

УДК 004.04

ВЛИЯНИЕ СЕМАНТИЧЕСКОЙ ОБРАТНОЙ СВЯЗИ НА КАЧЕСТВО ВЫРАВНИВАНИЯ МОДЕЛЕЙ, СГЕНЕРИРОВАННЫХ БОЛЬШОЙ ЯЗЫКОВОЙ МОДЕЛЮ: МАСШТАБИРОВАННАЯ ОЦЕНКА НА ПРИМЕРЕ ДОКУМЕНТОВ СЕРИИ «ЗАПРОС КОММЕНТАРИЕВ – ИНФОРМАЦИОННЫЕ ДОКУМЕНТЫ ИНТЕРНЕТ»

Проказин М. Р., МИРЭА – Российский технологический университет, 119454, Россия, Москва, пр-т Вернадского, д. 78, e-mail: mprokazin@outlook

Теремов И. А., МИРЭА – Российский технологический университет, 119454, Россия, Москва, пр-т Вернадского, д. 78, e-mail: ivan.teremow2011@yandex.ru

В работе исследовано влияние уточняющих инструкций и семантической обратной связи на качество выравнивая программных моделей объектов стандартизации, сгенерированных большими языковыми моделями. Проблема состоит в том, что независимо сгенерированные модели могут быть синтаксически корректными и содержательно близкими, но плохо сопоставимыми из-за различий в именовании сущностей, ролей, состояний, связей и единиц измерения. В отличие от предварительной проверки на малом числе документов, настоящее исследование выполнено на корпусе из 32 документов, шести генераторных моделей и пяти независимых итераций для каждой комбинации документа, модели и сценария. Всего обработано 2880 моделей, для из которых образовано 14400 пар и выполнено 85400 операций выравнивания с использованием двух СОМА-подобных методов: valentine и bdikit. В результате показано, что наличие общих инструкций по семантическому именованию и структурированию кода повышает относительное качество выравнивания на 46.2%, а семантическая обратная связь ухудшает результат, повышая качество лишь на 40.3% относительно контрольного. Данный результат важно учитывать при разработке инструментов генерации формальных моделей, статического анализа, межмодельного выравнивания и подготовки промежуточных представлений для анализа технических и нормативных документов.

Ключевые слова: большие языковые модели; формальные модели; C#; межмодельное выравнивание; schema matching; СОМА; семантическая обратная связь; prompting; RFC.

IMPACT OF SEMANTIC FEEDBACK ON THE ALIGNMENT QUALITY OF LLM-GENERATED MODELS: A SCALED EVALUATION USING DOCUMENTS OF RFC SERIES

Prokazin M. R., MIREA — Russian Technological University, 119454, Russia, Moscow, Vernadscogo avenue, 78, e-mail: mprokazin@outlook.com

Teremov I. A., MIREA — Russian Technological University, 119454, Russia, Moscow, Vernadscogo avenue, 78, e-mail: ivan.teremow2011@yandex.ru

The paper investigates the influence of clarifying instructions and semantic feedback on the quality of alignment between program models of standardization objects generated by large language models. The problem is that independently generated models may be syntactically correct and semantically close, yet difficult to align because of differences in the naming of entities, roles, states, relationships, and units of measurement. Unlike the preliminary validation on a small number of documents, the present study was conducted on a corpus of 32 RFC documents, six generative models, and five independent iterations for each combination of document, model, and scenario. In total, 2,880 models were processed; 14,400 pairs were formed from them, and 85,400 alignment operations were performed using two COMA-like methods: valentine and bdikit. The results show that general instructions on semantic naming and code structuring improve the relative alignment quality by 46.2%, whereas semantic feedback worsens the result, improving quality by only 40.3% relative to the control condition. This finding should be taken into account when developing tools for formal model generation, static analysis, model alignment, and the preparation of intermediate representations for the analysis of technical and regulatory documents.

Keywords: large language models; formal models; C#; model alignment; schema matching; СОМА; semantic feedback; prompting; RFC.

Введение

Автоматический анализ технических и нормативных документов требует перехода от естественного языка к формальному промежуточному представлению. Такое представление может использоваться для поиска противоречий, проверки совместимости требований, построения трассируемости и последующей логической обработки. В задачах нормативного анализа этот переход особенно важен, поскольку противоречие между требованиями не всегда выражено поверхностно: оно может возникать из сочетания обязанностей, запретов, условий, исключений и временных ограничений [1, 2].

Однако формальная корректность отдельной модели не гарантирует её сопоставимости с другими моделями. При независимом переводе одного и того же документа в формальный язык с разными авторами или разными генеративными моделями могут выбирать разные имена для одних и тех же сущностей, использовать разные уровни детализации, по-разному выражать роли, состояния, единицы измерения и связи. Поэтому две модели могут быть синтаксически корректными по спецификации языка [3], но плохо сопоставляться автоматическими алгоритмами выравнивания.

