

О ПОВЫШЕНИИ КАЧЕСТВА RAG ПРИ ВЫЯВЛЕНИИ ПРОТИВОРЕЧИЙ В СУДЕБНЫХ ПОСТАНОВЛЕНИЯХ

Е. А. Козлова¹ [0009-0002-5884-126X], А. М. Баюк² [0009-0002-6721-0005],
М. А. Петрова³ [0009-0006-0393-1400]

^{1,3}*DM Solutions, г. Москва, Россия*

^{2,3}*Институт языкознания РАН, г. Москва, Россия*

¹*e.kozlova@dm-solutions.ru*, ²*alexandra.m.baiuk@gmail.com*,

³*m.petrova@dm-solutions.ru*

Аннотация

Настоящее исследование является продолжением работы над пилотной системой по детекции противоречий в судебных постановлениях, целью которой является выявление противоречий между судебными постановлениями по делам об административных правонарушениях и регулируемыми нормами. Представленный ранее пайплайн, включающий правилую предобработку документов, поиск по репозиторию правовых норм и дообучение LLM с использованием адаптера LoRA, обеспечил приемлемый уровень качества [1], однако имел ряд ограничений, связанных с компонентом RAG. В отличие от LLM, RAG не обучался на юридических текстах, что приводило к потере релевантных для сопоставления фрагментов на этапе поиска сходных текстовых пар, в связи с чем проведен анализ качества RAG и выполнено улучшение RAG-части системы.

Для оценки полноты извлечения данных созданы датасет из 30 документов и автоматические тесты, содержащие все пары релевантных для сопоставления предложений из данного датасета (более ста пар). На этих данных рассчитана полнота для исходной версии модели, использующей эмбединги предложений `sbert_large_nlu_ru` для создания векторной базы и вычисления косинусной близости.

Для повышения полноты извлекаемых данных проведены следующие эксперименты: (1) дообучение исходных эмбедингов на обучающем датасете; (2) добавление модели-ранжировщика с дообучением и без; (3) комбинирование

вариаций «базовые/дообученные эмбединги» и «базовый/дообученный ранжировщик» на нескольких версиях параметра top_k (количество извлекаемых соответствий); (4) замена модели эмбедингов предложений на другие варианты. В результате существенно повысилось качество RAG – с 72% до 90%, однако увеличение количества извлекаемых фрагментов снизило общую точность системы, что требует ее дальнейшей калибровки. Кроме того, проведены реорганизация базы кодексов, расширение тестовой выборки и улучшение алгоритма фильтрации нерелевантных предложений из постановлений.

Ключевые слова: выявление противоречий, юридические тексты, обучение LoRA (Low-Rank Adaptation — низкоранговая адаптация), RTE (Recognizing Textual Entailment — распознавание логического следования), RAG (Retrieval-Augmented Generation — генерация с дополнением по результатам поиска).

ВВЕДЕНИЕ

Настоящая статья является продолжением работы [1], посвященной созданию пилотной модели по выявлению противоречий в судебных постановлениях.

Модель предназначена для выявления противоречий между судебными постановлениями по делам об административных правонарушениях и регулируемыми их нормативными положениями. Целью текущей работы является оптимизация предложенного ранее пайплайна, затрагивающая, с одной стороны, оценку масштабируемости созданного прототипа, с другой — решение проблем, выявленных на предыдущем этапе работы, в частности, улучшение качества RAG-компонента системы.

Пилотная модель показала достаточно высокие метрики: precision = 0.96, recall = 0.69, F1-мера = 0.81, однако использованный в предыдущей версии RAG (Retrieval-Augmented Generation) не обучался на юридических текстах, что приводило к существенной потере полноты фрагментов, извлекаемых на этом этапе, в связи с чем повышение качества RAG стало центральной задачей дальнейшей доработки.

Выявление противоречий является частью задачи распознавания логического следования в тексте (Recognizing Textual Entailment, RTE) и логического вывода на естественном языке (Natural Language Inference, NLI), представляя собой важный аспект понимания текста.

Число работ, посвященных RTE и, в частности, выявлению противоречий, значительно [2–11], см. также библиографию в них.

Исследования в этой области фокусируются на создании классификаций и типологий противоречий [2, 3, 12], формировании аннотированных текстовых корпусов, содержащих пары противоречащих друг другу предложений [5, 13], и на разработке автоматических систем выявления противоречий. За последние несколько лет для решения этой задачи применялся широкий спектр инструментов – от логики, правил и различных лингвистических средств, таких как WordNet [2] в более ранние годы, до применения моделей на основе трансформеров и больших языковых моделей (Large Language Models, LLM) [7, 11] и нейросимволических подходов [14, 15] в недавних работах.

Выявление противоречий актуально для текстов разной тематики. Особый интерес эта задача представляет для юристов, поскольку обнаружение расхождений между документами и регулируемыми их правовыми нормами требует, как правило, существенных временных и материальных затрат. Решение задачи RTE и выявление противоречий в юридических текстах рассматривались в [16–24], материалах конференции COLIEE (Competition on Legal Information Extraction/Entailment) и многих других работах (см. также обзор в [25]).

Генеративные модели вывели задачу выявления противоречий на новый уровень и позволили устанавливать не только противоречия, выражаемые через антонимы или отрицание, но и противоречия, основанные на структурных и лексических контрастах, а также на знаниях о мире (согласно типологии противоречий [3]). LLM успешно находят противоречия между двумя утверждениями или двумя документами и в большинстве случаев предлагают разумное объяснение, почему те или иные утверждения считаются противоречащими друг другу.

Однако, задача сравнения одного документа с крупным репозиторием, включающим законодательные базы, и выявления противоречий между рассматриваемым документом и всей базой данных к настоящему моменту не имеет удовлетворительного решения из-за требуемых ресурсов и сложности

природы противоречий. Лучший результат можно получить, если обучать модель на специальных данных, состоящих из релевантных правовых норм и базы фрагментов с противоречиями, типичными для указанной тематической области [14].

Следуя этой идее, мы разработали пилотную версию системы, предназначенной для выявления противоречий между судебными постановлениями по делам об административных правонарушениях, с одной стороны, и соответствующими правовыми нормами, с другой. Мы также определили набор типов противоречий, наиболее часто встречающихся в подобных юридических документах, и обучили модель их выявлять.

Сценарий работы системы выглядит следующим образом:

- репозиторий документов предварительно заполняется набором необходимых кодексов;
- пользователь загружает один или несколько документов в формате .docx;
- система проверяет, содержат ли загруженные документы утверждения, противоречащие статьям кодексов в репозитории, и, если такие противоречия обнаружены, представляет их пользователю.

Рассмотрим далее описание архитектуры системы, используемые данные, результаты нового этапа доработки системы и дальнейшие планы.

