УДК 004

АВТОМАТИЧЕСКИЕ И ПОЛУАВТОМАТИЧЕСКИЕ МЕТОДЫ ПОСТРОЕНИЯ ГРАФА ЗНАНИЙ ПРЕДМЕТНОЙ ОБЛАСТИ И РАСШИРЕНИЯ ОНТОЛОГИИ

А. П. Халов[1] [0009-0005-4584-8245], О. М. Атаева[2] [0000-0003-0367-5575]

[1]Московский физико-технический институт, г. Долгопрудный, Московская обл., Россия

[1, 2]Федеральный исследовательский центр «Информатика и управление» Российской академии наук, г. Москва, Россия

[1]khalov.a@phystech.edu, [2]oataeva@frccsc.ru

Аннотация

Рассмотрен цикл построения графа знаний и расширения онтологии для специальной предметной области, описывающей процесс управления потоками данных в службах информационной поддержки. Предложена методика формирования корпуса данных для наполнения онтологии с автоматической псевдоразметкой, включающей специальные категории для фиксации ранее не представленных классов и отношений. Обучена специализированная модель извлечения именованных сущностей на корпусе данных объемом 3 млн токенов с 92 метками. Результаты были использованы для интеграции извлеченных фактов, что увеличило граф знаний до 0.98 млн триплетов, при этом коэффициент расширения графа (отношение общего числа фактов к явным триплетам) увеличился с 2.65 до 3.52 при сохранении логической согласованности. Наборы токенов с одинаковыми метками были преобразованы в устойчивые семантические множества, что позволило полуавтоматически расширить онтологию. В онтологию добавлены 12 новых классов, которые были извлечены из неструктурированных текстовых данных. Показан прикладной пример запросов и дальнейшей аналитики.

***Ключевые слова**: онтология, DOLCE, граф знаний, NER, BIO-разметка, RDF/OWL, SPARQL.*

ВВЕДЕНИЕ

Онтологии и графы знаний стали классическими инструментами для интеграции разнородных данных и поддержки принятия решений в бизнес-процессах управления (IT-сервисы) потоками данных (IT-домен), (IT Service Management, ITSM). Ранее [1] была сформирована объединенная онтология IT-домена путем расширения онтологии верхнего уровня DOLCE [2] доменно-специфичной онтологией ITSMO [3], описывающей концепции библиотеки ITIL. На практике ценность графа знаний определяется скоростью и точностью пополнения фактами из текстов (данных IT-домена, который управляется IT-сервисами) при одновременной способности адаптировать схему (ITSMO-онтологию) под изменяющийся тезаурус домена.

Для динамического построения и пополнения графов знаний, а также для расширения онтологии необходимо наличие специализированных моделей извлечения именованных сущностей и отношений (далее модели NER/RE), обученных на корпусах данных, объекты в которых размечены в соответствии с онтологией. Однако на момент настоящего исследования отсутствуют открытые наборы данных для ITSM, размеченные в соответствии с онтологией из работы [1] (далее онтология). При этом есть доступ к большим объемам текстовых данных в неразмеченном виде. Ручная или полуавтоматическая разметки для обучения модели NER/RE являются трудоемким и дорогостоящим процессом.

В настоящей работе исследован метод полностью автоматической разметки обучающего корпуса данных. Одним из известных способов разметки для задач NER/RE является формат BIO (Beginning-Inside-Outside), который содержит три типа меток: для обозначения первого токена сущности (B); любого последующего токена сущности (I); «фонового» токена (O). Еще одной задачей исследования является создание метода расширения онтологии, т. е. обнаружения в текстах сущностей и отношений, которые не находят соответствия в имеющейся онтологии, однако могут являться источником ценной информации для ее расширения.

В исследовании можно выделить три этапа:

1) формирование BIO-корпуса текстов с псевдоразметкой с помощью больших языковых моделей (Large Language Model, LLM), где мы вводим три

множества меток, а также специальные UNK-метки (от анг. UNKnown) в каждом из определенных множеств для разметки слов, не имеющих соответствий в текущей онтологии;

2) обучение модели-энкодера из семейства BERT для задачи извлечения именованных сущностей и отношений с проверкой корректности на уровне типов меток;
3) автоматическое пополнение графа знаний и полуавтоматическое расширение онтологии через экспертную валидацию кластеров векторных представлений (эмбедингов) слов, которые были размечены как UNK.

Исследуемая предметная область характеризуется высокой динамикой изменения терминов и большим объемом текстовых артефактов (например, заявки клиентов, инциденты, изменения конфигураций). Данные характеристики определяют потребность в методе, сочетающем экономичную и масштабируемую разметку данных с автоматическим обогащением графа и контролируемым расширением онтологии при сохранении логической согласованности.

Статья имеет следующую структуру: в разд. 1 представлены обзор предметной области и связанные работы; в разд. 2 описаны корпус данных и методики псевдоразметки; в разд. 3 представлена архитектура модели; в разд. 4 – конвейер SQL-RDF-NER-RDF; в разд. 5 описаны основные полученные результаты; в разд. 6 – полуавтоматическое расширение онтологии; заключение.

В приложении приведены текст инструкций для LLM (промпт) на этапе тестирования моделей-кандидатов (приложение 1), промпт для итоговой псевдоразметки целевого корпуса данных (приложение 2), результаты эксперимента с тестированием разных моделей на BIO-бенчмарках (приложение 3), распределение меток после применения метода псевдоразмеки (приложение 4), примеры устойчивых семантических множеств (далее синсеты, анг. synset, сокр. от synonym set) и их преобразований в объекты онтологии (приложение 5).

1. ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ И СВЯЗАННЫЕ РАБОТЫ

Онтология – это формальная спецификация концептуализации [4, 5]; в более прикладной трактовке: согласованное представление предметной области, разделяемое сообществом [6, 7]. В задачах извлечения информации и построения графов знаний онтология выступает схемой данных: сущности и отношения

между ними выражаются через классы и свойства с явными логическими ограничениями (аксиомами), что обеспечивает интерпретируемость, логический вывод и проверку согласованности результата.

В нашей работе использована объединенная онтология: верхний уровень задан известной онтологией DOLCE с абстрактными категориями (объект, процесс, событие), что гарантирует онтологическую стройность; прикладной уровень образует онтология ITSMO, охватывающая сущности и отношения известной методологии ITIL (например, Incident, Service, Change, SLA) и их взаимосвязи. Такое объединение дает единую семантически согласованную структуру, одновременно концептуально строгую и практически релевантную для автоматической аннотации текстов и обогащения графа. Роль онтологии в этом исследовании: (i) это каркас для проекции извлеченных фактов в RDF-триплеты; (ii) это живая схема, которая должна расширяться по мере появления новых терминов и связей в поступающих данных.

1.1. Онтологии и графы знаний для предметной области IT

Работы по онтологическому моделированию охватывают как верхнеуровневые онтологии (SUMO, BFO, DOLCE [6]), так и прикладные доменные онтологии, например ITSMO для процессов ITIL [8]. Слияние ITSMO с DOLCE следует рекомендованной стратегии наследования и позволяет избежать ручного проектирования больших фрагментов схемы [9]. Практики построения и эксплуатации корпоративных графов знаний систематизированы в руководствах [7, 10], где акцент сделан на роли онтологий и логического вывода для контроля качества данных и согласованности.

1.2. Совместное извлечение именованных сущностей и отношений

Автоматическое пополнение графов часто сводится к извлечению из текста сущностей и отношений. Классические каскады последовательного извлечения сущностей и отношений из текстов (NER+RE) постепенно уступают место унифицированным решениям.

Ранним сквозным подходом стала архитектура, предложенная в работе [12]: две параллельные нейросети на основе общего BiLSTM-энкодера используются для одновременного извлечения сущностей и отношений. Такой подход обеспечивает стремление модели к согласованным предсказаниям, поскольку

противоречивые метки (когда токен одновременно помечается и как сущность, и как отношение) приводят к увеличению общей функции потерь модели.

Дальнейшие работы усилили логические ограничения в обучении: семантическая функция потерь, учитывающая априорные символические знания [13], позволила повысить когерентность предсказаний; исследованы взаимодействия задач NER и RE в единой архитектуре [14]; предложены модели для иерархической мультиклассовой классификации, гарантирующие непротиворечивость выходов за счет структуры сети [15]. Параллельно развивались линии структурно согласованных подходов извлечения информации (information extraction, далее IE): формулировка NER как разбора зависимостей [16], а в [17] авторы предложили генеративные универсальные модели IE. Эффективность подобных подходов растет при дообучении (Domain-adaptive pretraining) моделей методом моделирования естественного языка с маскированием части токенов входящей последовательности (Masked Language Modeling – MLM) [18], в этой же работе показано, что снижение функции потерь (и связанной с ним специфичной для MLM метрики perplexity) имеет прямую корреляцию с улучшением производительности модели на прикладных задачах.

1.3. Псевдоразметка и формирование словаря BIO-меток

Другим направлением, близким нашей задаче, является автоматическая разметка данных (далее псевдоразметка) с помощью генеративных моделей трансформеров (GPT, generative pretrained transformer). Авторы модели GPT-3 показали способность модели выполнять произвольные задачи с минимальной настройкой (one-shot/few-shot learning) [19]. В работе [20] продемонстрировано, что открытые LLM (например, LLaMA, Falcon) могут эффективно аннотировать тексты, приближаясь по качеству к модели GPT-4. В то же время отдельные работы указывают на риски применения генеративного искусственного интеллекта (ИИ) для подобных целей [21], в первую очередь это риск переноса в итоговый датасет галлюцинаций и предвзятости (biases). Стоит заметить, что указанные работы сфокусированы на задачах аннотации текстов в «свободной форме», т. е. в них не рассматриваются специализированные подходы к аннотации на уровне последовательности токенов, как этого требует BIO-формат разметки для задачи NER/RE.

В нашей работе мы опираемся на эти идеи и применяем модели GPT для полностью автоматической BIO-разметки специализированного русскоязычного корпуса обращений в службу ИТ-поддержки.

2. ПСЕВДОРАЗМЕТКА BIO

Для эксперимента использовался доменно-специфичный корпус, включающий в себя свыше 660000 текстовых обращений в службу ИТ-поддержки (датасет преимущественно на русском языке, порядка 20% текстов на английском). Доменные понятия сформированы на базе онтологии.

Из онтологии мы получили номенклатуру из 92 меток, куда кроме классов и отношений были включены специальные метки «UNK-x» (отдельно для классов, отношений/свойств и типов), а также вспомогательное отношение «I_A», обозначающее связь «is-a».

Рис. 1. Схема формирования словарей меток классов.

Рассматривается задача преобразования исходных текстов в датасет, который в дальнейшем (см. разд. 4) использован для обучения модели NER/RE. Для каждого текста требуется сгенерировать три параллельные последовательности BIO-меток. Первая последовательность фиксирует класс сущности (в том числе не входящий в онтологию класс), вторая – класс отношения (в том числе не входящее в онтологию отношение), третья, более абстрактная, последовательность указывает тип: токен относится к классу (CLA), отношению (REL), не распознан (UNK) либо не несет семантической нагрузки (O).

Специальные метки семейства UNK содержатся во всех трех типах: UNC-C – для неизвестных классов, UNK-R – для неизвестных отношений, UNK – для обозначения неопределенного типа. Метки UNK нужны для выполнения поставленной

задачи по двум причинам: во-первых, они позволяют модели корректно обработать неизвестную сущность без принудительного отнесения ее к ближайшему известному классу или отношению; во-вторых, они формируют отдельный набор данных. Токены, которые были размечены как UNK-C и UNK-R, после объединения в слова, образуют множество лексических единиц, которые после векторизации (создания эмбеддингов) можно кластеризовать. Кластеры образуют множество синсетов, позволяющих выявить кандидатов на новые классы и отношения/свойства, которые могут расширять онтологию. На рис. 1 показана схема разделения состава онтологии на подмножества классов и отношений. Объединенная онтология указана на рисунке как Seed.

На процесс псевдоразметки наложено ограничение, которое определяет максимально возможный объем разметки, что вызывает необходимость тщательно оптимизировать как объем корпуса, так и длину инструкций (промптов), передаваемых языковой модели.