При анализе больших массивов документов эта проблема становится значимой. Сопоставление формальных моделей нельзя выполнять вручную попарно: требуется массовое межмодельное выравнивание, то есть автоматическое нахождение соответствий между типами, свойствами, отношениями и другими элементами моделей [4, 5]. В смежных инженерных задачах стандарты обмена требованиями, например ReqIF, решают задачу переноса артефактов между инструментами, но не устраняют проблему семантической несогласованности моделей [6, 7].

В настоящей работе рассматривается не сам логический вывод и не экспертная проверка истинности модели относительно исходного текста, а более узкая задача: как сценарий инструктирования генеративных моделей, используемых для получения формального представления из текста нормативного документа, влияет на качество представления с точки зрения межмодельного выравнивания.

Ранее уже выполнялась предварительная оценка эффективности LSP-подобной семантической обратной связи [8]. Выполнялся анализ сгенерированного кода не только на уровне синтаксиса, но и на уровне именования, явности ролей, единиц измерения и структурной согласованности. Аналогия с Language Server Protocol связана с тем, что языковой сервер может возвращать диагностики, предупреждения и рекомендации для редактора или IDE [9], что позволяет предоставить семантическую обратную связь как человеку-автору, так и LLM. В данной работе, как и в предварительной, LSP-сервер не реализуется; используется LSP-подобная процедура, имитирующая слой семантических диагностик.

Связанные работы и контекст исследования. Формализация требований и нормативных описаний обычно предполагает выделение объектов предметной области, условий, ролей, состояний и ограничений. В прикладных сценариях такая формализация нужна не сама по себе, а как промежуточный слой для последующей обработки: проверки согласованности, поиска конфликтов, построения зависимостей и генерации исполнимых или проверяемых артефактов. В задачах нормативного анализа близкую роль играют деонтические логики, позволяющие описывать обязанности, разрешения, запреты и исключения [2].

В настоящей работе C# используется как структурированный промежуточный язык выражения модели некоторого объекта, к которому предъявляются требования. C#-модель может содержать типы, свойства, перечисления и отношения, которые затем извлекаются как элементы схемы. Такой выбор удобен по двум причинам. Во-первых, синтаксис C# достаточно строг и может быть проверен стандартными средствами. Во-вторых, имена типов, свойств и перечислений сохраняют богатую предметную информацию, важную для последующего выравнивания.

Сопоставление схем и межмодельное выравнивание направлено на поиск соответствий между элементами двух схем: таблицами, атрибутами, типами, свойствами или отношениями. Классический обзор подходов к автоматическому сопоставлению схем представлен в работе [4]. Близкая тематика исследований в области сопоставления онтологий рассматривает сопоставление классов, свойств и экземпляров онтологий [5].

Для данной работы важны методы, учитывающие не только локальное сходство имён, но и структуру модели. Так, алгоритмы COMA/COMA++ относятся к классическим системам комбинированного сопоставления схем и онтологий [10]. Другие известные подходы, такие как Similarity Flooding и Cupid, также используют сочетание лексических и структурных признаков [11, 12]. В настоящем эксперименте используются две COMA-подобные реализации: valentine:coma_py из Valentine [13] и bdikit:coma из BDI-Kit [14], показавшие повышенную эффективность в предварительном эксперименте [8].

Отдельно развивается направление исследований, где большие языковые модели участвуют в алгоритме сопоставления схем для повышения качества [15]. Настоящая работа отличается тем, что LLM применяются не

для сопоставления, а для генерации формальных моделей, которые затем сопоставляются классическими алгоритмами сопоставления.

Итеративное уточнение ответа (Self-Refine) модели по обратной связи является распространённой схемой в LLM-системах. Self-Refine и близкие подходы показывают, что модель или внешний рецензент может анализировать промежуточный результат и затем улучшать его в следующем шаге [16]. Для генерации формальных моделей такая схема выглядит естественно: сначала модель создаёт код, затем анализатор или другая модель формирует замечания, после чего исходная модель выполняет ревизию кода с учётом замечаний.

Однако улучшение результата с точки зрения человека не обязательно положительно влияет на качество межмодельного выравнивания. Удобные для восприятия длинные идентификаторы, уточнённые роли, явные единицы измерения могут быть не приспособлены для автоматического сопоставления, если разные генеративные модели выбирают разные уточняющие признаки и тем самым снижают поверхностное сходство имён.

Постановка задачи и исследовательские вопросы. Пусть исходный документ d преобразуется генераторной моделью m в C#-модель по сценарию s и номере независимого запуска r (необходимого чтобы снизить фактор случайности при генерации). Обозначим результат генерации как

$$G(d, m, s, r).$$

Для двух различных генеративных моделей m_i и m_j требуется оценить, насколько хорошо элементы моделей $G(d, m_i, s, r)$ и $G(d, m_j, s, r)$ сопоставляются алгоритмом выравнивания. Важно, что сравнение выполняется только при одинаковых d , s и r , то есть модели сравниваются внутри одного документа, одного сценария и одного номера запуска, но между разными генеративными моделями.