ОБЩАЯ АРХИТЕКТУРА СИСТЕМЫ

Архитектура системы состоит из трех компонентов. Первый компонент включает предварительную обработку документов.

Судебные постановления имеют типовую структуру и включают несколько частей: вводную (содержащую официальные реквизиты и идентификационные данные дела), описательную (содержащую краткое изложение сути заявленных требований), мотивировочную (содержащую логико-правовое обоснование постановления путем установления фактических обстоятельств дела) и резолютивную (итоговую часть судебного акта, содержащую решение суда об удовлетворении или отказе в удовлетворении требований).

Юридически значимая часть каждого судебного решения извлекается с помощью правилowego подхода из резолютивной части – эта часть почти всегда

следует за ключевым словом «Постановил». Далее текст разбивается на предложения с использованием Python-пакета `razdel`¹, а предложения дополнительно фильтруются на предмет удаления заведомо нерелевантных предложений.

В новой реализации улучшен алгоритм такой фильтрации, что позволяет точнее исключать фрагменты, не содержащие потенциальных правовых противоречий, в частности предложения с указанием реквизитов банковских счетов, подписей судей и адресов. Это, с одной стороны, снижает дальнейшую нагрузку на модель, с другой – формально влияет на количественные значения метрик: благодаря фильтрации модель получает на вход меньше простых для анализа предложений, что несколько снижает процент правильных ответов, однако не означает реального ухудшения качества.

С помощью второго компонента мы находим потенциально релевантные фрагменты в репозитории правовых норм для их дальнейшей передачи модели, которая, в свою очередь, будет сравнивать пары вида «гипотеза (фрагмент постановления) – предпосылка (фрагмент кодекса)». В зависимости от эксплицитного указания номера статьи в тексте постановления правовые нормы извлекаются из репозитория либо посредством правил, либо технологией RAG.

Поскольку RAG-обработка занимает больше времени, система сначала пытается извлечь номер релевантных статьи/части/пункта правилами. Правильное извлечение работает достаточно надежно благодаря четкой структуре постановлений. Если же предложение не содержит номера статьи, система использует семантический поиск. Для этого юридический корпус был преобразован в векторную базу данных с помощью библиотек FAISS² и LangChain³ с использованием модели sentence BERT⁴, что позволяет выполнять семантический поиск по косинусному сходству эмбедингов предложений.

Однако модель эмбедингов, используемая в предыдущей версии системы, не обучалась специально на юридических текстах, что приводило к суще-

¹ <https://pypi.org/project/razdel/>

² <https://github.com/facebookresearch/faiss>

³ <https://www.langchain.com/>

⁴ https://huggingface.co/ai-forever/sbert_large_nlu_ru

ственной потере фрагментов, извлекаемых на этом этапе (до 30%), и делало данный компонент наиболее слабым этапом текстовой обработки, создавая заведомые ограничения для полноты извлекаемых противоречий.

Наконец, извлеченные пары предложений разбиваются на батчи и обрабатываются LLM с адаптером LoRA, специально настроенным для обсуждаемой задачи.

Рассмотрим далее подробно все этапы: подготовку данных, способы повышения качества RAG как наиболее проблемного компонента первоначальной реализации, промпт, а также выбор и обучение моделей.

ПОДГОТОВКА ДАННЫХ

Система включает следующие датасеты: репозиторий релевантных правовых норм, обучающий корпус, валидационную выборку для анализа качества работы моделей по детекции противоречий, тестовый набор документов, представляющих собой полные судебные постановления для оценки качества RAG и всего пайплайна, а также автотесты, сформированные на их основе.

Репозиторий правовых норм

Репозиторий правовых норм, с которыми соотносятся судебные постановления, включает Кодекс Российской Федерации об административных правонарушениях (КоАП РФ), Кодекс административного судопроизводства РФ (КАС РФ), Гражданский кодекс РФ (ГК РФ), Уголовно-процессуальный кодекс РФ (УПК РФ), Гражданский процессуальный кодекс РФ (ГПК РФ), Уголовно-исполнительный кодекс РФ (УИК РФ), Уголовный кодекс РФ (УК РФ), а также инструкции по судебному делопроизводству.

На этапе предобработки из документов удаляются комментарии издателя и технические примечания (такие как «в редакции...»).

В предыдущей версии программы тексты преобразовывались далее в унифицированную структуру вида «Номер статьи. Текст статьи» с разделением статей на части, однако, поскольку постановления часто содержат ссылки не только на статьи и части статей, но и на пункты частей статей (например, «пункт 1 части 2 статьи 23.3»), мы добавили в базу кодексов также разделение по пунктам частей статей, что способствует повышению точности обработки. С этой целью были переработаны предобработанная база кодексов и векторная база.

При наличии в статье пунктов общее содержание излагается в части статьи, являющейся общей смысловой преамбулой, в то время как в пунктах описываются дальнейшие варианты действий. Это означает, что если пункт будет выступать отдельным фрагментом, смысл его будет утерян. В этой связи при каждом пункте хранится также и содержание части, к которой он относится.

Обучающие данные

Поскольку мы рассматриваем решаемую задачу как задачу текстового следования (Textual Entailment), обучающий набор данных был организован как коллекция пар предложений, где первый элемент (предпосылка) – это фрагмент юридической статьи, а второй (гипотеза) – фрагмент судебного постановления. Отношение между предложениями в каждой паре размечено вручную как “entailment” (следствие – одно предложение следует из другого), “contradiction” (противоречие – одно предложение противоречит другому) или “not mentioned” (одно предложение не является ни следствием, ни противоречием другого). Примеры обучающих данных представлены в [1].

Для задачи выявления противоречий в выделении класса entailment нет непосредственной необходимости, достаточно было ограничиться бинарной классификацией вида «противоречие» vs «отсутствие противоречий», где entailment и not mentioned объединены в один класс. Однако мы сохранили данные с тремя классами, чтобы иметь возможность преобразовывать их также в тип «похожие/непохожие» для дальнейшего дообучения модели эмбедингов предложений.

Процесс создания датасета

В первую очередь были определены наиболее типичные виды противоречий, встречающиеся в рассматриваемых документах. К таковым относятся:

- неверное указание суммы штрафа цифрами и/или прописью;
- отсутствие суммы штрафа в постановлении;
- назначение дополнительного наказания, не предусмотренного регулирующей статьей;
- неверные сроки обжалования постановления или иные процессуальные сроки;

- антонимы («включая случаи» vs «за исключением случаев»);
- неверное назначение места/режима содержания (меры обеспечения исполнения выдворения);
- отрицание («может быть» vs «не может быть»);
- неверные реквизиты закона (номер/дата).