2.1. Методика выбора модели для псевдоразметки

Для оценки способности языковых моделей к BIO-разметке были выбраны четыре известных корпуса для задач NER, размеченных вручную (далее бенчмарки, англ. benchmark):

- CoNLL 2003 [22] содержит 20 тыс. газетных предложений, размеченных девятью аббревиатурными метками; он считается классическим англоязычным эталоном;
- WikiAnn [23] представлен в двух версиях: англоязычной (en) и русскоязычной (ru), каждая включает примерно 40 тыс. объектов и аббревиатурных меток, что позволяет проверить мультиязычность моделей;
- Корпус WNUT 17 [24] – значительно меньший по объему (около 5.5 тыс. объектов), он отличается тем, что здесь метки являются самостоятельными понятиями: *corporation, creative work* и т. п., в отличие от остальных, где метки реализованы как трехбуквенные индексы.

Испытаны 12 LLM: 8 проприетарных (модели от поставщиков OpenAI и Anthropic) и 4 открытых модели (Llama3, Mistral, DeepSeek). Промпт содержит инструкции и примеры разметки [25] (см. приложение 1). Каждой модели было отправлено 2000 заданий на разметку, после чего была определена оптимальная

модель для псевдоразметки целевого корпуса текстов, которая показала максимальные значения метрик.

При тестировании моделей в качестве критерия оценивания использовалась метрика F1 двух типов. Token F1 рассматривает каждый токен отдельно: показатель может быть высоким, если модель верно предсказывает метки большинства токенов в последовательности. Seqeval F1 – более строгий критерий; предсказание считается верным только тогда, когда модель верно определяет и границы, и тип всей сущности. Разница между Token F1 и Seqeval F1 показывает способность модели удерживать целостность последовательности меток на уровне сущности целиком.

2.2. Методика псевдоразметки целевого корпуса текстов

В процессе псевдоразметки на каждой итерации языковая модель одновременно могла использовать все 92 метки. На этапе постобработки метки последовательности разделялись на два непересекающихся множества меток (классы и отношения) и третье производное производное множество, которое фиксирует только тип токена: класс, отношение, неизвестное (UNK) или токен без семантической нагрузки (O). Это гарантирует, что три BIO-последовательности взаимно исключаются и логически согласованы.

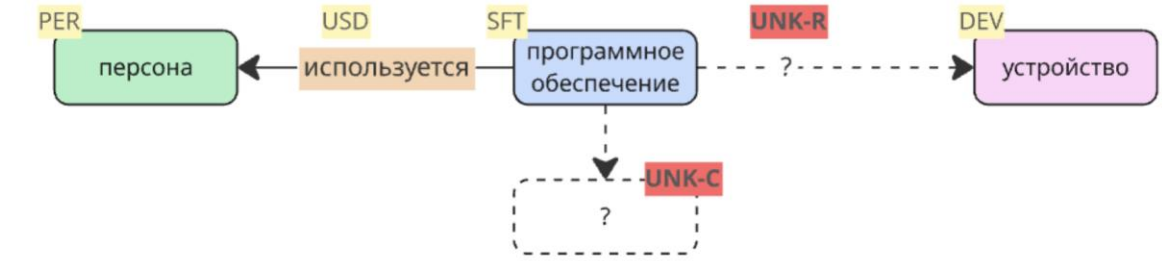
Промпт состоит из четырех частей: инструкция, пример разметки, текст для разметки и словарь меток с кратким описанием семантики каждой метки (см. приложение 2).

2.3. Иллюстрация разметки

Иллюстрация конвейера обработки представлена, как условная мини-онтология (на рис. 2 вверху), сущности которой преобразуются в набор меток, а далее применяются к предложению и раскладываются на три параллельные BIO-последовательности: классы (CLA set), отношения (REL set) и типы токенов (TYPE set), показано на рисунке внизу. Элементы, отсутствующие в мини-онтологии, размечены как UNK-C или UNK-R.

Для демонстрации (на рис. 2) использована примитивная онтология с тремя классами (PER – персона, SFT – программное обеспечение, DEV – устройство) и одним отношением (USD – используется). В предложении также присутствуют неизвестное отношение с меткой UNK-R и неизвестный класс с меткой UNK-C, которых

нет в онтологии, но они явно выражены в тексте. Это позволяет системе помечать кандидатов для последующего расширения онтологии.



Текст	Иван запустил Zoom из под Машиного планшета удаленно.						
Аннотация	Иван	запустил	Zoom	из под	Машиного	планшета	удаленно.
	PER	USD	SFT	UNK-R	PER	DEV	UNK-C
CLA set	PER	O	SFT	O	PER	DEV	UNK-C
REL set	O	USD	O	UNK-R	O	O	O
TYPE set	CLA	REL	CLA	REL	CLA	CLA	CLA

Рис. 2. Иллюстрация разметки на примере мини-онтологии.

Например, слово *«удаленно»* помечается как UNK-C, так как понятия *«способ»/«режим»* не имеют подходящего класса в онтологии; а словосочетание *«из под»* – как UNK-R, поскольку в онтологии нет подходящего отношения (например, им может стать новое отношение *«выполнено»*).

2.4. Формирование подмножества из корпуса данных для разметки

Для соблюдения ограничений внешних ресурсов необходимо сформировать репрезентативное подмножество объектов из корпуса данных (далее подвыборка) для псевдоразметки. Для каждого текстового объекта мы сгенерировали эмбеддинг при помощи модели из [1], затем провели кластеризацию и вычислили геометрические медианы кластеров. Предполагалось, что для сохранения семантической репрезентативности подвыборки нужно выбрать объекты, расположенные в пределах заданного радиуса от геометрической медианы каждого кластера в пространстве векторных представлений объектов корпуса данных.

На первом шаге, для уменьшения вычислительной сложности, размерность эмбеддингов была снижена с 768 (размер выходного скрытого слоя модели) до 128 с помощью анализа главных компонент (PCA):

$$X' = \mathrm{PCA}(X), \qquad X \in \mathbb{R}^{N \times D},$$

где X – исходная матрица эмбеддингов N объектов размерности D, а X' – матрица после снижения размерности.

К эмбеддингам пониженной размерности затем применялся алгоритм кластеризации HDBSCAN [26]:

$$L = \mathrm{HDBSCAN}(X', m),$$

где $L = \{-1, 0, 1, \ldots, K-1\}$ – вектор кластерных меток (–1 означает шум), а m – гиперпараметр минимального размера кластера по количеству объектов.

Для каждого обнаруженного кластера $C_k \subseteq X'$, где $k = 0, \ldots, K-1$, вычисляли геометрическую медиану по алгоритму Вайцфельда [27]:

$$\mu_k = \underset{\mu}{\operatorname{argmin}} \sum_{x \in C_k} ||x - \mu||_2, \mu_k \in \mathbb{R}^D,$$

которая выступает как координата «наиболее типичной» точки кластера k.

Для каждого кластера C_k мы подбирали такой радиус r_k, что если оставить все объекты $x_i \in \mathrm{C}_k$, удовлетворяющие условию

$$||x - \mu||_2 \leq r_k,$$

то общее число выбранных объектов по всем кластерам не превышает лимит 90000 (лимит был задан в соответствии с внешними ограничениями).

Радиусы r_k можно находить либо итеративным двоичным поиском, либо выбрав общий квантиль расстояния от медиан для всех кластеров.

Итоговая подвыборка имеет вид

$$X_{\mathrm{distil}} = \bigcup_{k=0}^{K-1} \{x_i \in C_k : ||x_i - \mu_k||_2 \leq r_k\}.$$

Таким образом, можно обобщить порядок действий. Сначала сжимались эмбеддинги исходных текстов (PCA), затем кластеризовались (HDBSCAN) и характеризовались с помощью геометрических медиан кластеров. Объекты для псевдоразметки отбирались по расстоянию до медианы – подход, позволяющий эффективно сформировать подвыборку из исходного корпуса данных при минимальной потере репрезентативности.

3. АРХИТЕКТУРА И ОБУЧЕНИЕ МОДЕЛИ NER/RE

Наличие 92 категорий в словаре меток свидетельствует о необходимости снижения вычислительной сложности и компенсации дисбаланса меток в выборке. Было предложено реализовать параллельный процесс разметки тремя множествами меток: CLA, REL, TYPE. Модель NER/RE представляет собой три нейросети, которые получают на вход выходной вектор каждого токена с последнего скрытого слоя энкодерной модели BERT (показана на рис. 3, далее базовый энкодер). Сети были обучены параллельно на одной и той же входящей последовательности токенов. Рассмотрены два варианта архитектуры: полносвязные слои и слои с механизмом внимания. Для каждого варианта мы протестировали 2, 4 и 8 слоев нейросетей, уменьшая размер скрытого представления вдвое на каждом слое до тех пор, пока он оставался большим или равным числу выходов соответствующего набора меток для каждой сети.

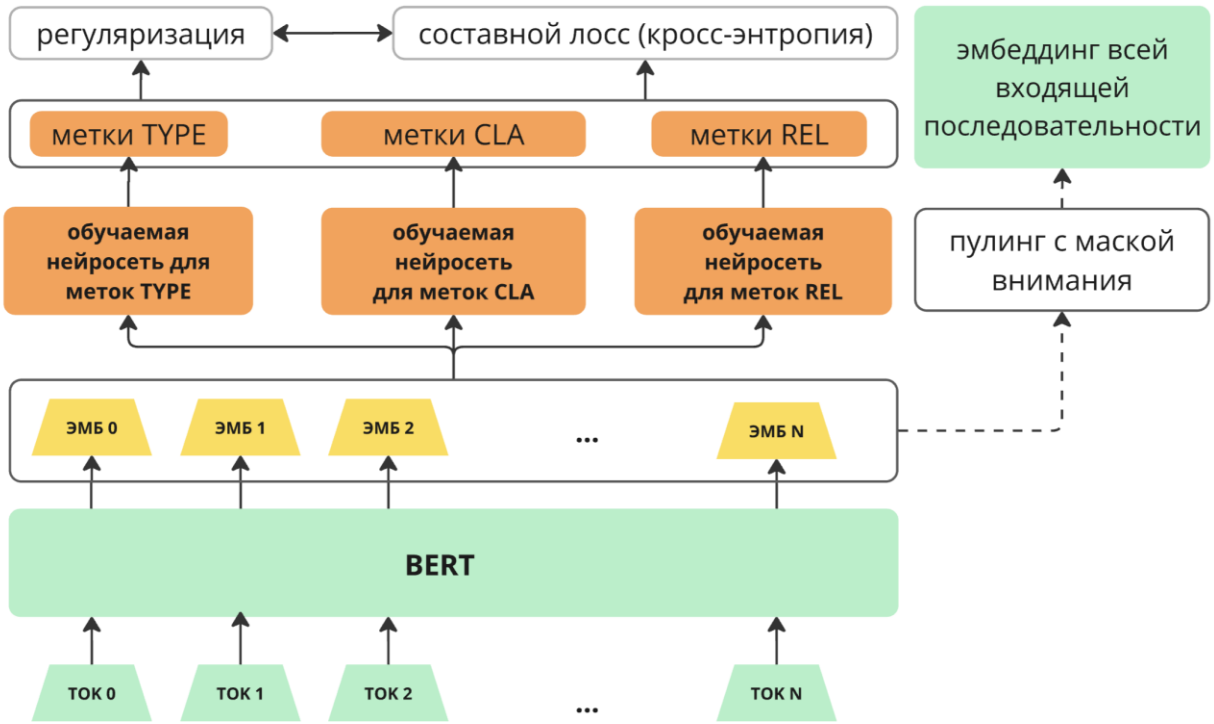


Рис. 3. Архитектура модели NER/RE.

3.1. Функция потерь для задачи NER/RE с регуляризацией

Представление токенов и последовательности реализовано следующим образом. Пусть $X = (x_1, \dots, x_n)$ – матрица входящей последовательности токенов.

Базовый энкодер порождает скрытые состояния $H = f_{\text{enc}}(X) \in \mathbb{R}^{n \times d}$. Каждая дополнительная нейросеть сопоставляет скрытому состоянию вектора выхода внутреннего слоя (логиты токенов) по своему отдельному словарю меток:

$$\hat{y}^{(a)} = f_a(h_i) \in \mathbb{R}^{|C_a|},\ a \in \{\text{type}, \text{class}, \text{rel}\}.$$

Дополнительно архитектура модели позволяет получать эмбеддинг всей последовательности через усреднение состояний всех токенов с учетом маски внимания базового энкодера (пулинг, от англ. mean pooling), за исключением отступов (паддингов, от англ. padding) [28]. Кодирование последовательности целиком потребуется на этапе обработки UNK-наборов, для формирования синсетов расширения онтологии:

$$h_{\text{seq}} = \frac{\sum_{i=1}^{n} m_i h_i}{\max(\sum_{i=1}^{n} m_i, \varepsilon)},\ \varepsilon = 10^{-9},\ m_i \in \{0,1\}.$$

Этот прием принят в моделях BERT, ориентированных на векторные представления всей входящей последовательности сразу, поскольку дает более стабильные и выразительные эмбеддинги, чем кодирование одного только CLS-токена (особенно для задач семантического поиска и кластеризации).