В данной работе не оценивается корректность модели относительно исходного RFC и не проверяется фактическое поведение сгенерированного кода. Основная измеряемая величина – насколько элементы формальных моделей могут быть сопоставлены автоматическими алгоритмами выравнивания моделей.

Исследовательские вопросы сформулированы следующим образом:

- Улучшает ли инструкция по структурированию и соблюдению семантики именования межмодельную выравниваемость C#-моделей относительно генерации без дополнительных инструкций?
- Улучшает ли обратная связь межмодельную выравниваемость относительно генерации с инструкциями по структурированию и семантическому именванию?
- Насколько устойчивы наблюдаемые эффекты по документам, алгоритмам выравнивания, способам проекции исходного кода на схемы и генеративным моделям?

Методика эксперимента

В качестве корпуса использованы 32 RFC-документа. В настоящей работе RFC рассматриваются как корпус технических спецификаций и документов нормативного характера. Не все RFC имеют одинаковый нормативный статус: серия RFC включает разные типы документов, такие как информационные документы, описания хороших практик, экспериментальные документы. [17]. Для целей настоящего исследования отобраны только документы нормативного характера, по 4 основным темам:

- **HTTP, URI и поведение веб-протоколов** (12 документов) – документы, описывающие HTTP-методы, коды состояния, заголовки, связывание веб-ресурсов (Web Linking), общеизвестные URI (well-known URI), механизм публикации контактной информации по безопасности (security.txt), описание проблемных ответов HTTP API (problem details) и общие рекомендации по построению протоколов поверх HTTP.
- **JSON и форматы данных** (8 документов) – документы, связанные с JSON Pointer, JSON Patch, JSON Merge Patch, I-JSON, JSON Web Token (JWT), схемой канонизации JSON (JSON Canonicalization Scheme) и базовой спецификацией формата JSON.
- **OAuth, токены, аутентификация и механизмы обнаружения** (8 документов) – документы, описывающие токены предъявителя (bearer tokens), отзыв токенов (token revocation), интроспекцию токенов (token introspection), PKCE, JWT-профили для OAuth 2.0, метаданные сервера авторизации (authorization server metadata), поток авторизации устройств (device authorization grant) и взаимную TLS-аутентификацию клиентов с привязкой токенов доступа к сертификатам (mutual TLS client authentication and certificate-bound access tokens).
- **Операционная безопасность и политики доступа** (4 документа) – документы, посвящённые SMTP MTA-STS, DNS CAA, протоколу исключений для роботов (Robots Exclusion Protocol) и универсальным уникальным идентификаторам (UUID).

Количественные характеристики корпуса документов представлены в таблице 1.

Таблица 1. Характеристики корпуса RFC

Метрика	Значение
Документов	32
Всего строк в исходных документах	30 714
Всего слов в исходных документах	156 985
Среднее слов на документ	4 905,8
Медиана слов на документ	4 327
Минимум / максимум слов на документ	1 737 / 14 595

Корпус существенно крупнее использованного в предварительном исследовании [8] и включает документы различной длины. Это позволяет оценивать не только средний эффект сценария, но и его устойчивость на неоднородных технических спецификациях.

Генеративные модели и сценарии. В эксперимент включены шесть генеративных моделей:

- anthropic/claude-haiku-4.5;
- deepseek/deepseek-chat-v3.1;
- google/gemini-2.5-flash;
- mistralai/mistral-medium-3.1;
- openai/gpt-5-mini;
- qwen/qwen3-235b-a22b.

Для каждой комбинации документа, модели и сценария выполнялось пять независимых запусков. Рассматривались три сценария генерации, более детально рассмотренных в таблице 2.

Таблица 2. Сценарии генерации C#-моделей

Сценарий	Описание
control	Генерация C#-доменной модели из исходного документа без дополнительных инструкций.
guided	Генерация с использованием инструкций, задающих правила именования, структурирования и представления элементов модели.
feedback	Сначала выполняется guided-генерация; затем отдельная, более крупная модель (openai/gpt-5) анализирует C#-файл на предмет нарушения инструкций и генерирует обратную связь; после чего исходная модель выполняет ревизию.

Матрица эксперимента задаётся следующими величинами:

$$D = 32, \quad M = 6, \quad S = 3, \quad R = 5,$$

где D – число документов, M – число генеративных моделей, S – число сценариев, R – число независимых запусков. Число полученных финальных артефактов равно $D \cdot M \cdot S \cdot R = 32 \cdot 6 \cdot 3 \cdot 5 = 2880$.