На текущем этапе модель обучена выявлять наиболее характерные противоречия и не нацелена на библиографические/цитатные ошибки, такие как неверный номер статьи/пункта/подпункта или неверная ссылка на правовой кодекс.

На основе этих данных был сгенерирован обучающий датасет, состоящий из пар предложений, где примеры противоречий охватывают вышеупомянутые типы: штрафы, выходящие за допустимый диапазон, неверные сроки, отсутствующие или лишние санкции и т. д. Датасет также включает пары предложений с entailment (где, например, суммы штрафов находятся в пределах законодательно допустимых диапазонов) и not mentioned.

Для защиты персональных данных исходные судебные постановления были анонимизированы путем замены личных идентификаторов на плейсхолдеры, при этом структура юридических документов была сохранена.

Исходный корпус включал около 500 пар, где гипотезам постановлений были вручную сопоставлены релевантные предпосылки кодексов. После этого набор данных был расширен с помощью GPT-5.2: для каждого предложения из постановления мы генерировали пары entailment, contradiction и not mentioned через специфичные для класса промпты и проверяли все сгенерированные примеры. Промпты были разработаны так, чтобы возвращать таблицу, где каждая строка содержала (i) норму кодекса как предпосылку, (ii) фрагмент судебного постановления как гипотезу и (iii) идентификатор цитируемой статьи/части.

Размер датасета и баланс классов

Полученный датасет был разделен на обучающую и валидационную выборки, содержащие описанные выше пары «предпосылка – гипотеза». Следуя предыдущим работам в области доменно-независимого NLI, мы стремились сохранить распределение трех классов сбалансированным, чтобы уменьшить эф-

фект перекося классов, который, как показано, влияет на обобщающую способность модели (см., в частности, [26]). Эксперименты по оптимизации баланса классов представлены в [1], а итоговые цифры для датасетов с тернарной и бинарной классификациями – в табл. 1 (бинарная классификация включает contradiction vs entailment + not mentioned).

Табл. 1. Итоговый датасет и распределение классовых меток.

Датасет	Всего текстовых фрагментов	Contradiction	Entailment	Not mentioned	Entailment + Not mentioned
Обучающий датасет	2671	971 (36.4%)	832 (31.1%)	868 (32.5%)	1700 (63.65%)
Валидационный датасет	393	134 (34.1%)	129 (32.8%)	129 (32.8%)	258 (65.90%)

Тестовые данные и автотесты

В текущей реализации мы увеличили тестовую выборку документов с 13 до 30 постановлений. Обновленная выборка включает как документы с противоречиями, так и документы без противоречий. В сумме датасет содержит 66 пар с отношением contradiction и 43 пары с отношением entailment.

Новые постановления касаются тех же статей, что и в предыдущей выборке, однако содержат новые формулировки, отсутствующие в обучающем датасете. Анализ данных показал, что в случае близости текста предпосылки к тексту гипотезы система успешно находит и оценивает нужную пару, несмотря на отсутствие формулировки в обучающей выборке. Если же текст гипотезы существенно отличается от предпосылки, например содержит дополнительные фрагменты, то модель чаще классифицирует такие пары, как not mentioned вместо entailment.

На основе выбранных судебных постановлений были сформированы автотесты, представляющие собой excel-таблицы, где для каждой пары вида «гипотеза – предпосылка» указан тег отношения между гипотезой и предпосылкой.

В случаях, когда одному предложению гипотезы соответствуют несколько предложений из предпосылок, предложение входит в датасет более одного раза – по отдельному вхождению для каждой релевантной предпосылки.

Автотесты предназначены для оценки как полноты семантического поиска RAG, так и общего качества работы всего пайплайна. Качество работы непосредственно LLM при этом измеряется на валидационной выборке независимо от работы RAG.

ПОВЫШЕНИЕ КАЧЕСТВА RAG

Работы по оптимизации RAG-компонента включают подготовку датасета для оценки полноты извлеченных предпосылок, создание автоматических тестов для быстрой оценки результатов проводимых экспериментов и проведение экспериментов по повышению качества.

Оценка качества исходного RAG

В качестве тестового корпуса для оценки качества RAG был взят расширенный тестовый корпус для оценки всего пайплайна (см. п. «Тестовые данные и автотесты»), включающий 30 документов, на которые в сумме приходится 109 пар-противоречий и пар-следований. Автотест представляет собой excel-таблицу с данными парами. Для оценки RAG не требуется разделение пар на отношения противоречия и следствия.

Для оценки полноты поиска релевантных предпосылок использовалась метрика *micro-averaged Recall@n*. Поскольку используемая нами система является гибридной, итоговый *Recall@n* (*total Recall@n*) характеризует полноту всей системы, тогда как *Recall@n* для retrieval-компонента (*FAISS Recall@n*) рассчитывается только на подмножестве гипотез, для которых был задействован семантический поиск. При микроусреднении число корректно найденных эталонных ссылок суммируется по всем гипотезам и делится на общее число эталонных ссылок:

$$\text{Recall@n} = \frac{\sum_{i=1}^N |G_i \cap R_i^{(n)}|}{\sum_{i=1}^N |G_i|},$$

где G_i – множество эталонных релевантных нормативных ссылок для i -й гипотезы, а $R_i^{(n)}$ – множество ссылок, найденных системой среди первых n результатов.

На начальном этапе оптимизации полнота была рассчитана для исходной версии RAG, используемой в [1]. Для создания векторной базы данная версия использует уже упоминавшиеся ранее эмбединги предложений `sbert_large_nlu_ru`. Для каждого предложения-гипотезы система RAG предоставляет некоторый список предложений-предпосылок, упорядоченный по косинусной близости с гипотезой. Первые k предпосылок отдаются далее модели для детекции противоречий, этот параметр далее будем называть `top_k`. В [1] `top_k` извлекаемых посылок был равен 20.

Метрики качества исходной системы, оцененной на обновленных тестах с учетом улучшенной фильтрации, изменились и для лучшей модели составили `precision = 1`, `recall = 0.59`, `F1 = 0.74` для класса `contradiction`.

Наиболее проблемными являются случаи, когда сходные объекты имеют разные наименования в предпосылке и гипотезе. Например:

Гипотеза:

*«Исполнение постановления возложить на межрайонный отдел судебных приставов по административному выдворению **ГУФССП России** по г. Москве».*

Предпосылка:

«Постановление об административном выдворении за пределы Российской Федерации иностранных граждан или лиц без гражданства исполняется:

*2) **федеральным органом исполнительной власти** <...>».*

«ГУФССП России по г. Москве» из гипотезы соответствует «федеральному органу исполнительной власти» из предпосылки, однако RAG-компонент подобные пары не находит.