Функция потерь L (далее лосс, от англ. loss function) является суммой кросс-энтропийных потерь по всем трем сетям. Дополнительно введена специальная регуляризационная функция логической согласованности предсказаний

$$\mathcal{L} = \sum_{a} \mathcal{L}_{\text{head}}^{(a)} + \lambda \mathcal{L}_{\text{reg}}, \qquad a \in \{\text{type}, \text{class}, \text{rel}\},$$

где для каждой сети функция лосс рассчитана по стандартной формуле

$$\mathcal{L}_{\text{head}}^{(a)} = -\frac{1}{N} \sum_{i=1}^{N} t_i^{(a)\top} \log \text{softmax}\left(y_i^{(a)}\right),$$

$$\left[\text{softmax}(y_i^{(a)})\right]_k = \frac{\exp\left(y_{i,k}^{(a)}\right)}{\sum_{j=1}^{K_a} \exp\left(y_{i,j}^{(a)}\right)}.$$

Регуляризатор $\mathcal{L}_{\text{reg}}$ увеличивает значение $\mathcal{L}$ в случае, если предсказания модели не проходят проверку согласованности, и тем самым штрафует модель за логические ошибки в выходных последовательностях меток.

Пусть $\hat{T}_i = \text{argmax softmax}\left(\hat{y}_i^{(a)}\right)$ – предсказанный тип токена i, а $w_i = \max \text{softmax}\left(\hat{y}_i^{(a)}\right)$ оценивает степень его предсказания («уверенность предсказания»). Введем предсказательные метки для классов и отношений: C_i и R_i соответственно, а также «скобки Айверсона» $[P] \in \{0, 1\}$, равные 1, если предсказание истинно. Обозначим метки a_i для класса и b_i для отношения:

$$a_i = [C_i \neq \mathcal{O}],\ b_i = [R_i \neq \mathcal{O}],$$

где $\mathcal{O}$– обозначение O-токена (фоновый токен, которому присваивается класс outside) из BIO-разметки. Тогда индикатор нарушения для токена i равен v_i:

$$v_i = \left[\hat{T}_i = CLA\right]([C_i = \mathcal{O}] \vee [R_i \neq \mathcal{O}]) + \left[\hat{T}_i = REL\right]([C_i \neq \mathcal{O}] \vee [R_i = \mathcal{O}]) +$$
$$+ \left[\hat{T}_i = \mathcal{O}\right]([C_i \neq \mathcal{O}] \vee [R_i \neq \mathcal{O}]) + \left[\hat{T}_i = UNK\right](1 - [C_i \neq \mathcal{O}] \oplus [R_i \neq \mathcal{O}]),$$

где $\vee$ – логическое «или», $\oplus$ – исключающее «или» (XOR). Тогда можно записать $a \oplus b = a + b - 2ab$.

После подстановки получим

$$v_i = \left[\hat{T}_i = CLA\right](1 - a_i(1 - b_i)) + \left[\hat{T}_i = REL\right](1 - b_i(1 - a_i)) +$$
$$+ \left[\hat{T}_i = \mathcal{O}\right](a_i + b_i - a_i\, b_i) + \left[\hat{T}_i = UNK\right](1 - a_i - b_i + 2a_i\, b_i).$$

Поскольку события (факт разметки) взаимно исключают друг друга, для каждого токена активной будет только одна скобка – v_i , которая принимает значение 1 или 0. Итоговый регуляризатор с весами уверенности по TYPE может быть записан как

$$\mathcal{L}_{\text{reg}} = \frac{1}{N}\sum_{i=1}^{N} w_i v_i.$$

Обозначим множество сетей как $\mathcal{A} = \{\text{type},\ \text{class},\ \text{rel}\}$, а кросс-энтропию как $CE(t, p) = -t^{\top}\log p$ (для one-hot вектора t). Тогда лосс примет вид

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left[\sum_{a\in\mathcal{A}} CE\left(t_i^{(a)}, \text{softmax}\left(y_i^{(a)}\right)\right) + \lambda w_i v_i\right].$$

Такая запись делает логические ограничения частью цели обучения. Были проведены тестовые циклы обучения как с регуляризатором ($\lambda > 0$), так и без него ($\lambda = 0$).

3.2. Оценка модели

Для оценки обученной модели NER/RE в качестве метрики оптимизации мы выбрали точность (далее – метрика precision), под которой понимаем долю корректно размеченных токенов среди всех токенов, которым модель присвоила ненулевую (не-«O») метку класса или отношения. В контексте обогащения графа знаний это позволяет минимизировать число ложноположительных фактов в RDF-графе. Каждая ошибочно добавленная сущность или связь нарушает семантическую согласованность и требует ручной очистки, тогда как пропущенные факты могут быть дополнительно извлечены на последующих итерациях конвейера. Это определяет приоритет метрики precision при умеренном снижении полноты, т. е. доли корректно размеченных токенов данного класса среди всех токенов данного класса в выборке (далее – метрика recall).

4. СОЗДАНИЕ И ОБОГАЩЕНИЕ ГРАФА ЗНАНИЙ

Для включения извлеченных фактов (в форме триплетов субъект – предикат – объект) в граф знаний мы разработали конвейер, преобразующий исходные данные и результаты BIO-разметки в формат RDF. На первом шаге из реляционной базы данных были выбраны основные таблицы: *Tasks* (заявки), *Companies* (компании-клиенты), *Devices* (оборудование). Эти таблицы были экспортированы, и для каждой записи был создан экземпляр соответствующего класса онтологии. Так, например, для строки в таблице *Tasks* генерируется экземпляр класса «*ServiceRequest»* (заявка). Поля заявки (тема, описание проблемы, комментарий решения, трудозатраты, временные метки и т. д.) преобразуются в литералы или связи графа. Приведение к соответствию (далее маппинг, от анг. mapping) объектов базы данных классам онтологии обеспечивает начальное наполнение графа знаний структурированной информацией из базы данных, схематично показано на рис. 4. На этом этапе было извлечено подмножество из 10000 заявок, которые далее трансформировались в начальный граф для последующего обогащения моделью NER/RE.

На втором шаге выполнялось извлечение знаний из текста при помощи ранее обученной модели NER/RE (разд. 4). Для каждой заявки извлекались тексто-

вые поля (тема, описание, комментарии) и соединялись в один текстовый фрагмент. Модель NER/RE обрабатывала текстовый фрагмент и возвращала последовательность меток для токенов.

Последовательности токенов длиной n с BIO-метками одного класса $\{B^c, I_1^c \dots I_n^c\}$ объединялись в единую именованную сущность, экземпляр соответствующего класса или отношения онтологии. Каждому такому объекту присваивалось уникальное имя URI (Uniform Resource Identifier, в пространстве имен экземпляров, например http://itog.it/ExtractedEntity/UUID).

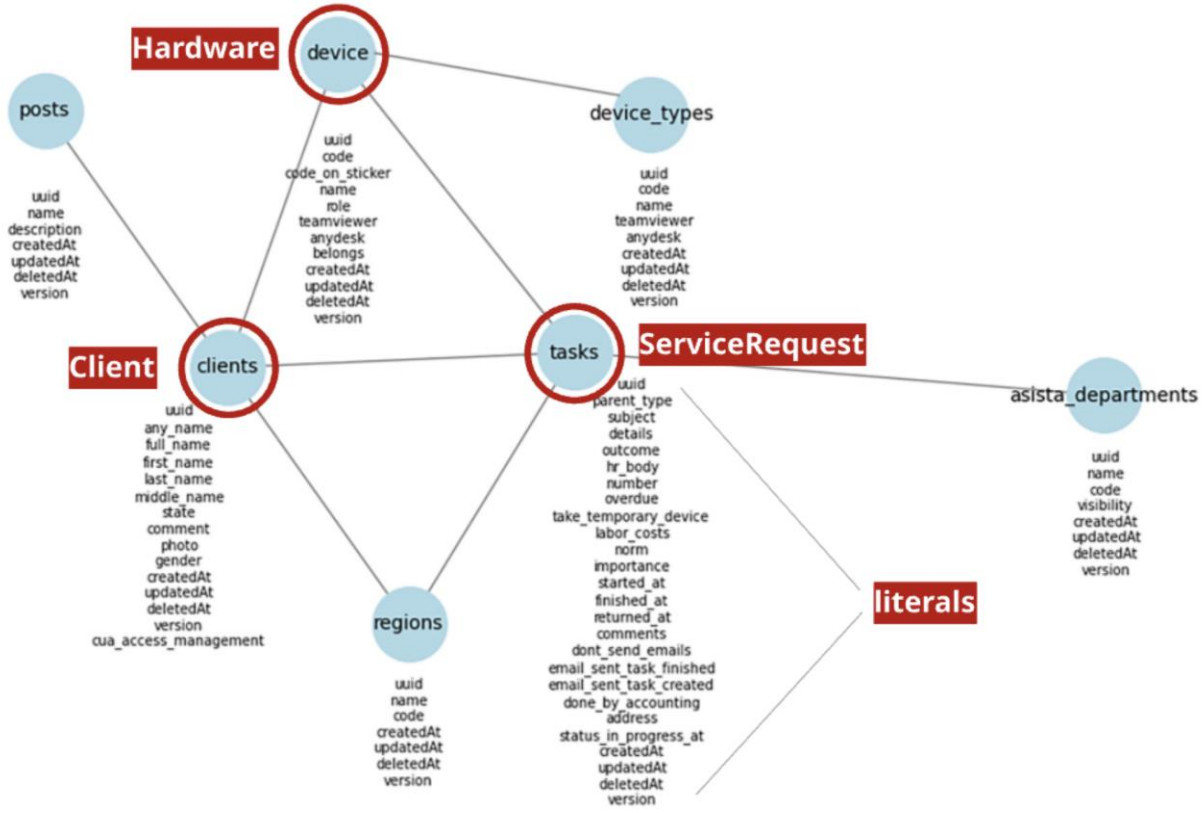


Рис. 4. Схема сопоставления таблиц из базы данных классам онтологии.

Затем на основе предсказанных отношений проверялся шаблон: если была обнаружена последовательность «сущность – отношение – сущность» (например, «сервер обеспечивает приложение») и при этом предсказание соответствовало онтологическому свойству, которое допустимо между классами этих сущностей, то формировалась RDF-тройка, связывающая две извлеченные сущности данным свойством.

5. РЕЗУЛЬТАТЫ

5.1. Тестирование и выбор модели для BIO псевдоразметки

Модели сравнивались на четырех бенчмарках (CoNLL 2003, WikiAnn (en), WikiAnn (ru), WNUT 17) по двум типам метрик F1. На рис. 5 показаны диаграммы распределения метрик, сгруппированных по датасетам. Модели отмечены разными цветами. На верхней диаграмме показаны значения метрики Token F1, на нижней – значения Seqeval F1 для каждой из моделей.