Извлечение элементов схем. После генерации из каждого итогового C#-файла извлекались элементы схемы четырёх типов: enum member, property, relation и type. Далее для выравнивания строились CSV-документы представляющие три слоя проекции: по свойствам, по типам и по отношениям. Разделение на слои нужно потому, что разные элементы модели несут разную информацию. Свойства сильнее зависят от имён и путей, типы – от состава членов, а отношения – от сочетания пути и метаданных. Поэтому агрегированная оценка по всем слоям дополняется отдельным анализом чувствительности к слою проекции.

Протокол выравнивания. Полученные артефакты группировались по исходному документу, сценарию генерации, и номеру запуска. Разные сценарии напрямую друг с другом не выравнивались.

Для каждой группы строились направленные пары между всеми артефактами от разных генеративных моделей, исключая пару модели к самой себе. При шести генеративных-моделях число таких направленных пар равно $M(M - 1) = 6 \cdot 5 = 30$.

Итоговое число направленных пар моделей во всех группах:

$$N_{\text{pairs}} = D \cdot S \cdot R \cdot M(M - 1) = 32 \cdot 3 \cdot 5 \cdot 6 \cdot 5 = 14400.$$

Для каждой пары выполнялось выравнивание по трём слоям проекции (L) и двум алгоритмам выравнивания (A). Таким образом, в ходе эксперимента суммарно выполнено N_{ops} операций выравнивания моделей:

$$N_{\text{ops}} = N_{\text{pairs}} \cdot L \cdot A = 14400 \cdot 3 \cdot 2 = 86400.$$

Использованы алгоритмы выравнивания valentine/coma_py и bdikit/coma.

Из 86 400 операций получено 71 523 непустых срезов выравнивания. Разница возникает в тех случаях, где для конкретной пары слоёв проекции и алгоритма не было непустого набора сопоставляемых элементов.

Метрики выравнивания. Пусть E_s – множество элементов первой модели в паре (исходной), E_t – множество элементов второй модели (целевой), а C – множество кандидатов, найденных алгоритмом сопоставления для данной пары. Тогда source_coverage определяется как доля элементов исходной модели, участвующих хотя бы в одном кандидате:

$$\text{source_coverage} = \frac{|\{e \in E_s : \exists c \in C, \text{src}(c) = e\}|}{|E_s|}.$$

Аналогично target_coverage:

$$\text{target_coverage} = \frac{|\{e \in E_t : \exists c \in C, \text{tgt}(c) = e\}|}{|E_t|}.$$

Метрика coverage_f1 вычисляется как гармоническое среднее между source и target coverage:

$$\text{coverage_f1} = \frac{2 \cdot \text{source_coverage} \cdot \text{target_coverage}}{\text{source_coverage} + \text{target_coverage}}.$$

Метрика top1_mean_score – средняя оценка лучшего кандидата для исходных элементов, у которых есть хотя бы один кандидат. Если $C(e)$ – множество кандидатов для исходного элемента e , то

$$\text{top1_mean_score} = \frac{1}{|E_s^*|} \sum_{e \in E_s^*} \max_{c \in C(e)} \text{score}(c),$$

где E_s^* – множество элементов исходной модели с непустым набором кандидатов.

Основная метрика – pair_alignment_f1:

$$\text{pair_alignment_f1} = \text{coverage_f1} \cdot \text{top1_mean_score}.$$

Эта метрика объединяет два аспекта: насколько большая часть структуры может быть вовлечена в выравнивание и насколько уверенно matcher оценивает лучшие соответствия. Следует отметить, что эта метрика не оценивает истинность соответствий и не заменяет оценку соответствия эталонной разметки, но позволяет эффективно оценить относительное качество в разных комбинациях исходных данных перед выравниванием.

Агрегация и парные сравнения. Основной единицей анализа является срез оценки эффективности выравнивания для пар моделей, определяемый исходным документом, сценарием, номером запуска, двумя генеративными моделями, породившими пару артефактов, слоем проекции и методом выравнивания. Кандидаты сопоставления внутри одной пары не рассматриваются как независимые наблюдения, поскольку они статистически зависимы и порождены одним выполнением сопоставления.

Для сравнения сценариев данные агрегировались в рамках одного сценария. Например, значение guided - control означает разность агрегированных значений метрики в сценариях guided и control, а не прямое выравнивание модели из guided с моделью из control. Дополнительно выполнялось попарное сравнение на уровне документов: для каждого документа считалась разность средних значений pair_alignment_f1 между сценариями, после чего анализировались средняя разность, медиана, число документов с положительной разностью и 95% доверительным интервалом.

Результаты эксперимента. Эксперимент выполнен на полном запланированном объёме из 2880 сгенерированных моделей. Это исключает смещение результатов, связанное с отсутствующими генерациями в отдельных комбинациях документа, модели, сценария или номера запуска. Некоторые количественные характеристики результатов, определенных в ходе эксперимента, приводятся в таблице 4.