Другой контекст, вызывающий сложности, – это случаи, когда релевантные статьи указываются в мотивационной и отсутствуют в резолютивной части постановления. Например, статьи 3.10 часть 5 и 27.19 КоАП в большинстве случаев указываются только в мотивационной части, поскольку сами по себе не являются статьями обвинения в нарушении, а лишь процессуально обосновывают принудительное содержание во ЦВСИГ.

Наконец, интерес представляют случаи соотношения одной гипотезы с несколькими предпосылками. Практика показывает, что одно предложение постановления может регулироваться несколькими нормами, причем текст может содержать отсылки не на все регулирующие статьи.

В качестве примера приведем следующее постановление:

«До исполнения административного наказания в виде принудительного выдворения за пределы Российской Федерации, содержать гражданина Рахимова в Центре временного содержания иностранных граждан и лиц без гражданства ГУ МВД России по городу Москве на основании ст. 27.19.1 ч 1 Кодекса РФ об АП сроком, не превышающем 100 суток <...>»

Верным было бы указать в данном случае три предпосылки: статья 3.10 Часть 5 (право поместить задержанного в ЦВСИГ до выдворения), статья 27.19 Часть 1 (регулирует помещение задержанных в ЦВСИГ до выдворения), статья 27.19.1 КоАП РФ (регулирует срок содержания гражданина в специальном учреждении).

Базовый вариант RAG извлекает в таких случаях не все статьи, а статью без указания номера при указании в тексте других статей не извлекает вообще, так как при текущем описании семантический поиск применяется только при отсутствии эксплицитно указанных в тексте номеров статей. Подобные примеры немногочисленны, однако их наличие свидетельствует в пользу применения семантического поиска не только в случаях отсутствия ссылок на статьи.

Оптимизация исходного RAG

Для повышения качества извлечения семантически релевантных статей были проведены следующие эксперименты:

- (1) дообучение эмбеддингов `sbert_large_nlu_ru` на нашем обучающем датасете;
- (2) добавление модели-ранжировщика с дообучением и без;
- (3) комбинирование вариаций «базовые/дообученные эмбеддинги `sbert`» и «базовый/дообученный ранжировщик» с разными параметрами `candidate_top_k` (количество кандидатов предпосылок, которые предоставляет модель до применения ранжировщика) и `final_top_k` (количество предпосылок, которые предоставляет ранжировщик);

(4) замена эмбеддингов предложений на эмбеддинги моделей bge-m3⁵, multilingual-e5-large-instruct⁶ и Qwen3-Embedding-0.6B⁷.

Первая серия экспериментов состояла в комбинированном дообучении текущих эмбеддингов предложений (назовем их базовыми) и использовании предобученного либо дообученного на наших данных ранжировщика⁸. В этих вариантах использовались параметры 20:10 и 40:20⁹. Следует отметить, что для правилочной части извлечения предпосылок полнота не изменялась и всегда составляла 0.93. Для обеих версий параметров лучше всех показала себя версия с дообученным ранжировщиком (см. табл. 2).

Табл. 2. Дообучение текущих эмбеддингов предложений и использование ранжировщика.

Эмбеддинги	Ранжировщик	Candidate_top_k	Final_top_k	RAG Recall@n	Total Recall@n
Базовые	отсутствует	20	10	0.55	0.71
		40	20	0.58	0.72
	предобученный	20	10	0.45	0.64
		40	20	0.49	0.67
	дообученный	20	10	0.58	0.72
		40	20	0.63	0.75
Дообученные	отсутствует	20	10	0.63	0.75
		40	20	0.63	0.75
	предобученный	20	10	0.62	0.74
		40	20	0.62	0.74
	дообученный	20	10	0.63	0.75
		40	20	0.63	0.75

Сразу несколько вариантов дали одинаковое качество, поэтому для дальнейших экспериментов был выбран вариант с базовыми эмбеддингами и дообученным ранжировщиком для последующих экспериментов с заменой модели эмбеддингов.

⁵ <https://huggingface.co/BAAI/bge-m3>

⁶ <https://huggingface.co/intfloat/multilingual-e5-large-instruct>

⁷ <https://huggingface.co/Qwen/Qwen3-Embedding-0.6B>

⁸ <https://huggingface.co/Alibaba-NLP/gte-multilingual-reranker-base>

⁹ Здесь и далее указываются параметры в виде (candidate_top_k:final_top_k).

Вторая серия экспериментов состояла в исследовании влияния параметров candidate_top_k и final_top_k. Результаты представлены в табл. 3.

Табл. 3. Влияние параметров candidate_top_k и final_top_k.

Вариант	Candidate_top_k	Final_top_k	RAG Recall@n	Total Recall@n
Базовые эмбединги + дообученный ранжировщик	20	10	0.58	0.72
	40	20	0.63	0.75
	60	40	0.71	0.8
	80	60	0.75	0.83
	100	10	0.74	0.82
	100	20	0.74	0.82
	100	40	0.74	0.82
	100	60	0.77	0.83
	100	80	0.77	0.83

Как видно из табл. 3, общая полнота получается наилучшей (и одинаковой) для версий 100:60 и 100:80 (поскольку полнота становится неизменной, очевидно, что увеличивать далее final_top_k нецелесообразно).

В третьей серии экспериментов мы взяли параметры 100:60, учитывая полученные во второй серии результаты, и попробовали заменить модель эмбедингов предложений. Результаты представлены в табл. 4. При этом модель sbert_large_nlu_ru – классический трансформер, базирующийся на архитектуре BERT, так же, как и bge-m3; e5, представляет собой архитектуру, состоящую только из энкодеров, но принимает на вход также промпт, влияющий на итоговый эмбединг предложения, а Qwen3-Embedding – модель, построенная на базе генеративной Qwen. Соответственно, последние две модели при генерации эмбедингов предложений получают короткий промпт вида «Instruct: Given a Russian legal statement from the operative section of a court decision, retrieve the relevant provision of Russian law. Query:».

Табл. 4. Замена модели эмбедингов предложений.

Модель	RAG Recall@n	Total Recall@n
sbert_large_nlu_ru	0.77	0.83
bge-m3	0.86	0.89
Qwen3-Embedding-0.6B	0.88	0.9
multilingual-e5-large-instruct	0.57	0.72

Лучший результат полноты RAG-модуля показала версия с эмбедингами Qwen3-Embedding-0.6B.

В результате выполненных экспериментов полнота RAG-компонента системы увеличилась на 18%, что достигается в версии RAG-компонента с эмбедингами QWEN и ранжировщиком 100:60 (табл. 5).

Табл. 5. Итоговые показатели RAG по результатам экспериментов.