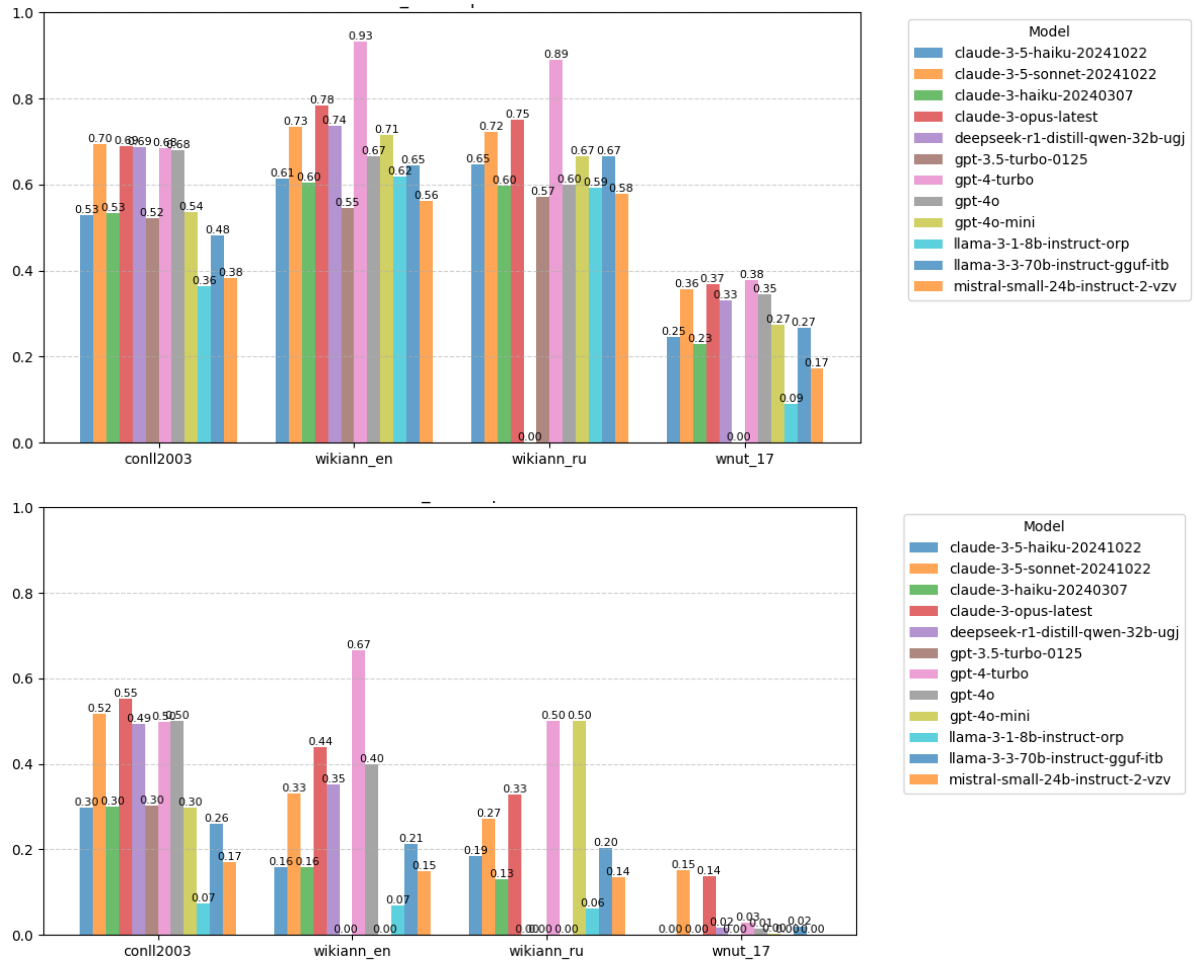


Рис. 5. Сравнительная диаграмма метрик Token F1 и Seqeval F1.

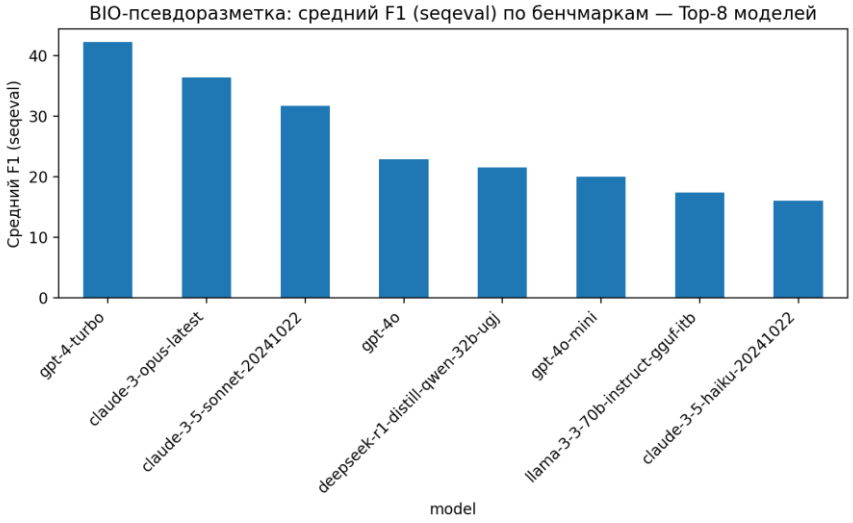


Рис. 6. Сравнение моделей по усредненной Seqeval F1.

Модель GPT-4-turbo на задаче BIO-разметки превосходит конкурентов в большинстве случаев. Интересно, что эта модель на момент проведения экспериментов не являлась самой современной, но показала лучшие результаты. На рис. 6 показана диаграмма метрики Seqeval F1, усредненная по всем испытаниям. Самые худшие результаты получены на датасете WNUT 17, предположительно это может быть вызвано со спецификой меток: модели сложнее проводить разметку, если сами метки представлены большими последовательностями символов или содержат в себе самостоятельный смысл.

Таким образом, можно по среднему Seqeval F1 среди всех датасетов сделать следующие выводы: по усредненной оценке лидируют крупные модели; модель GPT-4-turbo входит в число лучших по среднему и занимает первое место на двух наборах WikiAnn. На английском новостном корпусе CoNLL 2003 первое место у модели Claude-3-opus, а WNUT 17 остается сложным для всех претендентов из списка моделей-кандидатов для BIO-разметки (см. приложение 3).

Результаты эксперимента показали, что: наилучшим практическим выбором является модель GPT-4-turbo; она стабильно работает на мультиязычных бенчмарках, демонстрирует высокую метрику precision, и дает лучшую/сопоставимую метрику Seqeval F1 на мультиязычных корпусах.

5.2. Формирование подмножества выборки и псевдоразметка

Несмотря на то что модель GPT-4-turbo продемонстрировала лучшие результаты на бенчмарках, для финальной разметки корпуса данных была выбрана

модель GPT-4o. Такое решение обусловлено следующими факторами: (1) значительная экономия затрат при сопоставимом качестве на русскоязычных данных; (2) лучшие показатели на мультиязычных корпусах, соответствующих профилю датасета; (3) оптимальное соотношение precision/recall (см. приложение 2).

Приблизительный расчет позволил получить количество объектов, которые можно будет разметить в рамках внешних ограничений: 90000 объектов к разметке или порядка 13% от исходного корпуса данных. Была проведена фильтрация по методу, указанному в п. 2.4. На рис. 7 показана двумерная проекция кластеризованных эмбеддингов объектов исходного корпуса данных (слева) и отфильтрованного подмножества (справа).

После применения модели GPT-4o были извлечены три подмножества токенов, размеченных как типы, классы и отношения. В результате получился размеченный корпус данных объемом 81505 объектов, содержащий 3037348 токенов, из них 773324 размеченных (размеченным считается слово с любой меткой кроме “O”) (см. приложение 4).

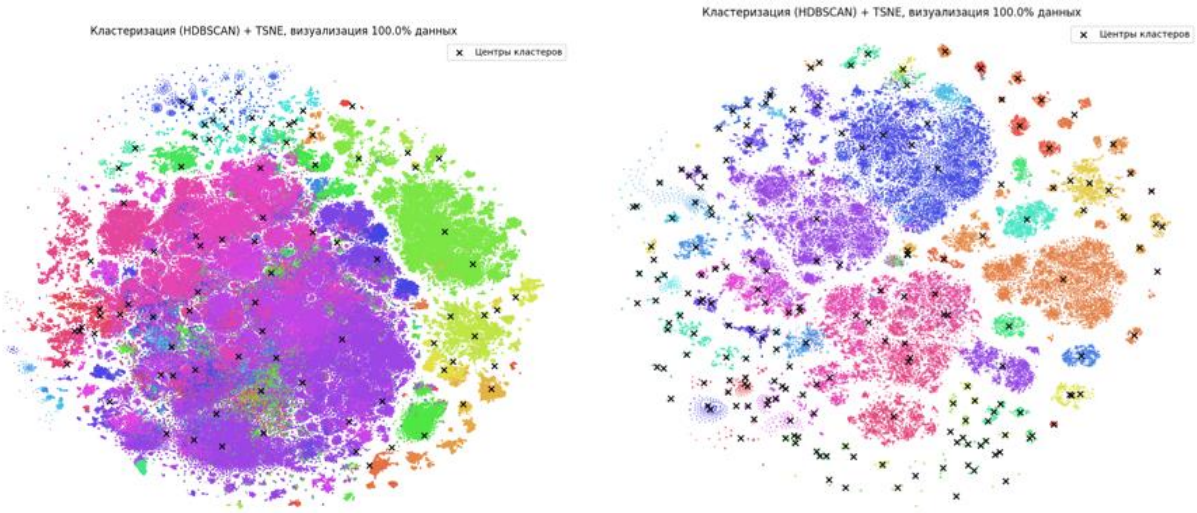


Рис. 7. Двухмерная проекция эмбеддингов до и после фильтрации.

5.3. Дообучение базовой модели

В качестве базовой выбрана модель XLM-RoBERTa-large (24 слоя трансформера, ~550 млн параметров), предобученная на корпусе CommonCrawl для 100 языков [29]. Чтобы учесть специфику и увеличить качество модели на доменных данных, мы провели дополнительное доменное дообучение модели. Обучающий корпус для задачи MLM (общим объемом > 650 тыс. объектов) был составлен из

исходного корпуса данных, состоящего из текстов внутренней базы заявок техподдержки и документации.

При обучении в последовательности маскировались 15% токенов, что является рекомендуемым гиперпараметром для задачи моделирования естественного языка методом маскирования. Обучение проводилось в течение 33 эпох, из табл. 1 видно, что лосс и метрика perplexity значительно уменьшились после обучения, что указывает на рост способности модели производить доменно-специфичные векторные репрезентации текстов на естественном языке. Дообучение проводилось на одном GPU ускорителе NVIDIA A100 в течение 250 часов.

Табл. 1. Сравнение метрик до и после дообучения модели XLM-RoBERTa-large на валидационной (вал.) и тестовой (тест.) выборках.

	Loss (вал.)	Perplexity (вал.)	Loss (тест.)	Perplexity (тест.)
Исходная модель	1.52	4.55	1.53	4.60
Дообученая модель	0.52	1.68	0.52	1.68

5.4. Обучение модели NER/RE

На этапе подбора гиперпараметров было установлено, что предлагаемая в разд. 4 регуляризация негативно влияет на метрики (см. приложение 5). Архитектуры с механизмом внимания показали наилучшие результаты. В табл. 2 показаны сравнительные метрики по лучшим тестовым циклам обучения, в колонке «head» отмечены два типа сетей – с механизмом внимания (att, attention) и без него (ff, ffed forward), в колонке «reg» указан коэффициент регуляризации, а в «layers» – количество слоев нейросети. Полная таблица по всем тестовым циклам обучения модели NER/RE представлена в приложении 5.

Табл. 2. Сравнительные метрики моделей.

			f1			accuracy			precision			recall		
head	reg	layers	class	rel	avg	class	rel	avg	class	rel	avg	class	rel	avg
ff	0	2	0.56	0.72	0.64	0.45	0.58	0.51	0.85	0.95	0.90	0.45	0.58	0.51
att	0	4	**0.57**	**0.76**	**0.67**	**0.46**	**0.61**	**0.54**	**0.85**	**0.96**	**0.91**	**0.46**	**0.61**	**0.54**

5.5. Оценка модели

Оценка в табл. 2 выполнена на всей входящей последовательности токенов с учетом фонового класса «O», здесь также использована метрика Token F1. Такой режим демонстрирует высокую метрику precision (≈ 0.91) при умеренных значениях recall (≈ 0.54), что соответствует выбранной стратегии: модель не стремится предсказать все потенциально информативные токены, однако при назначении метки CLASS или REL демонстрирует высокую надежность.