Таблица 4. Полнота артефактов

Группа артефактов	Количество
Сгенерировано модели	2 880 / 2 880
Выполнено проекций	2 880 * 3
Выделено элементов моделей	115 614
Построено направленных пар	14 400
Выполнено операций выравнивания	86 400
Сопоставлено элементов	648 269
Оценено попарных срезов	71 523
Оценено срезов по моделям	15 368

Группа артефактов	Количество
Оценено срезов по сценариям	566

Число сопоставленных элементов не используется как число независимых наблюдений. Оно характеризует объём работы алгоритма сопоставления, но статистические сравнения выполняются на агрегированных срезах по направленным парам.

Ошибки в грамматике исходного кода полученных моделей были редкими: медиана числа ошибок равна нулю во всех сценариях. При этом суммарное число количество ошибок парсинга возрастает от control к guided и feedback: 51, 118 и 247 соответственно (см. таблицу 5). Такое распределение показывает, что усложнение сценария представления инструкций для модели может повышать риск нарушения синтаксиса, хотя доля артефактов без ошибок остаётся высокой во всех сценариях.

Таблица 5. Синтаксические ошибки по сценариям

Сценарий	Артефактов	Суммарное число ошибок	Среднее число ошибок на артефакт	Максимальное число ошибок на артефакт	Доля артефактов без ошибок
control	960	51	0,053	7	97,7%
guided	960	118	0,123	50	99,1%
feedback	960	247	0,257	64	98,4%

Важная особенность заключается в том, что guided имеет большее суммарное число ошибок, чем control, но более высокую долю артефактов без ошибок. Следовательно, ошибки в guided концентрируются в меньшем числе файлов, но в отдельных артефактах могут быть многочисленными.

Таблица 6. Синтаксические ошибки по генеративным моделям

Сценарий	Артефактов	Суммарное число ошибок	Среднее число ошибок на артефакт	Доля артефактов без ошибок
anthropic/claude-haiku-4.5	480	0	0,000	100,0%
deepseek/deepseek-chat-v3.1	480	0	0,000	100,0%
qwen/qwen3-235b-a22b	480	36	0,075	99,0%
google/gemini-2.5-flash	480	56	0,117	95,2%
openai/gpt-5-mini	480	99	0,206	99,2%
mistralai/mistral-medium-3.1	480	225	0,469	97,1%

Синтаксические ошибки распределены неоднородно по генеративным моделям. Модели anthropic/claude-haiku-4.5 и deepseek/deepseek-chat-v3.1 не породили ошибок вообще, тогда как на mistralai/mistral-medium-3.1 пришлось наибольшая сумма ошибок (см. таблицу 6). Это важно для практической интерпретации: выигрыш по выравниванию от сценария сценария должен оцениваться вместе с синтаксической устойчивостью генеративной модели.

Размер и структура извлечённых схем. Всего из 2880 C#-артефактов было извлечено 115 614 элементов схем. Сценарий control дал больше элементов, чем guided и feedback. Это означает, что улучшение выравниваемости в структурированных сценариях не объясняется простым ростом числа элементов; наоборот, guided и feedback дают более компактные схемы. Подробные значения по сценариям приводятся в таблицах 7-8.

Таблица 7. Выделено элементов схем по сценариям

Сценарий	Enum members	Properties	Relations	Types	Total
control	6 125	25 319	4 165	8 598	44 207
guided	3 655	21 686	3 618	6 677	35 636
feedback	3 658	21 668	3 644	6 801	35 771

Таблица 8. Среднее число элементов на модель

Сценарий	Среднее	Медиана	Минимум	Максимум
control	46,3	33	1	285
guided	37,2	28	2	219
feedback	37,3	27	2	315

Снижение среднего числа элементов в `guided` и `feedback` можно интерпретировать как эффект структурирования: модели чаще производят менее раздутые схемы с более согласованным набором типов и свойств. Однако само по себе число элементов не является метрикой качества; оно важно только как контекст для интерпретации результатов выравнивания.

Средние оценки по всем документам, парам генеративных моделей, запускам, слоям проекции, и методам выравнивания, сгруппированные по сценариям, показывают устойчивое преимущество сценариев `guided` и `feedback` над `control`. Основные метрики представлены на рис. 1.