Вариант системы	Candidate_top_k	Final_top_k	RAG Recall@n	Правильный Recall@n	Итоговый Recall@n
Бейзлайн RAG	20	20	0.58	0.93	0.72
RAG-Qwen 100:60	100	60	0.88	0.93	0.90
RAG-Qwen 40:20	40	20	0.82	0.93	0.86

РАЗРАБОТКА ПРОМПТОВ

При работе с генеративными моделями важным аспектом является создание промптов. Влияние промптов на модель не менее значимо, чем обучающий набор данных, и даже небольшие изменения в промпте могут оказать существенное влияние на результаты. Промпты, используемые в рассматриваемой системе, а также описания экспериментов по их оптимизации представлены в [1].

ВЫБОР И ДООБУЧЕНИЕ МОДЕЛЕЙ

Вследствие конфиденциальности обрабатываемых документов выбор модели осуществлялся между локальными (on-premise) моделями, соответствующими нашим ресурсным возможностям, вместо облачных версий.

Были выбраны следующие модели с версиями в 8 млрд параметров, потребляющие приблизительно 22 ГБ видеопамати GPU при дообучении LoRA и

15 ГБ при инференсе: LLAMA 3.1-8B, MINISTRAL-8B-INSTRUCT, QWEN3-8B и T-LITE-IT-2.1. О мотивации выбора можно посмотреть [1, разд. 5.1].

Все модели были протестированы в тернарной и бинарной постановках задачи в режиме zero-shot. Мы рассматриваем Llama 3.1-8B как базовую модель (baseline), поскольку это самая простая модель без явной поддержки русского языка. Затем модели дообучили с помощью адаптера LoRA, используя библиотеку trl, в течение трех эпох с размером батча 2, накоплением градиента 8 и скоростью обучения 2×10^{-4} . Гиперпараметры были намеренно выбраны одинаковыми для всех моделей. Качество моделей оценивалось промежуточно на валидационном наборе, где лучшие метрики оказались у бинарной версии модели T-Lite (F1 = 0.9982), однако метрики на тестовых данных представляют более значительный интерес, поскольку тестовые данные серьезно отличаются от обучающих по распределению классов и более близки к реальным данным.

В табл. 6 приведены метрики на исходном тестовом датасете для моделей с LoRA-адаптерами (это метрики, оценивающие непосредственно качество работы моделей, без учета RAG-компонента).

Табл. 6. Метрики моделей после дообучения LoRA для бинарной и тернарной постановок на тестовом датасете.

	Бинарный вариант						Тернарный вариант					
	contradiction			no			contradiction			not mentioned		
Модель	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Llama 3.1-8B	0.42	0.52	0.46	0.98	0.97	0.97	0.54	0.28	0.37	0.97	0.99	0.98
Ministral-8B-Instruct-2410	0.47	1	0.64	1	0.95	0.98	0.96	1	0.98	1	1	1
Qwen3-8B	0.49	0.96	0.65	1	0.96	0.98	0.52	1	0.68	1	0.96	0.98
T-Lite-it-2.1	0.43	1	0.6	1	0.94	0.97	0.69	1	0.82	1	0.98	0.99

Лучшие результаты довольно неожиданно показала модель Ministral. Вероятно, это связано с тем, что архитектура модели довольно простая, а метод обучения исходной модели максимально близок к методу дообучения адаптера.

ОЦЕНКА И ОГРАНИЧЕНИЯ

Во время инференса результирующая генеративная модель получает промпт, содержащий пару «фрагмент постановления – правовая норма», и выдает метку класса для отношения между гипотезой и предпосылкой. В зависимости от выбора классификации (бинарной или тернарной) модель либо различает contradiction/entailment/not mentioned, либо объединяет entailment и not mentioned в единый класс «no». Далее мы приводим показатели для всех классов тернарной классификации как более наглядной, однако актуальными для текущей задачи являются показатели для contradiction. Вывод результатов пользователю представлен на рис. 1.

	Предложение	Номер статьи	Текст статьи	Результат сравнения
1	Признать Алимова Орхана Илхамовича виновным в совершении административного правонарушения, предусмотренного ч.3 ст.20.20 Кодекса РФ об административных правонарушениях, и назначить ему наказание в виде административного штрафа в размере <u>двух тысяч рублей</u> с административным выдворением за пределы Российской Федерации в форме контролируемого самостоятельного выезда из Российской Федерации.	КоАП РФ: Статья 20.20. п. 3.	КоАП РФ: Статья 20.20. п. 3. Действия, указанные в части 2 настоящей статьи, совершенные иностранным гражданином или лицом без гражданства, - влекут наложение <u>административного штрафа в размере от четырех тысяч до пяти тысяч рублей</u> с административным выдворением за пределы Российской Федерации либо административный арест на срок до пятнадцати суток с административным выдворением за пределы Российской Федерации.	противоречие

Рис. 1. Вывод программы.

Для итогового измерения качества общей системы были взяты три конфигурации:

- 1) бейзлайн (исходная система с базовым RAG-компонентом, но с обновленными правилами фильтрации);
- 2) система с RAG-компонентом на основе эмбедингов Qwen3 с дообученным ранжировщиком, параметры top_k – 100:60;
- 3) система с RAG-компонентом на основе эмбедингов Qwen3 с дообученным ранжировщиком, параметры top_k – 40:20 (данная версия была взята для того, чтобы иметь возможность сравнить бейзлайн также с более близкими параметрами top_k).

Результаты представлены в табл. 7. В ней указаны включает оценки качества полного цикла работы системы для следующих параметров: вариант RAG,

тип модели (базовая vs LoRA-адаптер) и название модели. Кроме того, приведены точность (contradiction precision), полнота (contradiction recall) и F-мера (F1) для детекции противоречий, а также отличие данных значений при доработанном RAG от аналогичных значений при базовом RAG: ΔP – дельта по точности, ΔR – дельта по полноте, $\Delta F1$ – дельта по f-мере.

Табл. 7. Оценка качества работы полного цикла системы.

Тип RAG	Тип модели	Имя модели	Contradiction precision	ΔP	Contradiction recall	ΔR	Contradiction F1	$\Delta F1$
Baseline	Базовая LLM	Llama	0.05	0	0.15	0	0.07	0
		Ministral	0	0	0	0	0	0
		Qwen	0.08	0	0.86	0	0.15	0
		T-Lite	0.44	0	0.68	0	0.53	0
	LoRA	Llama	0.82	0	0.14	0	0.23	0
		Ministral	1	0	0.52	0	0.68	0
		Qwen	0.53	0	0.82	0	0.64	0
		T-Lite	0.68	0	0.83	0	0.75	0
QWEN 100:60	Базовая LLM	Llama	0.01	-0.04	0.12	-0.03	0.02	-0.05
		Ministral	0	0	0	0	0	0
		Qwen	0.03	-0.05	0.82	-0.04	0.05	-0.1
		T-Lite	0.16	-0.28	0.8	0.12	0.26	-0.27
	LoRA	Llama	0.57	-0.25	0.12	-0.02	0.2	-0.03
		Ministral	0.97	-0.03	0.56	0.04	0.71	0.03
		Qwen	0.23	-0.3	0.91	0.09	0.37	-0.27
		T-Lite	0.44	-0.24	0.94	0.11	0.6	-0.15
QWEN 40:20	Базовая LLM	Llama	0.02	-0.03	0.08	-0.07	0.03	-0.04
		Ministral	0	0	0	0	0	0