Табл. 3. Список наиболее сложных лейблов и процент ошибок в протоколе оценки «без O»

Labels hardest by recall				Labels hardest by precision		
Label	**Type**	**Confusion (% err)**		**Label**	**Type**	**Confusion (% err)**
PES	class	PCS (26.9%)		CMO	class	SDY (27.8%)
RRE	class	STM (22.0%)		TLE	class	AET (25.0%)
RNT	class	DDY (13.6%)		**SST**	**class**	**SDY (40.4%)**
QIE	**class**	**CCY (38.4%)**		**CPO**	**class**	**SDY (68.6%)**
CAM	class	HWE (19.5%)		CON	class	CUN (17.3%)
UNK-R	relation	O (14.9%)		**HRS**	**relation**	**HER (42.1%)**
HSR	**relation**	**HRS (60.9%)**		HCB	relation	HER (25.0%)
HRS	**relation**	**HER (42.1%)**		HIE	relation	HLY (33.3%)
RIS	relation	UES (22.6%)		HAT	relation	HIS (16.7%)
HES	relation	HRS (20.9%)		UES	relation	HIE (15.0%)
HIS	relation	HES (21.3%)		HME	relation	HES (33.3%)
DEF	**relation**	**O (37.5%)**		**HVE**	**relation**	**HRS (40.0%)**

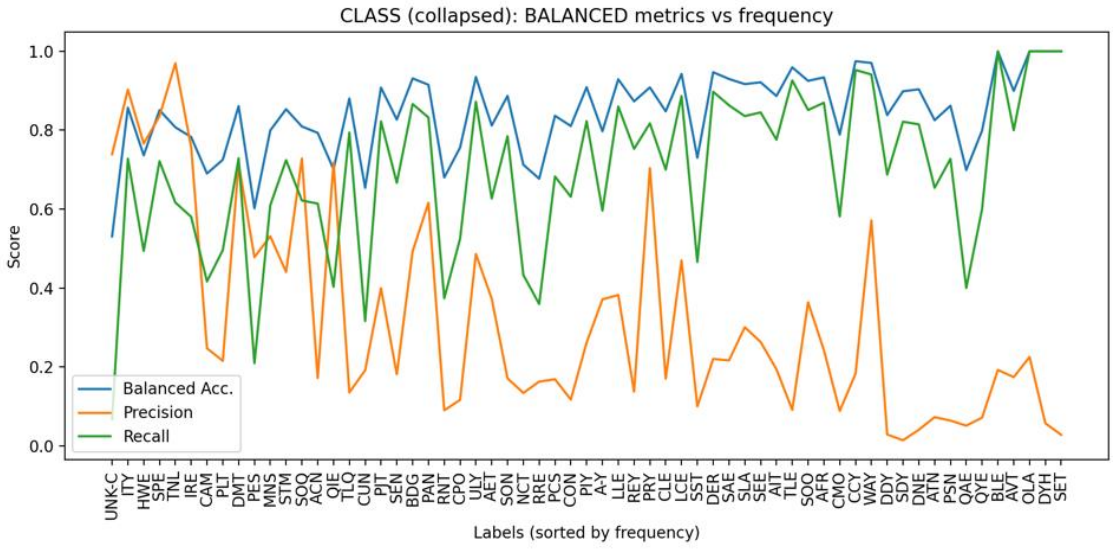


Рис. 8. График изменения метрик в зависимости от частоты метки.

После завершения обучения мы провели оценку модели в протоколе «без O», т. е. с исключением вклада фоновых токенов. В этом режиме внутриклассовая метрика точности (accuracy) и F1 остаются высокими, тогда как метрика precision систематически падает по мере уменьшения поддержки метки – ожидаемый эффект «длинного хвоста». На рис. 8 показано влияние частоты меток в датасете на метрики. В табл. 3 приведены наиболее сложные метки по метрикам recall и precision с указанием топ-ошибок. Причинами ошибок можно назвать семантическую близость объектов и токен-уровневую природу BIO аннотации.

5.6. NER-обогащение графа знаний

Представлены суммарные количественные и качественные эффекты применения предложенного конвейера к корпоративной базе ITSM. Эти результаты демонстрируют не только рост объема знаний, но и то, что новые сущности и связи корректно вписываются в структуру графа знаний и остаются логически согласованными с ограничениями онтологии.

Из табл. 4 видно, что уже на первом этапе обработки каждый объект преобразуется около 12 триплетов, каждый триплет можно интерпретировать как факт, полученный из данных.

Табл. 4. Сравнение базы данных и графов знаний.

Данные	Источник и объем	Узлы	Явные триплеты	Выведенные	Всего фактов	Expansion ratio[1]
SQL	10 000 строк tasks (+ ключи внутри строк)	≈ 10000	–	–	–	–
Базовый RDF	прямой маппинг SQL → RDF	13899	115661	191262	**306923**	2.65
NER- RDF	базовый граф + автоматически извлеченные факты	66194	279546	705330	**984876**	3.52

После применения модели NER извлекается еще более 50 тыс. сущностей и образуется более 160 тыс. связей, автоматический вывод новых фактов приумножает их в 2.5 раза. В результате общий объем знаний, извлеченный из 10 тыс. заявок в службу поддержки, приближается к миллиону триплетов, а коэффициент

[1] Expansion ratio рассчитывается как пропорция общего количества фактов в графе к явным триплетам (которые явно были импортированы в граф), является важным показателем вывода новых фактов автоматически.

расширения растет с 2.65 до 3.52. Доля выводимых фактов достигла приблизительно 71.6% от всего графа, что свидетельствует об интенсивном расширении – автоматически выведенные факты составляют большую часть графа. При этом целостность структуры графа сохранилась: как и прежде, все узлы связаны в единую компоненту (благодаря узлам-классам из онтологии).

Таким образом, каждая исходная строка базы данных трансформируется в сотни взаимно связанных фактов, что существенно повышает выразительность данных и открывает возможность сложных запросов на языке SPARQL, недоступных ни на уровне языка SQL, ни в графе знаний в первоначальном состоянии.

Из табл. 5 видно, что увеличение узлов и ребер сопровождается ожидаемым «разрежением» сети, но появление ненулевой кластеризации (0.08) показывает, что новые сущности формируют связанные смысловые кластеры. Это означает, что новые узлы образовали локальные группы. Например, упоминания названий программного обеспечения могли связаться через общий класс *SoftwarePackage* или через одно устройство, класс *Hardware*, к которому они относятся.

Табл. 5. Топологические изменения графа после NER обогащения.

Метрика	До NER	После NER	Δ
Количество узлов	13899	66194	+ 376%
Количество ребер	40094	151670	+ 278%
Средняя степень ⟨k⟩	5.77	4.58	−21%
Плотность	$4.1 \cdot 10^{-4}$	$6.9 \cdot 10^{-5}$	↓x6
Кластеризация $\bar{C}$	0.00	0.08	+0.08

Получившаяся семантическая сеть обладает свойствами безмасштабной структуры: распределение степеней узлов монотонно убывает, в распределении существуют узлы с относительно высокой степенью – это узлы онтологии и наиболее частые сущности, такие как, например, типовые услуги; большинство же узлов имеет малую (в десятки раз меньше) степень. Такой характер свидетельствует о корректности интеграции новых данных: они не превратили граф в хаотичную структуру, где все связано со всем, а вписались в уже существующую семантическую конструкцию, образуя смысловые кластеры вокруг известных концепций.

На рис. 9 сравнивается одна и та же сущность, соответствующая исходному объекту данных (заявка в службу поддержки) до и после NER-обогащения. Как видно, в исходном состоянии с центральным узлом связан ограниченный набор объектов, а после обогащения появилось девять новых связей, которых не было в базе данных. Они описывают временные интервалы, режим, организацию, ответственных лиц и т. д., вся эта значимая информация содержалась имплицитно (не явно) в тексте заявки на естественном языке.

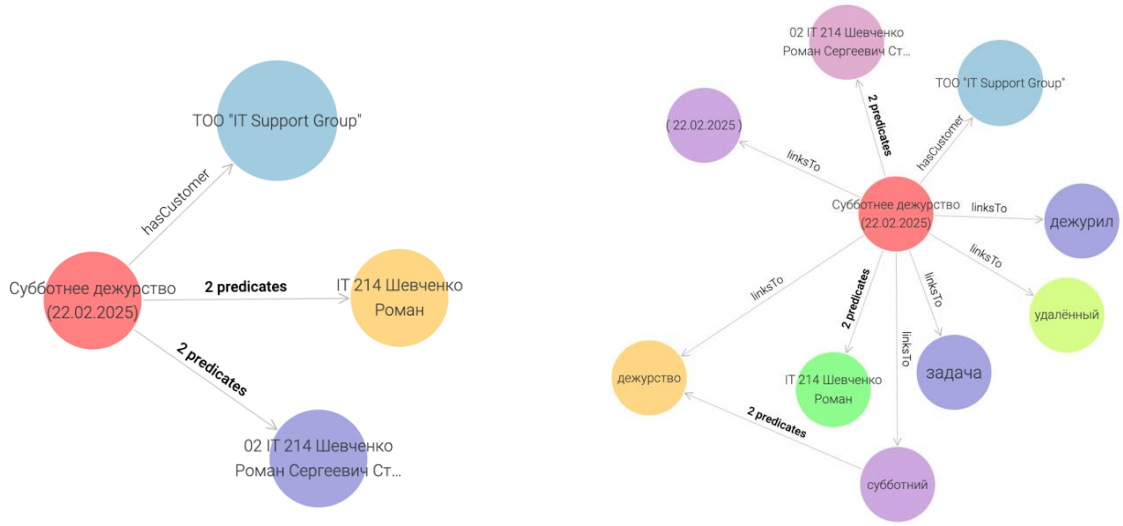


Рис. 9. Фрагмент графа знаний до и после NER-обогащения.

Модель автоматически обнаружила экземпляры классов, после чего между ними были установлены взаимосвязи в соответствии с аксиомами, которые в явном виде содержатся в онтологии. Кластерный и семантический анализ сущностей с метками категорий UNK-C и UNK-R далее позволит выявить новые классы и отношения для расширения онтологии.

5.7. Обработка UNK-сущностей и расширение онтологии

Все сущности с метками UNK-C (кандидаты классов) и UNK-R (кандидаты отношений) были кластеризованы для синсетов, пригодных для экспертной валидации и включения в онтологию.

Для каждой UNK-сущности были построены эмбеддинги способом, указанным в разд. 4.

Начальный набор содержал 244679 элементов. На первом этапе мы устранили шумовые и несловесные единицы и провели фильтрацию по косинусной близости, задав порог как 10-й перцентиль распределения расстояний внутри пар отдельно для классов и для отношений. После этой процедуры осталось 50630 кандидатов классов и 141317 кандидатов отношений. Дополнительная проведенная очистка по длине токена (не менее трех символов) сократила выборку до 44631 и 112695 объектов соответственно. Далее эмбеддинги были понижены до размерности 128 в помощью метода главных компонент (PCA), после чего была выполнена кластеризация методом *k*-средних (mini-batch *k*-means) с гиперпараметром *k*=12 для каждого набора объектов. Для каждого кластера была вычислена геометрическая медиана по алгоритму Вайцфельда; вокруг этой медианы сформировалась компактная окрестность фиксированного размера (не более 50 ближайших), исходные объекты которой сформировали синсет для экспертного анализа. На рис. 10 показан эффект фильтрации по геометрической медиане в кластерах объектов, размеченных как UNK-CLA (неизвестные представители классов, извлеченные из входящих текстов), представлена двухмерная проекция до и после фильтрации. В результате мы получили плотные и однородные локальные семантические множества (приложение 6).

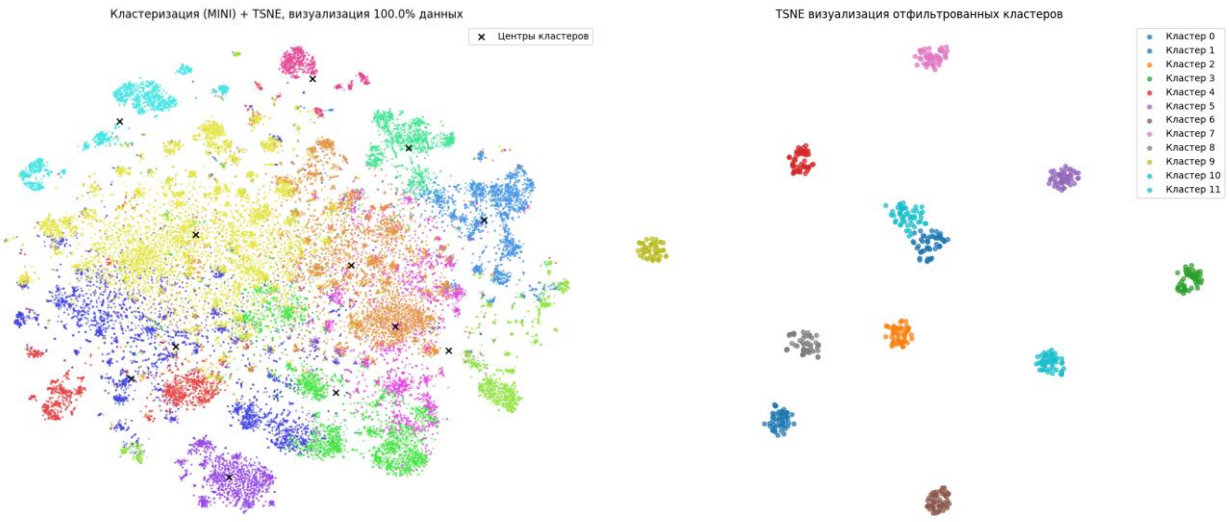


Рис. 10. Двухмерная проекция эмбеддингов UNK до и после фильтрации.