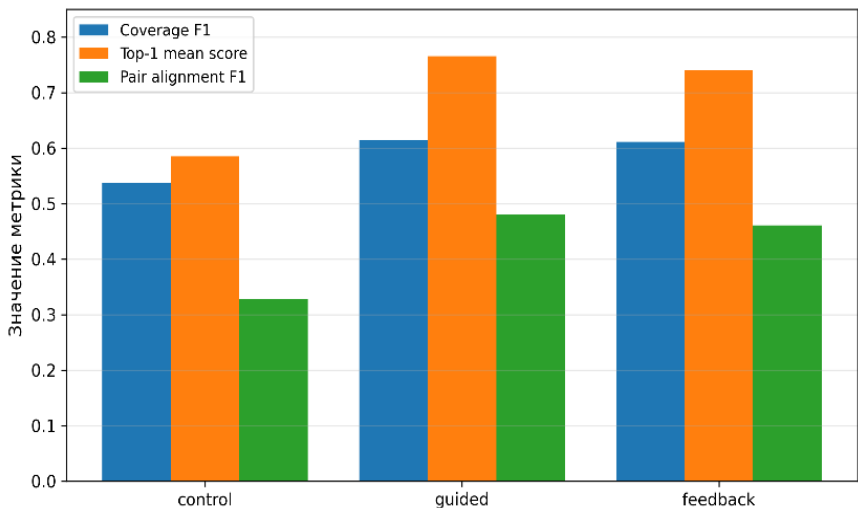


Рис. 1. Основные метрики по сценариям

Подход `guided` увеличивает `pair_alignment_f1` относительно `control` на 0,1517, что соответствует относительному приросту 46,2%. Подход `feedback` также существенно превосходит `control`: абсолютный прирост составляет 0,1323, относительный – 40,3%.

Таблица 9. Сравнение изменений метрик между сценариями

Сравнение	Изменение Coverage F1	Изменение Pair alignment F1	Относительное изменение pair-alignment
guided - control	+0,0766	+0,1517	+46,2%
feedback - control	+0,0738	+0,1323	+40,3%
feedback - guided	-0,0028	-0,0193	-4,0%

Наиболее важный результат состоит в том, что `feedback` не превосходит `guided` по основной качеству выравнивания. Разность `feedback` - `guided` равна -0.0193, то есть `feedback` сохраняет большую часть выигрыша относительно `control`, но не улучшает его.

Механизм этого результата виден из разложения метрики. У `guided` и `feedback` очень близкие значения `coverage_f1`: 0,6140 и 0,6111. Основное различие связано с `top1_mean_score`: 0,7657 у `guided` против 0,7406 у `feedback`. Следовательно, `feedback` не столько ухудшает покрытие, сколько снижает среднюю уверенность алгоритма сопоставления в лучших кандидатах.

Попарное сравнение на уровне документов по метрике `pair_alignment_f1` подтверждает устойчивость улучшения относительно `control`. И `guided`, и `feedback` превосходят `control` во всех 32 документах. При этом `feedback` превосходит `guided` только в 9 из 32 документов. Средняя разность `feedback` - `guided` при этом отрицательна (см. рис. 2).

Этот результат показывает, что инструкции по структурированию и соблюдению семантики именования улучшают межмодельную выравниваемость. Более того, наличие обратной связи не улучшает межмодельную выравниваемость относительно сценария с дополнительными инструкциями.

Оценка в разрезе алгоритмов выравнивания. Оба алгоритма выравнивания моделей дают одинаковое направление эффекта: `guided` и `feedback` превосходят `control`, но `feedback` уступает `guided`. Для `bdikit:coma` разность `feedback` - `guided` равна -0,0213, для `valentine:coma_py` – -0,0174. Это показывает, что основной вывод не является артефактом одной реализации алгоритма сопоставления.

Различие абсолютных показателей метрики качества выравнивания между bdikit:coma и valentine:coma_py, показанных на рисунке 3, вполне ожидаемо, поскольку реализации выравнивания могут использовать разные алгоритмы нормализации, признаки и внутренние параметры. Для данного исследования в первую очередь важны совпадения направления эффекта между методами, а не абсолютные значения метрик.