		Qwen	0.07	−0.01	0.79	−0.07	0.13	−0.02
		T-Lite	0.3	−0.14	0.79	0.11	0.43	−0.1
	LoRA	Llama	0.62	−0.2	0.12	−0.02	0.2	−0.03
		Ministral	0.97	−0.03	0.53	0.01	0.69	0.01
		Qwen	0.34	−0.19	0.86	0.04	0.49	−0.15
		T-Lite	0.57	−0.11	0.91	0.08	0.7	−0.05

Как видно из табл. 7, наиболее сильное влияние на итоговую оценку работы системы оказывает тип модели (в данном случае, дообучение LoRA-адаптера). Наилучшее качество для класса contradiction дают версии системы с LoRA-адаптером, при этом повышение параметров top_k улучшает полноту, однако может серьезно ухудшить точность.

Модель, показавшая лучшие результаты в предыдущей работе (Ministral в тернарном варианте), на обновленных тестах в бейзлайне дала P = 1.00, R = 0.52 и F1 = 0.68, проиграв при этом по F1-мере некоторым другим моделям из-за падения полноты. Версии с улучшенным RAG-компонентом несколько подняли ее полноту (до 0.56 в лучшем случае), но ухудшили точность до 0.97.

Лучшие результаты по полноте, полученные с использованием обновленных тестов, представлены в табл. 8, лучшие результаты по F1-мере – в табл. 9.

Табл. 8. Лучшие результаты по полноте полного пайплайна.

Модель	Тип оценки	Precision	Recall	F1
T-Lite	RAG-Qwen: 100:60	0.44	0.94	0.60
	RAG-Qwen: 40:20	0.57	0.91	0.70
Qwen	RAG-Qwen: 100:60	0.23	0.91	0.37

Табл. 9. Лучшие результаты по F1-мере полного пайплайна.

Модель	Тип оценки	Precision	Recall	F1
T-Lite	Бейзлайн RAG	0.68	0.83	0.75
Ministral	RAG-Qwen: 100:60	0.97	0.56	0.71
T-Lite	RAG-Qwen: 40:20	0.57	0.91	0.70

Наилучшую полноту демонстрирует вариант с моделью T-Lite, где полнота выявленных противоречий достигает 94%. Тем не менее в этом случае мы сталкиваемся с проблемой слишком большого количества кандидатов: если модель обрабатывает большее количество пар предпосылка – гипотеза, точность детекции класса contradiction заметно падает. Очевидно, что версия RAG-компонента с эмбедингами QWEN и ранжировщиком лучше извлекает релевантные предпосылки (полнота собственно RAG-компонента системы прирастает почти на 20%), однако детектор противоречий чувствителен к возникающему из-за большего количества предпосылок шуму. Несмотря на снижение точности, улучшение качества RAG важно для системы, так как с точки зрения сценария работы наиболее критичным является пропуск имеющегося в тексте противоречия, что делает полноту более значимым параметром, чем точность.

ЗАКЛЮЧЕНИЕ

В результате проведенной оптимизации качество RAG существенно возросло, что позволило повысить до 94% полноту извлекаемых системой противоречий, однако увеличение числа рассматриваемых пар снизило точность. Следующей задачей, таким образом, является снижение количества пар-кандидатов при возможном сохранении полноты RAG.

Дальнейшая работа будет развиваться также в следующих направлениях.

Мы планируем провести эксперименты с гиперпараметрами дообучения LoRA, расширить обучающий датасет для увеличения его репрезентативности и улучшения устойчивости к более редким случаям, а также расширить таксономию выявляемых ошибок, а также масштабировать систему на документы других типов.

Благодарности

Исследование поддержано проектом FMNE-2025-0002 «Проблемы прикладной лингвистики и гуманитарного знания в свете современных цифровых технологий», выполняемым в Институте языкознания РАН.

СПИСОК ЛИТЕРАТУРЫ

1. Kozlova E., Petrova M., Baiuk A. Contradiction Detection in Administrative Offense Rulings: A Case Study Analysis // Proc. of Conference on Computational Linguistics and Intellectual Technologies Dialog-2026. 2026. P. 294–310.
2. Harabagiu S., Hickl A., Lacatusu F. Negation, contrast and contradiction in text processing // Proc. of the 21st National Conference on Artificial Intelligence (AAAI-06). 2006. P. 755–762.
3. De Marneffe M.-C., Rafferty A.N., Manning C.D. Finding contradictions in text // Proc. of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-08: HLT). 2008. P. 1039–1047.
4. Mehdad Y., Negri M., Federico M. Towards cross-lingual textual entailment // Proc. of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2010). 2010. P. 321–324.
5. Bowman S., Angeli G., Potts C., Manning C.D. A large annotated corpus for learning natural language inference // arXiv:1508.05326. 2015. <https://doi.org/10.48550/arXiv.1508.05326>
6. Dragos V. Detection of contradictions by relation matching and uncertainty assessment // Procedia Computer Science. 2017. Vol. 112. P. 71–80.
7. Pielka M., et al. A linguistic investigation of machine learning based contradiction detection models: an empirical analysis and future perspectives // arXiv:2210.10434. 2022. <https://doi.org/10.48550/arXiv.2210.10434>
8. Putra I M. S., Siahaan D., Saikhu A. Recognizing textual entailment: a review of resources, approaches, applications, and challenges // ICT Express. 2024. Vol. 10, No. 1. P. 132–155. <https://doi.org/10.1016/j.icte.2023.08.012>
9. Shajalal M., et al. Textual entailment recognition with semantic features from empirical text representation // arXiv:2210.09723v4. 2022. <https://doi.org/10.48550/arXiv.2210.09723>