Качественный анализ синсетов, проведенный экспертом, показал семантическую когерентность групп и позволил предложить онтологические спецификации. Так, синсет, в котором преобладают такие глагольные формы, как «выдал», «поменяли», «сохранил», «протестил», «восстановили», был проинтерпретирован как класс процессов модификации: *rdfs:label – ModificationProcess; rdfs:comment – «тип процесса, отвечающий за изменение конфигурационных единиц, настроек и параметров»*. Такие свойства, как «ответственность» и «роль», в этом случае наследуются из онтологии и необходимы для дополнения RDF-троек.

6. ПРАКТИЧЕСКАЯ ЗНАЧИМОСТЬ И СЦЕНАРИЙ ИСПОЛЬЗОВАНИЯ

Автоматизация обогащения графа знаний существенно снижает трудоемкость поддержки актуальности знаний в ITSM. В отличие от ручного дополнения онтологии экспертами, предложенный метод может быть реализован как процесс в фоновом режиме, с минимальным вмешательством. Интеграция результатов NER/RE с онтологическими ограничениями обеспечивает высокую точность извлечения знаний из текстов: модель реализует консервативную стратегию извлечения и добавляет в граф только те факты, в которых уверена и которые соответствуют схеме. Благодаря этому свойству достигается семантическая согласованность данных – система не нарушает правила и ограничения, определенные онтологией. Показательно, что даже после почти пятикратного увеличения граф сохраняет строгую структуру и связность.

Автоматизированный граф удовлетворяет специфические запросы, которые недостижимы в реляционной модели данных. На рис. 11 показан интерфейс GraphDB Chat: LLM, подключенная к обогащенному графу, отвечает на вопрос *«что у компании „XXXXXX“ ломается чаще всего и что можно им предложить?»*. Далее по фактам графа выводятся наиболее проблемные устройства клиента и генерируется предложение, релевантное поведению клиента, которое хранится как связанный набор фактов в графе знаний.

Этапы формирования ответа:

1. SPARQL-подзапрос извлекает агрегированную статистику поломок (COUNT по сущностям класса Hardware, связанных с ServiceRequest и ClientCompany).

2. Второй запрос ищет оборудование того же типа, отсутствующее у клиента (FILTER NOT EXISTS { ?new hw itsmo:hasCustomer :XXX }).
3. LLM интерпретирует результаты, ранжируя их по частоте.

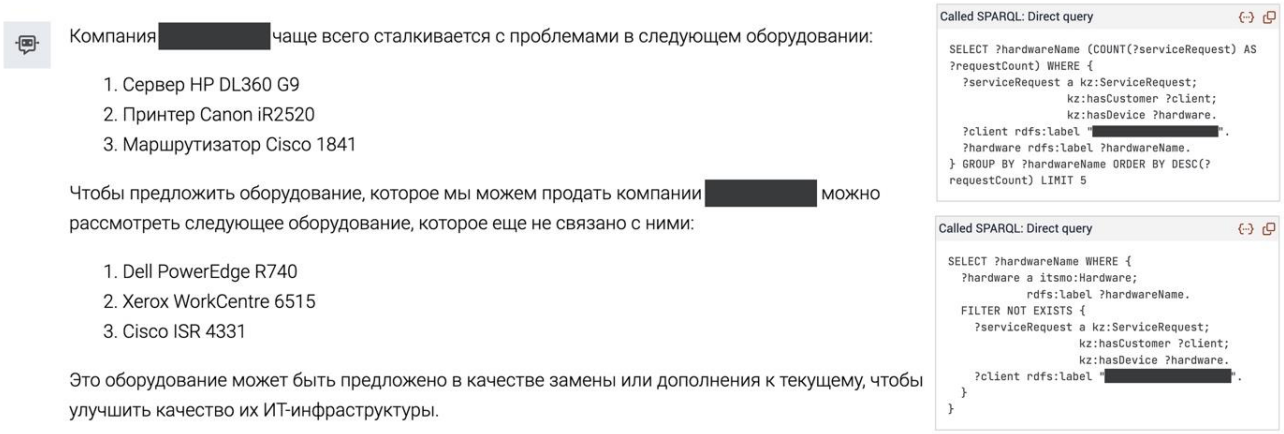


Рис. 11. Интерфейс обработки запросов к графу знаний на естественном языке.

В табличной SQL-схеме такой синтез осложнен: факты о состоянии оборудования хранятся строками разных таблиц, а понятие отсутствующего, но совместимого оборудования не представлено явно. Графовая модель вместе со специализированной языковой моделью позволяют задать это условие SPARQL-конструкцией. Таким образом, продемонстрировано, что наш конвейер повышает экспрессивность запросов и открывает дорогу к рекомендательным сценариям без ручной подготовки витрин данных.

ЗАКЛЮЧЕНИЕ

Разработан и экспериментально валидирован воспроизводимый цикл «граф – модель – граф» для автоматического обогащения корпоративного графа знаний и полуавтоматического расширения онтологии в домене ITSM. Ключевыми компонентами предложенного подхода являются:

- единая онтологическая схема, объединяющая онтологию DOLCE с доменной онтологией ITSMO;
- значительный BIO-корпус (3 млн токенов, 92 метки);
- оригинальная архитектура модели NER/RE;
- механизм UNK-меток.

Проведено расширение онтологии двенадцатью новыми классами и двенадцатью новыми отношениями. Включение осуществлялось с проверкой согласованности по иерархиям и ограничениям онтологии.

Предложенный цикл – от разметки до онтологически оформленных конструктов – опирается на приоритет точности, что отмечено в разд. 4: мы сознательно выбираем консервативную модель с высокой метрикой precision, поскольку для полуавтоматической интеграции важнее надежность единичного решения, чем исчерпывающий охват. На практике такая стратегия уменьшает долю ложноположительных кандидатов в наборах UNK, снижает нагрузку на эксперта и ускоряет прохождение этапов «UNK – синсет – спецификация – включение».

В совокупности это демонстрирует, что UNK-метки, усиленные контекстными эмбеддингами и компактной кластерной выборкой, служат эффективным механизмом выявления ранее отсутствующих понятий и свойств. Метод обеспечивает воспроизводимое, управляемое и реактивное расширение онтологии полуавтоматическим способом.

Эффективность подхода подтверждена количественными метриками: увеличение графа в 4.76 раза (с 13899 до 66194 узлов), рост коэффициента расширения с 2.65 до 3.52 при сохранении логической согласованности. UNK-метки для полуавтоматического определения новых классов и отношений позволяют значительно сократить время необходимо для актуализации онтологий.

Представлен набор методов создания синтетических наборов данных с использованием словаря онтологии как пространства меток, показаны возможности явной и неявной логических регуляризаций в процессе обучения модели-энкодера, проведена апробация результатов исследования в рабочих сценариях.

СПИСОК ЛИТЕРАТУРЫ

1. *Khalov A., Ataeva O*. Automating Ontology Mapping in IT Service Management: A DOLCE and ITSMO Integration // Data Science Journal. 2025. Vol. 24. P. 23. https://doi.org/10.5334/dsj-2025-023

2. *Borgo S. et al.* DOLCE: A descriptive ontology for linguistic and cognitive engineering // Applied Ontology. 2023. Vol. 17, No. 1. P. 45–69.

3. *IT Service Management Ontology (ITSMO).* Canonical resolver; catalog entry in LOV “IT Service Management Ontology (itsmo)”.
https://w3id.org/itsmo; ontology.it; lov.linkeddata.es (Accessed: 08 August 2025).

4. *Gruber T.R.* A translation approach to portable ontology specifications // Knowledge Acquisition. 1993. Vol. 5, No. 2. P. 199–220. https://doi.org/10.1006/knac.1993.1008

5. *Gruber T.R.* Toward principles for the design of ontologies used for knowledge sharing // International Journal of Human-Computer Studies. 1995. Vol. 43, No. 5–6. P. 907–928. https://doi.org/10.1006/ijhc.1995.1081

6. *Smith B.* Ontology (Science) // Formal Ontology in Information Systems, IOS Press, 2008. P. 21–35. https://doi.org/10.1038/npre.2008.2027.2

7. *Studer R., Benjamins V. R., Fensel D.* Knowledge Engineering: Principles and Methods // Data & Knowledge Engineering. 1998. Vol. 25, No. 1–2. P. 161–197. https://doi.org/10.1016/S0169-023X(97)00056-6

8. *El Yamami A. et al.* An ontological representation of ITIL framework service level management process // Smart Data and Computational Intelligence. Lecture Notes in Networks and Systems. 2019. Vol. 66. P. 88–94. https://doi.org/10.1007/978-3-030-11914-0_9

9. *Barrasa J., Webber J.* Building Knowledge Graphs: A Practitioner's Guide. O'Reilly Media, 2023. 250 p.

10. *Hogan A., Blomqvist E., Cochez M. et al.* Knowledge Graphs. Morgan & Claypool Publishers, 2021. 257 p.

11. *Valiente M.-C., Vicente-Chicote C., Rodriguez D.* An Ontology-Based and Model-Driven Approach for Designing IT Service Management Systems // Int. J. of Service Science, Management, Engineering, and Technology. 2011. Vol. 2 (2). P. 65–81.

12. *Miwa M., Bansal M.* End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures // Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Berlin, Germany: Association for Computational Linguistics, 2016. P. 1105–1116. https://doi.org/10.18653/v1/P16-1105

13. *Xu J., Zhang Z., Friedman T., Liang Y., Van den Broeck G.* A Semantic Loss Function for Deep Learning with Symbolic Knowledge // Proceedings of the 35th International Conference on Machine Learning (ICML). PMLR, 2018. Vol. 80. P. 5502–5511. URL: https://proceedings.mlr.press/v80/xu18h.html

14.*Sun K., Zhang R., Mensah S., Mao Y., Liu X.* Learning Implicit and Explicit Multi-task Interactions for Information Extraction // ACM Transactions on Information Systems. 2023. Vol. 41, No. 2. P. 1–29. https://doi.org/10.1145/3533020

15.*Giunchiglia E., Lukasiewicz T.* Coherent Hierarchical Multi-label Classification Networks // Advances in Neural Information Processing Systems 33 (NeurIPS 2020). 2020. URL: https://proceedings.neurips.cc/paper/2020/file/6dd4e10e3296fa63738371ec0d5df818-Paper.pdf

16.*Yu J., Bohnet B., Poesio M.* Named Entity Recognition as Dependency Parsing // Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL). 2020. P. 6470–6476. https://doi.org/10.18653/v1/2020.acl-main.577

17.*Lu Y., Liu Q., Dai D., Xiao X., Lin H., Han X., Sun L., Wu H.* Unified Structure Generation for Universal Information Extraction // Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2022. P. 5755–5772. https://doi.org/10.18653/v1/2022.acl-long.395

18.*Gururangan S., Marasović A., Swayamdipta S., Lo K., Beltagy I., Downey D., Smith N.A.* Don't Stop Pretraining: Adapt Language Models to Domains and Tasks // Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL). 2020. P. 8342–8360. https://doi.org/10.18653/v1/2020.acl-main.740

19.*Brown T. B. et al.* Language Models are Few-Shot Learners // Advances in Neural Information Processing Systems. 2020. Vol. 33.
URL: https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

20.*Alizadeh M. et al*. Open-source LLMs for text annotation: a practical guide for model setting and fine-tuning // Journal of Computational Social Science. 2025. Vol. 8. P. 17. https://doi.org/10.1007/s42001-024-00345-9

21.*Eiras F. et al.* Position: Near to Mid-term Risks and Opportunities of Open-Source Generative AI // Proceedings of the 41st International Conference on Machine Learning (ICML 2024). Proceedings of Machine Learning Research. 2024. Vol. 235. P. 12348–12370. URL: https://proceedings.mlr.press/v235/eiras24b.html

22.*Tjong Kim Sang E. F., De Meulder F*. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition // Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003. Edmonton, Canada: Association for Computational Linguistics, 2003. P. 142–147.

https://doi.org/10.3115/1119176.1119195

23.*Pan X., Zhang B., May J., Nothman J., Knight K., Ji H.* Cross-lingual Name Tagging and Linking for 282 Languages // Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vancouver, Canada: Association for Computational Linguistics, 2017. P. 1946–1958.
https://doi.org/10.18653/v1/P17-1178