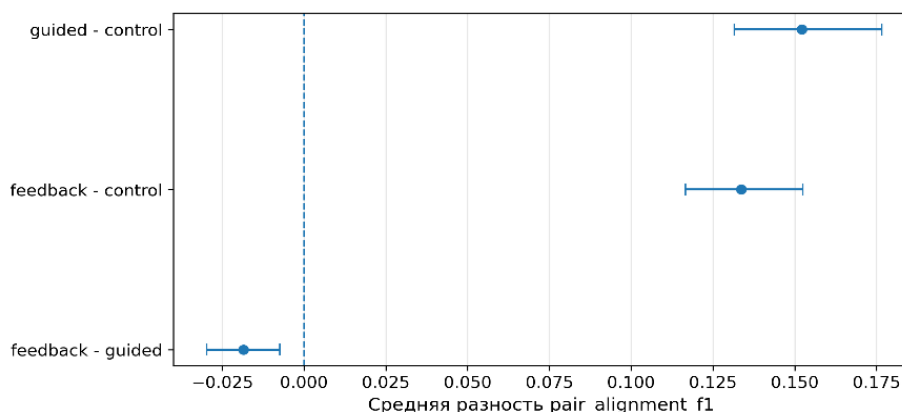


Рис. 2. Средние разности при попарном сравнении на уровне документов

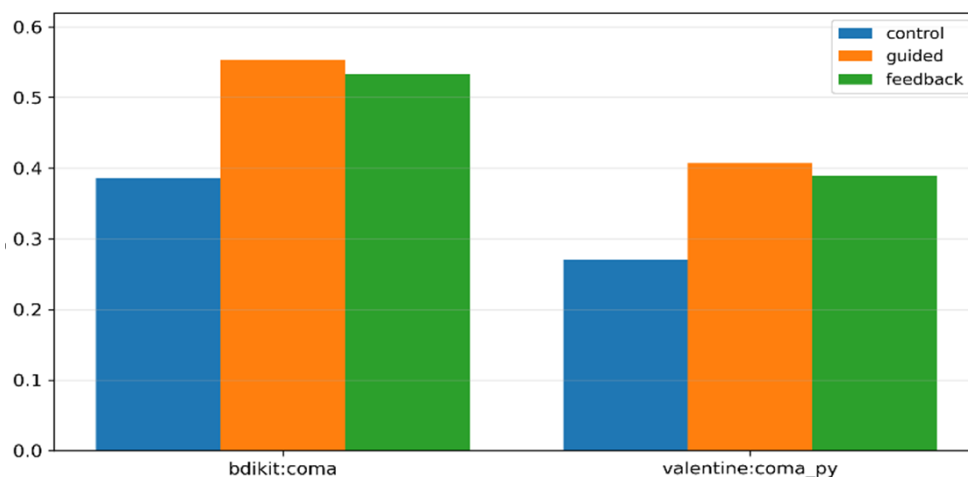


Рис. 3. Значения метрики F1 для попарного выравнивания в разрезе по использованным алгоритмам

Оценка в разрезе слоев проекции. По слоям проекции наблюдается аналогичный порядок: control < feedback < guided по итоговой метрике pair_alignment_f1. Наибольший прирост относительно control наблюдается на уровне сопоставления типов, это означает, что сценарии с дополнительными инструкциями особенно заметно улучшают сопоставимость типов и их членов.

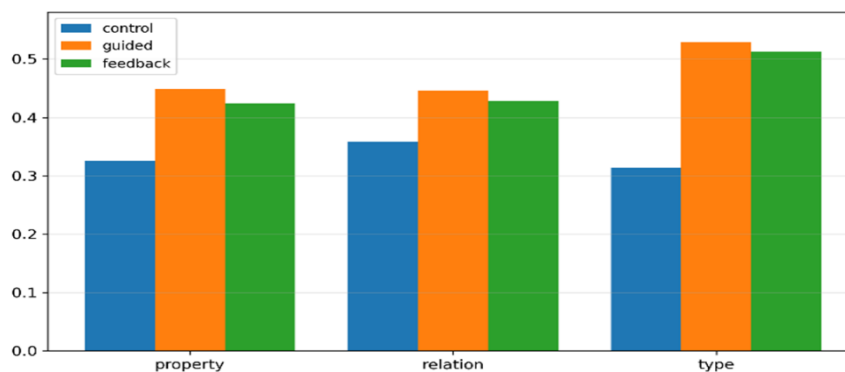


Рис. 4. Значение метрики F1 для попарного выравнивания по слоям проекции

Метрика `coverage_f1` в разрезе слоев проекции показывает, что на слое `relation` сценарий `feedback` слегка превосходит `guided`: 0,6413 против 0,6389. Однако итоговый `pair_alignment_f1` всё равно выше у `guided`, поскольку у `feedback` ниже `top1_mean_score`.

Этот результат важен для интерпретации результатов сценария `feedback`. Обратная связь не разрушает покрытие схемы и даже может немного улучшать его, но одновременно снижает уверенность алгоритмов выравнивания. Поэтому итоговая метрика `feedback` остаётся ниже, чем у `guided`.

Оценка в разрезе генеративных моделей. В разрезе генеративных моделей виден тот же качественный результат: `guided` и `feedback` выше `control`, а `feedback` ниже `guided`. Это снижает вероятность того, что основной эффект вызван одной сильной или слабой моделью.

Таблица 10. `coverage_f1` по `projection layer`

Проекция	Control	Guided	Feedback	Feedback – guided
property	0,5071	0,5727	0,5700	-0,0027
relation	0,6011	0,6389	0,6413	+0,0024
type	0,5274	0,6410	0,6355	-0,0054