10. *Rahimi Z., ShamsFard M.* A knowledge-based approach for recognizing textual entailments with a focus on causality and contradiction // Research Square preprint. 2024. <https://doi.org/10.21203/rs.3.rs-3826973/v1>
11. *Asif M., Ahmed Khan T., Song W.-C.* Evaluating large language models for optimized intent translation and contradiction detection using KNN in IBN // IEEE Access. 2025. Vol. 13. P. 20316–20327.
12. *Wu X., Niu X., Rahman R.* Topological analysis of contradictions in text // Proc. of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2022). 2022. P. 2478–2483. <https://doi.org/10.1145/3477495.3531881>
13. *Williams A., Nangia N., Bowman S.* A broad-coverage challenge corpus for sentence understanding through inference // Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018). 2018. P. 1112–1122. <https://doi.org/10.18653/v1/N18-1101>
14. *Camarda A.D., Ianni G.* A study on contradiction detection using a neuro-symbolic approach // Proc. of 40th Italian Conference on Computational Logic (CILC 2025). 2025, paper 08.
15. *Feng X., Hunter A.* Identification of entailment and contradiction relations between natural language sentences: a neurosymbolic approach // arXiv:2405.01259. 2024. <https://doi.org/10.48550/arXiv.2405.01259>
16. *Walton D., Reed C.* Argumentation schemes and defeasible inferences // Working Notes of the ECAI 2002 Workshop on Computational Models of Natural Argument (CMNA 2002). 2002. P. 45–55.
17. *Ibraim S.S., Mendonça P.C.C.* Book Review: «Argumentation Schemes» // Ensaio Pesquisa em Educação em Ciências. 2013. Vol. 15, No. 3. P. 255–262.
18. *Walton D.* Legal argumentation and evidence // Penn State University Press, 2010. 392 p.
19. *Mantravadi A., et al.* LegalWiz: a multi-agent generation framework for contradiction detection in legal documents // arXiv:2510.03418. 2025. <https://doi.org/10.48550/arXiv.2510.03418>

20. Koreeda Y., Manning C.D. ContractNLI: a dataset for document-level natural language inference for contracts // Findings of the Association for Computational Linguistics: EMNLP 2021. 2021, P. 1907–1919.
<https://doi.org/10.18653/v1/2021.findings-emnlp.164>
21. Nguyen H.-T., et al. Transformer-based approaches for legal text processing: JNLP team – COLIEE 2021 // The Review of Socionetwork Strategies. 2022. Vol. 16, No. 1. P. 135–155. <https://doi.org/10.1007/s12626-022-00102-2>
22. Nguyen C., et al. CAPTAIN at COLIEE 2023: efficient methods for legal information retrieval and entailment tasks // arXiv:2401.03551. 2024.
<https://doi.org/10.48550/arXiv.2401.03551>
23. Aires J.P., Strube de Lima V.L., Meneguzzi F. Identifying potential conflicts between norms in contracts // 18th International Workshop on Coordination, Organizations, Institutions, and Norms (COIN 2015). 2016. P. 179–188.
24. Surana S., Dembla S., Bihani P. Identifying contradictions in the legal proceedings using natural language models // SN Computer Science. 2022. Vol. 3, No. 3. Art. no. 187. <https://doi.org/10.1007/s42979-022-01075-3>
25. Ariai F., Mackenzie J., Demartini G. Natural language processing for the legal domain: a survey of tasks, datasets, models, and challenges // ACM Computing Surveys. 2025. Vol. 58, No. 6. P. 1–37. <https://doi.org/10.48550/arXiv.2410.21306>
26. Hosseini M.J., Petrov A., Fabrikant A., Louis A. A synthetic data approach for domain generalization of NLI models // Proc. of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024). 2024. P. 2212–2226.
<https://doi.org/10.18653/v1/2024.acl-long.120>

ON IMPROVING RAG QUALITY FOR CONTRADICTION DETECTION IN COURT RULINGS

E. A. Kozlova¹ [0009-0002-5884-126X], A. M. Baiuk² [0009-0002-6721-0005],

A. M. Petrova³ [0009-0006-0393-1400]

^{1,3}*DM Solutions, Moscow, Russia*

^{2,3}*Institute of Linguistics of the Russian Academy of Sciences, Moscow, Russia*

¹e.kozlova@dm-solutions.ru, ²alexandra.m.baiuk@gmail.com,

³m.petrova@dm-solutions.ru

Abstract

The paper extends and continues the work on a pilot system for detecting contradictions in court rulings. The tool aims to identify contradictions between court rulings on administrative offense cases and the legislative norms governing them. The previously presented pipeline, which includes rule-based document preprocessing, search over a structured repository of legal norms, and LLM fine-tuning using a LoRA adapter, demonstrated fairly high metrics [1].

However, this implementation had a number of limitations, primarily related to the RAG component. Unlike LLM, the RAG module was not trained on legal texts, which led to a substantial loss of relevant fragments for comparison at the retrieval stage.

The present work therefore focuses on a detailed quality analysis and improving the RAG component of the system. A dataset for evaluating premise retrieval recall was compiled, comprising 30 documents, along with automated tests containing all relevant sentence pairs (more than one hundred pairs in total). Using this data, the recall@n was calculated for the original (baseline) version of the model.

To improve the recall of the retrieved data, the following experiments were conducted: (1) fine-tuning the original embeddings on the system's training dataset; (2) adding a reranking model, both with and without fine-tuning; (3) combining variations of base/fine-tuned embeddings/reranker across several values of the top_k parameter (the number of retrieved matches); and (4) replacing the sentence embedding model with alternative options.

These experiments improved RAG recall from 72% to 90%. However, increasing the number of the retrieved fragments reduced the system's overall precision metrics, indicating a need for further training.

Other notable optimizations include reorganizing the structure of the legal codes database, expanding the test set, and improving the algorithm for filtering out irrelevant sentences from the rulings.

Keywords: *contradiction detection, legal domain, LoRA (Low-Rank Adaptation) fine-tuning, RTE (Recognizing Textual Entailment), RAG (Retrieval-Augmented Generation).*

Acknowledgments

This research was supported by project FMNE-2025-0002 «Problems of Applied Linguistics and the Humanities in the Light of Modern Digital Technologies», carried out at the Institute of Linguistics, Russian Academy of Sciences.

REFERENCES

1. Kozlova E., Petrova M., Baiuk A. Contradiction Detection in Administrative Offense Rulings: A Case Study Analysis // Proc. of Conference on Computational Linguistics and Intellectual Technologies Dialog-2026. 2026. P. 294–310.
2. Harabagiu S., Hickl A., Lacatusu F. Negation, contrast and contradiction in text processing // Proc. of the 21st National Conference on Artificial Intelligence (AAAI-06). 2006. P. 755–762.
3. De Marneffe M.-C., Rafferty A.N., Manning C.D. Finding contradictions in text // Proc. of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-08: HLT). 2008. P. 1039–1047.
4. Mehdad Y., Negri M., Federico M. Towards cross-lingual textual entailment // Proc. of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2010). 2010. P. 321–324.
5. Bowman S., Angeli G., Potts C., Manning C.D. A large annotated corpus for learning natural language inference // arXiv:1508.05326. 2015.
<https://doi.org/10.48550/arXiv.1508.05326>