24.*Derczynski L., Nichols E., van Erp M., Limsopatham N.* Results of the WNUT2017 Shared Task on Novel and Emerging Entity Recognition // Proceedings of the 3rd Workshop on Noisy User-generated Text (W-NUT 2017). Copenhagen, Denmark: Association for Computational Linguistics, 2017. P. 140–147.
https://doi.org/10.18653/v1/W17-4418

25.*Brown T.B. et al.* Language Models are Few-Shot Learners // Advances in Neural Information Processing Systems 33 (NeurIPS 2020). 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

26.*Campello R.J.G.B., Moulavi D., Zimek A., Sander J.* Hierarchical density estimates for data clustering, visualization, and outlier detection // ACM Transactions on Knowledge Discovery from Data. 2015. Vol. 10, No. 1. P. 1–51.
https://doi.org/10.1145/2733381

27.*Vardi Y., Zhang C.-H.* A modified Weiszfeld algorithm for the Fermat–Weber location problem // Mathematical Programming. 2001. Vol. 90. P. 559–566.
https://doi.org/10.1007/PL00011435

28.*Reimers N., Gurevych I.* Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks // Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics, 2019. P. 3982–3992. https://doi.org/10.18653/v1/D19-1410

29.*Hugging Face.* XLM-RoBERTa (large): specs (24 layers, ~550M params). 2020–2024.
URL: https://huggingface.co/transformers/v3.4.0/pretrained_models.html

30.*Côté M.-A. et al*. TextWorld: A Learning Environment for Text-Based Games // Computer Games (CGW@IJCAI 2018). 2019. Vol. 1017 (CCIS). P. 41–75.
https://doi.org/10.1007/978-3-030-24337-1_3

31.*Russell S., Norvig P.* Artificial Intelligence: A Modern Approach. 4th ed. Pearson, 2020. Chapter 11: Automated planning.

32.*Schmidhuber J.* Gödel Machines: Self-Referential Universal Problem Solvers Making Provably Optimal Self-Improvements // Artificial General Intelligence. 2007. P. 199–226. https://doi.org/10.1007/978-3-540-68677-4_7

33.*Yin X. et al.* Gödel Agent: A Self-Referential Agent Framework for Recursively Self-Improvement // Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2025). 2025. P. 27890–27913. https://aclanthology.org/2025.acl-long.1354/

34. *Атаева О.М., Серебряков В.А.* Онтология цифровой семантической библиотеки LibMeta // Информатика и ее применения. 2018. Т. 12, № 1. С. 2–10.

Приложение 1. Промпт, используемый для тестирования способностей моделей к BIO-разметки. В sorted_labels и text подаются все метки классов из словаря классов и текст для разметки соответственно.

```
Below is a sentence. You have to annotate each token using BIO format for named entity recognition.
The possible entity classes are: {sorted_labels}
Use the 'O' tag if the token does not belong to any entity.
Sentence: {text}
Return the annotations as valid JSON with a single key 'ner', whose value is a list of the same length as the token list.
Each element in that list should be a BIO tag (e.g., 'B-PER', 'I-PER', 'O').
For example: [\"B-PER\", \"O\", \"B-LOC\", ...]
```

Приложение 2. Промпт, используемый LLM псевдоразметки. Фрагмент-промпта для многоклассовой NER-разметки состоит из двух основных частей: часть с примерами, приведенными в требуемом JSON формате, примеры содержат все типы меток, вторая часть – это сам промпт, который содержит в себе рабочую нагрузку в виде текста, подлежащего разметке и список меток с описанием семантики каждой метки (в промпте не указаны аннотации каждого класса).

```
few_shot_examples = (
    Example 1:
    Input: Не работает Windows, не удается подключится к базе
    Output (JSON):
    {
      "ner": {
        "Не": "O",
        "работает": "O",
        "Windows": "B-SPE",
        ",": "O",
        "не": "O",
        "удается": "O",
        "подключится": "B-RIS",
        "к": "O",\n'
        "базе": "B-UNK-C"       // unknown class (e.g. data-base)
      }
    }

system_prompt = (
    "You're an expert NER and Relation Extraction model specialized in the IT domain and ITIL. Annotate each
token using BIO tagging for both Named Entity Recognition (NER) and relation extraction."
```

Приложение 3. Результаты эксперимента с тестированием разных моделей на BIO-бенчмарках.

		F1		**precision**		**recall**	
benchmark	model	seq.	bin	seq.	bin	seq.	bin
conll2003	claude-3-5-haiku-20241022	29.9%	52.8%	27.2%	52.2%	33.1%	53.4%
	claude-3-5-sonnet-20241022	51.6%	**69.5%**	47.4%	**66.7%**	56.6%	72.5%
	claude-3-haiku-20240307	30.1%	53.5%	28.0%	51.7%	32.4%	55.3%
	claude-3-opus-latest	**55.2%**	69.0%	**53.6%**	65.5%	**56.9%**	73.0%
	deepseek-r1-distill-qwen-32b-ugj	49.3%	68.6%	45.1%	61.2%	54.3%	**78.2%**
	gpt-3.5-turbo-0125	30.3%	52.3%	30.4%	52.2%	30.2%	52.3%
	gpt-4-turbo	49.7%	68.4%	48.0%	62.3%	51.6%	75.9%
	gpt-4o	50.2%	67.9%	47.5%	63.1%	53.1%	73.6%
	gpt-4o-mini	29.9%	53.6%	28.0%	49.2%	32.0%	58.9%
	llama-3-1-8b-instruct-orp	7.3%	36.4%	5.3%	25.2%	11.7%	65.8%
	llama-3-3-70b-instruct-gguf-itb	26.0%	48.2%	23.8%	44.5%	28.7%	52.5%
	mistral-small-24b-instruct-2-vzv	17.1%	38.3%	15.2%	35.8%	19.6%	41.1%
wikiann_en	claude-3-5-haiku-20241022	15.8%	61.3%	14.3%	75.0%	17.8%	51.8%
	claude-3-5-sonnet-20241022	33.1%	73.3%	29.7%	85.1%	37.5%	64.4%
	claude-3-haiku-20240307	15.8%	60.4%	14.1%	74.8%	17.9%	50.7%
	claude-3-opus-latest	44.0%	78.3%	40.8%	86.7%	47.7%	71.3%
	deepseek-r1-distill-qwen-32b-ugj	35.1%	73.6%	31.9%	84.9%	39.1%	64.9%
	gpt-3.5-turbo-0125	0.0%	54.5%	0.0%	75.0%	0.0%	42.9%
	gpt-4-turbo	**66.7%**	**93.3%**	**66.7%**	**87.5%**	**66.7%**	**100.0%**
	gpt-4o	40.0%	66.7%	50.0%	80.0%	33.3%	57.1%
	gpt-4o-mini	0.0%	71.4%	0.0%	71.4%	0.0%	71.4%
	llama-3-1-8b-instruct-orp	6.9%	61.8%	5.2%	57.5%	10.2%	66.9%
	llama-3-3-70b-instruct-gguf-itb	21.3%	64.5%	18.4%	74.1%	25.3%	57.1%
	mistral-small-24b-instruct-2-vzv	15.0%	56.1%	13.5%	71.8%	16.8%	46.1%
wikiann_ru	claude-3-5-haiku-20241022	18.5%	64.8%	15.9%	74.8%	22.1%	57.1%
	claude-3-5-sonnet-20241022	27.1%	72.2%	23.1%	83.8%	32.6%	63.5%
	claude-3-haiku-20240307	13.1%	59.8%	11.2%	69.5%	15.8%	52.4%
	claude-3-opus-latest	32.9%	75.0%	29.5%	84.5%	37.4%	67.4%
	deepseek-r1-distill-qwen-32b-ugj	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	gpt-3.5-turbo-0125	0.0%	57.1%	0.0%	66.7%	0.0%	50.0%
	gpt-4-turbo	**50.0%**	**88.9%**	**50.0%**	**100.0%**	**50.0%**	**80.0%**
	gpt-4o	0.0%	60.0%	0.0%	60.0%	0.0%	60.0%
	gpt-4o-mini	50.0%	66.7%	50.0%	60.0%	50.0%	75.0%
	llama-3-1-8b-instruct-orp	6.1%	59.3%	4.4%	52.4%	9.6%	68.4%
	llama-3-3-70b-instruct-gguf-itb	20.4%	66.6%	17.3%	73.6%	24.9%	60.9%
	mistral-small-24b-instruct-2-vzv	13.5%	57.9%	12.3%	71.7%	15.0%	48.6%
wnut	claude-3-5-haiku-20241022	0.0%	24.5%	0.0%	17.9%	0.0%	39.0%
	claude-3-5-sonnet-20241022	**15.2%**	35.6%	10.6%	26.5%	**27.1%**	54.2%
	claude-3-haiku-20240307	0.0%	23.0%	0.0%	16.1%	0.0%	40.3%
	claude-3-opus-latest	13.6%	36.7%	**9.6%**	26.2%	23.3%	61.5%
	deepseek-r1-distill-qwen-32b-ugj	1.7%	33.0%	1.1%	22.2%	3.3%	**64.4%**
	gpt-3.5-turbo-0125	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	gpt-4-turbo	2.8%	**37.7%**	2.1%	**26.8%**	4.5%	63.2%
	gpt-4o	1.4%	34.5%	1.0%	24.1%	2.3%	60.7%
	gpt-4o-mini	0.2%	27.4%	0.1%	20.2%	0.3%	42.8%
	llama-3-1-8b-instruct-orp	0.1%	9.0%	0.1%	4.9%	0.6%	55.3%
	llama-3-3-70b-instruct-gguf-itb	2.0%	26.8%	1.3%	19.1%	3.7%	44.6%
	mistral-small-24b-instruct-2-vzv	0.0%	17.3%	0.0%	12.4%	0.0%	28.7%

Приложение 4. Распределение меток в синтезированном датасете

[Распределение cat_rel_none]:

O: 2232028

CLA: 667908›

REL: 105496

UNK: 31916

[Распределение class_tags]:

O: 2369495	SOQ: 8429	PCS: 2136	CLE: 815	DNE: 303
UNK-C: 125902	ACN: 6258	ULY: 1992	SAE: 736	ATN: 258
ITY: 100113	QIE: 5575	SON: 1954	SLA: 698	PSN: 229
HWE: 81238	TLQ: 4470	CON: 1808	CCY: 651	SDY: 213
SPE: 80533	CUN: 3938	RRE: 1769	TLE: 632	BLE: 151
TNL: 62252	PJT: 3242	PIY: 1472	SOO: 564	AVT: 146
IRE: 49713	PAN: 2772	PRY: 1124	DDY: 515	SET: 115
CAM: 18107	SEN: 2545	LLE: 1076	AFR: 513	OLA: 102
PLT: 16653	BDG: 2539	DER: 974	AIT: 511	QYE: 94
PES: 15758	CPO: 2384	SST: 969	SEE: 481	DYH: 82
DMT: 14996	NCT: 2249	LCE: 935	CMO: 437	
MNS: 13858	RNT: 2179	A-Y: 927	WAY: 409	
STM: 1295	AET: 2159	REY: 873	QAE: 370	

[Распределение rel_tags]:

O: 2931877	HOE: 1536	HGN: 679	HCB: 269	_A: 100
RIS: 39290	HRS: 1494	HPS: 671	HIE: 227	HIY: 67
UNK-R: 30466	HSD: 1255	HVE: 511	HER: 184	HRY: 19
HSR: 15628	HNE: 1157	HME: 469	HET: 174	HNH: 18
HIS: 2782	HNS: 1131	HGS: 440	HOY: 168	
HES: 2361	HFD: 863	DEF: 313	HCN: 143	
UES: 1933	HAT: 720	HEN: 279	HLY: 12	

Приложение 5. Синсеты и их преобразование в объект онтологии

Пример класса и его определение:

['выдал',	'Отмена',	'Сбросили',	'добавила',
'поменяли',	'сохранил',	'сменила',	'Выслал',
'сделал',	'сбросил',	'Добавил',	'сохранил',
'Прописал',	'Поменял',	'Узнал',	'переименую',
'Выслал',	'Создал',	'сбросила',	'Сменил',
'сменила',	'сменила',	'Отправил',	'Отмена',
'сохранил',	'Запросила',	'Изменили',	'Выбрал',
'Сменили',	'ввела',	'Узнал',	'Просит',
'Прописала',	'дал',	'Освободил',	'сменил',
'Сменил',	'Выслал',	'изменений',	'Отмена',
'Протестил',	'Выслать',	'Проверил',	'Включена']
'Сменил',	'Readdressed',	'восстановили',	
'Вывел',	'применятся',	'Запросила',	

Пример отношения и его определение:

['Дмитрием.',	'проверить.',	'тимвивера:.',	'евернот.',
'отсутствует.',	'Вн.',	'отсутствует.',	'отсутствует.',
'отсутствует.',	'доставленно.',	'модем.',	'отсутствует.',
'UEFI,',	'ноутбуком.',	'переподкл.',	'переадрес:',
'отсутствует.',	'скайп.',	'Дарье.',	'email.',
'отсутствует.',	'okay,',	'hSPjds.',	'год.',
'отсутствует.',	'Planned.',	'ключи.',	'Жанатов.',
'195.',	'issue.',	'аутлук.',	'Астана.',
'Mail.',	'тех.',	'папка.',	'lift.',
'NdqkyRI.',	'битрикс.',	'Запланир.',	'работает.',
'евернот.',	'отсутствует.',	'Дмитрий.',	'эв.'],
'отсутствует.',	'неделю.',	'Джимайл.',	
'Ugpx.',	'отсутствует.',	'Султан.',	

rdfs:label: ModificationProcess

rdfs:comment: A type of process that handles the modification or change of a configuration item, setting, or parameter.

