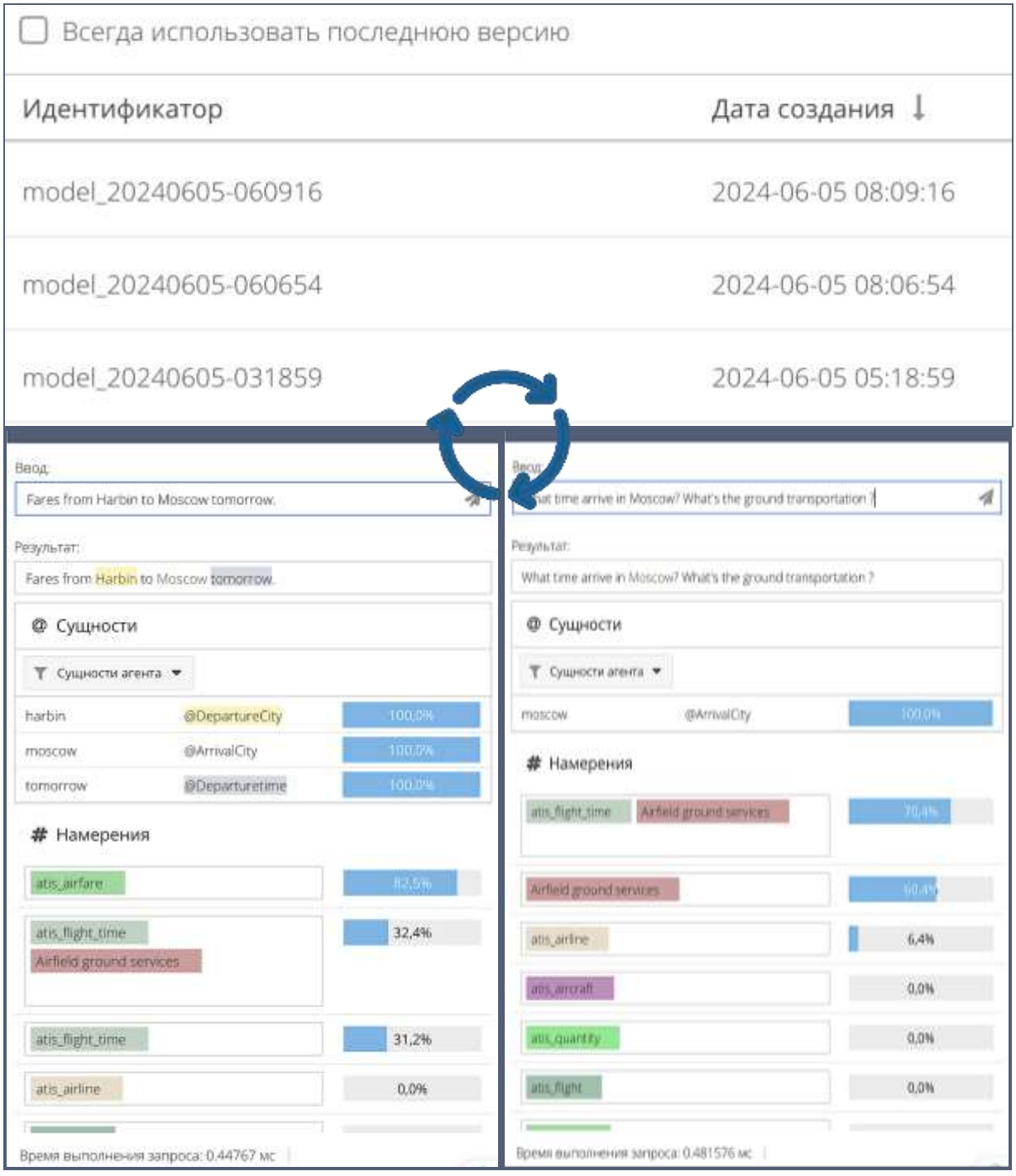

Рис. 5. Результаты обучения IRERAIR-TWIN, результаты диагностики намерений и именованных сущностей.

Заключение

Разработана новая модель диагностики намерений и именованных сущностей IRERAIR-TWIN для предметной области обслуживания

авиапассажиров на основе интерактивной и визуальной платформы «без кода» “TWIN”. В сравнении с традиционными методами разработки прикладных моделей ВОС на основе “ручного” программирования показывает, что применение платформ «без кода» для прикладных задач существенно снижает требования к квалификации пользователя-разработчика и существенно снижает объем работ. Визуальный интерфейс разработки удобен для предметных специалистов (непрограммирующих пользователей), предлагая интуитивно понятный веб-интерфейс.

Более того, технология «без кода», обладая способностью инкапсулировать код в модульных конструкциях, предоставляет возможности для создания модели многократного агентного взаимодействия через многомодульность и распределенность вычислений (например, можно разработать несколько разных программных агентов (роботов обзвона или чат-ботов), каждый из которых решает узкую задача, но в случае необходимости может перенаправить / переключить общение с пользователем-клиентов на другого агента). Таким образом, данная парадигма не только упрощает доступ к сложным функциональным возможностям, но и предоставляет профессионалам инструменты для эффективного управления сложными задачами взаимодействия распределенных систем, тем самым усиливая их способности к решению задач в различных предметных областях.