6. *Dragos V.* Detection of contradictions by relation matching and uncertainty assessment // *Procedia Computer Science*. 2017. Vol. 112. P. 71–80.
7. *Pielka M., et al.* A linguistic investigation of machine learning based contradiction detection models: an empirical analysis and future perspectives // *arXiv:2210.10434*. 2022. <https://doi.org/10.48550/arXiv.2210.10434>
8. *Putra I M. S., Siahaan D., Saikhu A.* Recognizing textual entailment: a review of resources, approaches, applications, and challenges // *ICT Express*. 2024. Vol. 10, No. 1. P. 132–155. <https://doi.org/10.1016/j.icte.2023.08.012>
9. *Shajalal M., et al.* Textual entailment recognition with semantic features from empirical text representation // *arXiv:2210.09723v4*. 2022. <https://doi.org/10.48550/arXiv.2210.09723>
10. *Rahimi Z., ShamsFard M.* A knowledge-based approach for recognizing textual entailments with a focus on causality and contradiction // *Research Square preprint*. 2024. <https://doi.org/10.21203/rs.3.rs-3826973/v1>
11. *Asif M., Ahmed Khan T., Song W.-C.* Evaluating large language models for optimized intent translation and contradiction detection using KNN in IBN // *IEEE Access*. 2025. Vol. 13. P. 20316–20327.
12. *Wu X., Niu X., Rahman R.* Topological analysis of contradictions in text // *Proc. of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2022)*. 2022. P. 2478–2483. <https://doi.org/10.1145/3477495.3531881>
13. *Williams A., Nangia N., Bowman S.* A broad-coverage challenge corpus for sentence understanding through inference // *Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018)*. 2018. P. 1112–1122. <https://doi.org/10.18653/v1/N18-1101>
14. *Camarda A.D., Ianni G.* A study on contradiction detection using a neuro-symbolic approach // *Proc. of 40th Italian Conference on Computational Logic (CILC 2025)*. 2025, paper 08.
15. *Feng X., Hunter A.* Identification of entailment and contradiction relations between natural language sentences: a neurosymbolic approach // *arXiv:2405.01259*. 2024. <https://doi.org/10.48550/arXiv.2405.01259>

16. Walton D., Reed C. Argumentation schemes and defeasible inferences // Working Notes of the ECAI 2002 Workshop on Computational Models of Natural Argument (CMNA 2002). 2002. P. 45–55.
17. Ibraim S.S., Mendonça P.C.C. Book Review: «Argumentation Schemes» // Ensaio Pesquisa em Educação em Ciências. 2013. Vol. 15, No. 3. P. 255–262.
18. Walton D. Legal argumentation and evidence // Penn State University Press, 2010. 392 p.
19. Mantravadi A., et al. LegalWiz: a multi-agent generation framework for contradiction detection in legal documents // arXiv:2510.03418. 2025.
<https://doi.org/10.48550/arXiv.2510.03418>
20. Koreeda Y., Manning C.D. ContractNLI: a dataset for document-level natural language inference for contracts // Findings of the Association for Computational Linguistics: EMNLP 2021. 2021, P. 1907–1919.
<https://doi.org/10.18653/v1/2021.findings-emnlp.164>
21. Nguyen H.-T., et al. Transformer-based approaches for legal text processing: JNLP team – COLIEE 2021 // The Review of Socionetwork Strategies. 2022. Vol. 16, No. 1. P. 135–155. <https://doi.org/10.1007/s12626-022-00102-2>
22. Nguyen C., et al. CAPTAIN at COLIEE 2023: efficient methods for legal information retrieval and entailment tasks // arXiv:2401.03551. 2024.
<https://doi.org/10.48550/arXiv.2401.03551>
23. Aires J.P., Strube de Lima V.L., Meneguzzi F. Identifying potential conflicts between norms in contracts // 18th International Workshop on Coordination, Organizations, Institutions, and Norms (COIN 2015). 2016. P. 179–188.
24. Surana S., Dembla S., Bihani P. Identifying contradictions in the legal proceedings using natural language models // SN Computer Science. 2022. Vol. 3, No. 3. Art. no. 187. <https://doi.org/10.1007/s42979-022-01075-3>
25. Ariai F., Mackenzie J., Demartini G. Natural language processing for the legal domain: a survey of tasks, datasets, models, and challenges // ACM Computing Surveys. 2025. Vol. 58, No. 6. P. 1–37. <https://doi.org/10.48550/arXiv.2410.21306>
26. Hosseini M.J., Petrov A., Fabrikant A., Louis A. A synthetic data approach for domain generalization of NLI models // Proc. of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024). 2024. P. 2212–2226.
<https://doi.org/10.18653/v1/2024.acl-long.120>

СВЕДЕНИЯ ОБ АВТОРАХ



КОЗЛОВА Елена Алексеевна – текстовый аналитик в ООО «Датамайнинг Солюшинс» (DM Solutions, Москва). Область научных интересов: автоматизация юридической деятельности (LegalTech), обработка естественного языка (NLP), большие языковые модели (LLM), логический вывод на естественном языке (NLI), синтетические датасеты для обучения моделей.

Elena Alekseevna KOZLOVA – a text analyst at DM Solutions LLC (Moscow, Russia). Research interests: legal process automation (LegalTech), natural language processing (NLP), large language models (LLMs), natural language inference (NLI), and synthetic datasets for model training.

email: e.kozlova@dm-solutions.ru

ORCID: 0009-0002-5884-126X



БАЮК Александра Михайловна – младший научный сотрудник Института языкознания Российской академии наук. Научные интересы включают машинное обучение, большие языковые модели, извлечение информации, лингвистическую разметку, мультязычность и автоматическую обработку малоресурсных языков.

Alexandra Mikhailovna BAIUK – junior researcher at the Institute of Linguistics of the Russian Academy of Sciences. Her research interests include machine learning, large language models, information extraction, linguistic annotation, multilingualism, and natural language processing for low-resource languages.

email: alexandra.m.baiuk@gmail.com

ORCID: 0009-0002-6721-0005



ПЕТРОВА Мария Андреевна – кандидат филологических наук, заведующая отделом прикладной лингвистики Института языкознания Российской академии наук, руководитель направления текстовой аналитики в ООО «Датамайнинг Солюшинс» (DM Solutions, Москва). Сфера научных интересов – компьютерная лингвистика, формальные описания, языковые модели, семантика, текстовая аналитика, лингвистическая экспертиза.

Maria Andreevna PETROVA – Candidate of Philological Sciences, head of the Applied Linguistics Department at the Institute of Linguistics of the Russian Academy of Sciences, head of Text Analytics at DM Solutions LLC (Moscow, Russia). Research interests include computational linguistics, formal descriptions, language models, semantics, text analytics, and linguistic expertise.

email: m.petrova@dm-solutions.ru

ORCID: 0009-0006-0393-1400

Материал поступил в редакцию 25 августа 2026 года