Потенциальные Properties: hasResponsible, hasAccountable, hasInformed, и т. п. (уже определенные в ITSMO для процессов)

owl:DatatypeProperty: hasNote

rdfs:comment: Holds additional notes or short remarks about the resource's current status or environment.

```
<owl:DatatypeProperty rdf:about="http://ontology.it/itsmo/v1#hasNote">
  <rdfs:domain rdf:resource="http://ontology.it/itsmo/v1#RunnableResource"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:label>hasNote</rdfs:label>
  <rdfs:comment>Additional short remark or note about a re-source.</rdfs:comment>
</owl:DatatypeProperty>
```

AUTOMATIC AND SEMI-AUTOMATIC METHODS FOR DOMAIN KNOWLEDGE-GRAPH CONSTRUCTION AND ONTOLOGY EXPANSION

A. P. Khalov[1] [0009-0005-4584-8245], O. M. Ataeva[2][0000-0003-0367-5575]

[1]*Moscow Institute of Physics and Technology (MIPT), Dolgoprudny, Russia*

[1, 2]*Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Moscow, Russia*

[1]khalov.a@phystech.edu, [2]oataeva@frccsc.ru

Abstract

We present a combined pipeline for knowledge-graph construction and ontology expansion. The approach builds a BIO-tagged corpus via fully automatic LLM-based pseudo-annotation and introduces dedicated UNK reserve categories to capture previously unseen classes and relations. A specialized NER/RE model is trained on a 3-million-token dataset with 92 labels. The model exhibits a conservative quality profile – high precision with moderate recall – suited for safe graph enrichment: integrating the extracted facts expands the graph to ~0.98 million triples, while the expansion ratio (total inferred facts to explicit triples) increases from 2.65 to 3.52, with logical consistency preserved. UNK label pools are converted into stable synsets, enabling semi-automatic ontology expansion; 12 new classes derived from unstructured texts were added. We also demonstrate practical value for querying and analytics using an LLM + SPARQL setup.

Keywords: *ontology, DOLCE, knowledge graph, NER, BIO tagging, RDF/OWL, SPARQL.*

REFERENCES

1. *Borgo S. et al.* DOLCE: A descriptive ontology for linguistic and cognitive engineering // Applied Ontology. 2023. Vol. 17, No. 1. P. 45–69.

2. *IT Service Management Ontology (ITSMO).* Canonical resolver; catalog entry in LOV "IT Service Management Ontology (itsmo)".
https://w3id.org/itsmo; ontology.it; lov.linkeddata.es (Accessed: 08 August 2025).

3. *Khalov A., Ataeva O*. Automating Ontology Mapping in IT Service Management: A DOLCE and ITSMO Integration // Data Science Journal. 2025. Vol. 24. P. 23. https://doi.org/10.5334/dsj-2025-023

4. *Gruber T.R.* A translation approach to portable ontology specifications // Knowledge Acquisition. 1993. Vol. 5, No. 2. P. 199–220. https://doi.org/10.1006/knac.1993.1008

5. *Gruber T.R.* Toward principles for the design of ontologies used for knowledge sharing // International Journal of Human-Computer Studies. 1995. Vol. 43, No. 5–6. P. 907–928. https://doi.org/10.1006/ijhc.1995.1081

6. *Smith B.* Ontology (Science) // Formal Ontology in Information Systems, IOS Press, 2008. P. 21–35. https://doi.org/10.1038/npre.2008.2027.2

7. *Studer R., Benjamins V.R., Fensel D.* Knowledge Engineering: Principles and Methods // Data & Knowledge Engineering. 1998. Vol. 25, No. 1–2. P. 161–197. https://doi.org/10.1016/S0169-023X(97)00056-6

8. *Hogan A., Blomqvist E., Cochez M. et al.* Knowledge Graphs. Morgan & Claypool Publishers, 2021. 257 p.

9. *Barrasa J., Webber J.* Building Knowledge Graphs: A Practitioner's Guide. O'Reilly Media, 2023. 250 p.

10.*El Yamami A. et al.* An ontological representation of ITIL framework service level management process // Proceedings of the 3rd International Conference on Signals, Distributed Systems and Artificial Intelligence (SDSAI 2018). 2019. Springer.

11.*Valiente M.-C., Vicente-Chicote C., Rodriguez D.* An Ontology-Based and Model-Driven Approach for Designing IT Service Management Systems // Int. J. of Service Science, Management, Engineering, and Technology. 2011. Vol. 2 (2). P. 65–81.

12.*Miwa M., Bansal M.* End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures // Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2016. P. 1105–1116. Berlin, Germany: Association for Computational Linguistics. https://doi.org/10.18653/v1/P16-1105

13.*Xu J., Zhang Z., Friedman T., Liang Y., Van den Broeck G.* A Semantic Loss Function for Deep Learning with Symbolic Knowledge // Proceedings of the 35th International Conference on Machine Learning (ICML). PMLR, 2018. Vol. 80. P. 5502–5511. URL: https://proceedings.mlr.press/v80/xu18h.html

14.*Sun K., Zhang R., Mensah S., Mao Y., Liu X.* Learning Implicit and Explicit Multi-task Interactions for Information Extraction // ACM Transactions on Information Systems. 2023. Vol. 41, No. 2. P. 1–29. https://doi.org/10.1145/3533020

15.*Giunchiglia E., Lukasiewicz T.* Coherent Hierarchical Multi-label Classification Networks // Advances in Neural Information Processing Systems 33 (NeurIPS 2020). 2020. URL: https://proceedings.neurips.cc/paper/2020/file/6dd4e10e3296fa63738371ec0d5df818-Paper.pdf

16.*Yu J., Bohnet B., Poesio M.* Named Entity Recognition as Dependency Parsing // Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL). 2020. P. 6470–6476. https://doi.org/10.18653/v1/2020.acl-main.577

17.*Lu Y., Liu Q., Dai D., Xiao X., Lin H., Han X., Sun L., Wu H.* Unified Structure Generation for Universal Information Extraction // Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2022. P. 5755–5772. https://doi.org/10.18653/v1/2022.acl-long.395

18.*Gururangan S., Marasović A., Swayamdipta S., Lo K., Beltagy I., Downey D., Smith N. A.* Don't Stop Pretraining: Adapt Language Models to Domains and Tasks // Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL). 2020. P. 8342–8360. https://doi.org/10.18653/v1/2020.acl-main.740

19.*Brown T.B. et al.* Language Models are Few-Shot Learners // Advances in Neural Information Processing Systems. 2020. Vol. 33. URL: https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

20.*Alizadeh M., Kubli M., Samei Z., Dehghani S., Zahedivafa M., Bermeo J.D., Korobeynikova M., Gilardi F.* Open-source LLMs for text annotation: a practical guide for model setting and fine-tuning // Journal of Computational Social Science. 2025. Vol. 8. Article 17. https://doi.org/10.1007/s42001-024-00345-9

21.*Eiras F. et al.* Position: Near to Mid-term Risks and Opportunities of Open-Source Generative AI // Proceedings of the 41st International Conference on Machine Learning (ICML 2024). Proceedings of Machine Learning Research. 2024. Vol. 235. P. 12348–12370. URL: https://proceedings.mlr.press/v235/eiras24b.html

22.*Tjong Kim Sang, E. F., De Meulder F.* Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In: HLT-NAACL 2003 (CoNLL-2003).

23.*Zhang B., May J., Nothman J., Knight K., Ji H.* Cross-lingual Name Tagging and Linking for 282 Languages. ACL 2017.

24.*Derczynski, L. et al.* Results of the WNUT2017 Shared Task on Novel and Emerging Entity Recognition. W-NUT 2017 (ACL Workshop).

25.*Brown T.B. et al.* Language Models are Few-Shot Learners // Advances in Neural Information Processing Systems. 2020. Vol. 33.

26.*Campello R.J.G.B., Moulavi D., Sander J.* Hierarchical density estimates for data clustering, visualization, and outlier detection // ACM Transactions on Knowledge Discovery from Data (TKDD). 2015. Vol. 10 (1). P. 5. https://doi.org/10.1145/2733381

27.*Vardi Y., Zhang C.-H.* A modified Weiszfeld algorithm for the Fermat–Weber location problem // Mathematical Programming. 2001. Vol. 90. P. 559–566. https://doi.org/10.1007/PL00011435

28.*Reimers N., Gurevych I.* Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks // Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019. P. 3982–3992. Hong Kong, China: Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1410

29.*Hugging Face.* XLM-RoBERTa (large): specs (24 layers, ~550M params). 2020–2024.
URL: https://huggingface.co/transformers/v3.4.0/pretrained_models.html

30.*Côté M.-A. et al*. TextWorld: A Learning Environment for Text-Based Games // Computer Games (CGW@IJCAI 2018). 2019. Vol. 1017 (CCIS). P. 41–75. https://doi.org/10.1007/978-3-030-24337-1_3

31.*Russell S., Norvig P.* Artificial Intelligence: A Modern Approach. 4th ed. Pearson, 2020. Chapter 11: Planning and Acting.

32.*Schmidhuber J.* Gödel Machines: Self-Referential Universal Problem Solvers Making Provably Optimal Self-Improvements // Artificial General Intelligence. 2007. P. 199–226. https://doi.org/10.1007/978-3-540-68677-4_7

33.*Yin X. et al.* Gödel Agent: A Self-Referential Agent Framework for Recursively Self-Improvement // Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2025). 2025. P. 27890–27913. https://aclanthology.org/2025.acl-long.1354/

34.*Ataeva O.M., Serebryakov V.A.* Ontology of the Digital Semantic Library Lib-Meta // Informatics and Its Applications. 2018. Vol. 12, No. 1. P. 2–10 (In Russian).

СВЕДЕНИЯ ОБ АВТОРАХ

ХАЛОВ Андрей Петрович – аспирант МФТИ (ФПМИ), кафедра «Интеллектуальные системы». Область научных интересов: онтологическое моделирование, графы знаний, извлечение знаний из текстов (NER/RE, RAG), многоагентные системы и планирование, применение LLM в корпоративных ИС.

Andrey Petrovich KHALOV – PhD student, Moscow Institute of Physics and Technology (MIPT), Phystech School of Applied Mathematics and Informatics, Department of Intelligent Systems. Research interests: ontological modeling, knowledge graphs, information extraction from text (NER/RE, RAG), multi-agent systems and planning, application of LLMs in enterprise information systems.

email: khalov.a@phystech.edu
ORCID: 0009-0005-4584-8245

АТАЕВА Ольга Муратовна – старший научный сотрудник Вычислительного центра им. А.А. Дородницына Федерального исследовательского центра «Информатика и управление» Российской академии наук, к. т. н. Область научных интересов: системное программирование, базы данных, инженерия знаний и онтологии.

Olga Muratovna ATAEVA – Senior Researcher, Dorodnitsyn Computing Center, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences (FRC CSC RAS); Candidate of Technical Sciences (PhD). Research interests: systems programming, databases, knowledge engineering, ontologies.

email: oataeva@frccsc.ru
ORCID: 0000-0003-0367-5575

Материал поступил в редакцию 11 ноября 2025 года