В ходе разработки ВОС было установлено, что модель распознавания намерений, разработанная с использованием технологии «без кода», предоставленной компанией “TWIN”, также имеет свои ограничения. Например, при тестировании с использованием длинных текстов было выявлено, что исключительно применение техник «без кода» не удовлетворяет требованиям контекстного семантического понимания. Чтобы решить эту проблему, следующий этап нашей работы стратегически

планируется таким образом, чтобы интегрировать технологию «без кода», с возможностями «низкий код» на основе ChatGPT, направленную на увеличение возможностей модели «IRERAIR-TWIN» для анализа и понимания длинных текстов.

Литература

1. Zhu Z., Mao K. Knowledge-based BERT word embedding fine-tuning for emotion recognition. Neurocomputing, 2023, vol.552, No.126488.

2. Аксенов К.А., Сунь Л. Эволюция и современное состояние систем ответов на вопросы: технологии распознавания намерений и именованных сущностей с использованием модели BERT. Инженерный вестник Дона. 2024. No.7. URL: ivdon.ru/ru/magazine/archive/n7y2024/9371

3. Alshammari N., Alanazi S. The impact of using different annotation schemes on named entity recognition. Egyptian Informatics Journal. 2021, No.22(3). Pp.295-302.

4. Ерискина Е.В., Курушин Д.С, Яруллин Д.В., Корюкин Ю. Д. Объектно-ориентированная модель морфологического анализатора русскоязычного текста. Инженерный вестник Дона. 2018. No.4. URL: ivdon.ru/ru/magazine/archive/n4y2018/5528

5. Li L.Y., Wu Z.W. How can No/Low code platforms help end-users develop ml applications?-a systematic review // International Conference on Human-Computer Interaction. Cham: Springer Nature Switzerland. 2022. Pp.338-356.

6. Luo C.J., Gonda D.E. Code Free Bot: An easy way to jumpstart your chatbot! // IEEE International Conference on Engineering, Technology and Education. 2019. Pp.1-3.

7. Jungo P., Hewer E. Code-free machine learning for classification of central nervous system histopathology images. Journal of Neuropathology & Experimental Neurology, 2023, No.82(3). Pp.221-230.

8. Ashraf A.R., Somogyi V.A., Merczel S., Gyimesi N. Fittler A. Leveraging code-free deep learning for pill recognition in clinical settings: A multicenter, real-world study of performance across multiple platforms. Artificial Intelligence in Medicine. 2024. vol.150. No.102844.

9. Liu D., Jiang H., Guo S., Chen Y., Qiao L. What's Wrong With Low-Code Development Platforms? An Empirical Study of Low-Code Development Platform Bugs. IEEE Transactions on Reliability, 2023. Vol.73(1). Pp.695-709.

10. TWIN User Support Documentation URL: confluence.twin24.ai/pages/viewpage.action?pageId=82448622.

References

1. Zhu Z., Mao K. Knowledge-based BERT word embedding fine-tuning for emotion recognition. Neurocomputing, 2023, 2023, vol.552, No.126488.

2. Aksenov K.A., Sun L. Evoliutsiia i sovremennoe sostoianie sistem otvetov na voprosy: tekhnologii raspoznavaniia namerenii i imenovannykh sushchnostei s ispolzovaniem modeli BERT, Inzhenernyj vestnik Dona. 2024. No.7. URL: ivdon.ru/ru/magazine/archive/n7y2024/9371.

3. Alshammari N., Alanazi S. The impact of using different annotation schemes on named entity recognition. Egyptian Informatics Journal. 2021, No.22(3). Pp.295-302.

4. Eriskina E.V., Kurushin D.S, Iarullin D.V., Koriukin Iu. D. Obieektno-orientirovannaia model morfologicheskogo analizatora russkoiazychnogo teksta, Inzhenernyj vestnik Dona. 2018. No.4. URL:ivdon.ru/ru/magazine/archive/n4y2018/5528.

5. Li L.Y., Wu Z.W. How can No/Low code platforms help end-users develop ml applications?-a systematic review // International Conference on Human-Computer Interaction. Cham: Springer Nature Switzerland, 2022. Pp.338-356.

6. Luo C J, Gonda D E. Luo C.J., Gonda D.E. Code Free Bot: An easy way to jumpstart your chatbot! // IEEE International Conference on Engineering, Technology and Education. 2019. Pp.1-3.

7. Jungo P, Hewer E. Hewer E. Code-free machine learning for classification of central nervous system histopathology images, Journal of Neuropathology & Experimental Neurology. 2023. Vol.82(3). Pp.221-230.

8. Ashraf A.R., Somogyi V.A., Merczel S. Gyimesi N., Fittler A. Leveraging code-free deep learning for pill recognition in clinical settings: A multicenter, real-world study of performance across multiple platforms, Artificial Intelligence in Medicine. 2024. Vol.150. No.102844.

9. Liu D., Jiang H., Guo S., Chen Y., Qiao L. What's Wrong With Low-Code Development Platforms? An Empirical Study of Low-Code Development Platform Bugs . IEEE Transactions on Reliability, 2023. Vol.73(1). Pp.695-709.

10. TWIN User Support Documentation URL: confluence.twin24.ai/pages/viewpage.action?pageId=82448622.

Дата поступления: 22.08.2024

Дата публикации: 25.09.2024