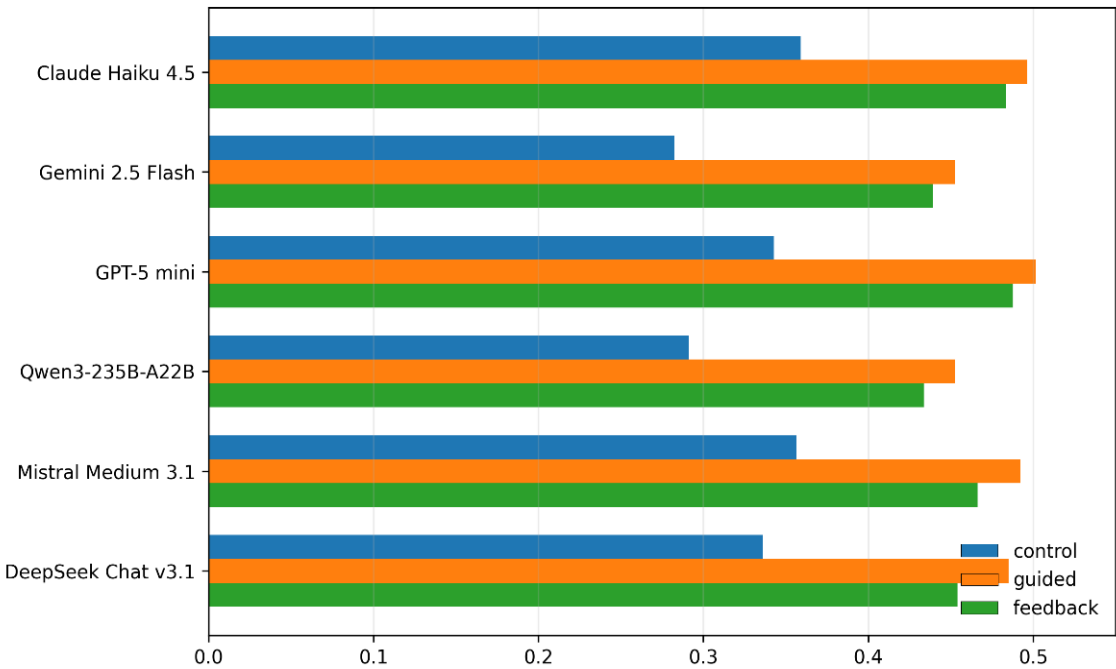


Рис. 5. Значения метрики F1 для попарного выравнивания в разрезе используемой модели

Обсуждение результатов. Почему `guided` оказывается сильнее `feedback` по `alignment`? Сценарии с дополнительными инструкциями задают общий формат генерации до создания модели. Все генеративные модели получают одинаковые правила именования, структурирования и представления элементов. Это снижает произвольность выбора имён для элементов схем и повышает вероятность того, что разные модели используют сходные типы, свойства и отношения. На алгоритмах выравнивания моделей такой эффект сказывается положительно, так как они получают более согласованные элементы для сравнения.

Сценарий с обратной связью, помимо этого стимулирует модель к уточнению. Уточнение может делать имена более предметными и менее общими, но одновременно увеличивать разнообразие конкретных формулировок. Например, одна модель может уточнить поле как «MaximumPayloadSizeOctets», другая – как «MessageBodyLengthLimitBytes», третья – как «AllowedPayloadLength». Для человека эти имена могут быть полезнее, чем абстрактное «Size», но для алгоритма сопоставления они менее похожи друг на друга.

Именно это объясняет наблюдаемую структуру результатов. Значения `coverage_f1` у `guided` и `feedback` почти совпадают, тогда как `top1_mean_score` у `feedback` ниже. Следовательно, `feedback` сохраняет способность вовлекать элементы в выравнивание, но ухудшает среднюю уверенность `matcher` в наилучшем кандидате для сопоставления конкретного элемента.

Для задач массового межмодельного выравнивания текущие результаты указывают на преимущество использования дополнительных инструкций. Для цели получения моделей, которые лучше выравниваются

СОМА-подобными алгоритмами, показано что дополнительные инструкции по структурированию и семантическому именованию сами по себе оказываются более эффективными, чем построение дополнительной ревизии по обратной связи.

Первичная гипотеза о том, что семантическая обратная связь повышает выравниваемость [8], в расширенном эксперименте требует уточнения. Обратная связь действительно повышает выравниваемость относительно контрольного сценария. Однако он не превосходит сценарий *guided* (который сам включён как первый этап сценария *feedback*). Следовательно, положительный эффект *feedback* относительно *control* в значительной степени может быть обусловлен первым этапом (*guided*), а не самой обратной связью и ревизией.

Ограничения исследования. Первое ограничение связано с отсутствием экспертной разметки соответствий. Метрики *coverage_f1*, *top1_mean_score* и *pair_alignment_f1* оценивают поведение автоматических алгоритмов выравнивания, а не истинность каждого соответствия. Поэтому результаты следует интерпретировать как оценку выравниваемости, а не семантической корректности.

Второе ограничение связано с зависимостью основной метрики от алгоритма выравнивания. Использованы две СОМА-подобные реализации (имевшие лучшие показатели в предварительном эксперименте [8]), что позволяет считать выводы более устойчивыми в рамках данного класса алгоритмов. Однако другие алгоритмы, особенно основанные на эмбедингах или больших языковых моделях, могут показать противоположный результат при сравнении *guided-feedback*.

Третье ограничение состоит в том, что модели, полученные из разных сценариев, напрямую друг с другом не сопоставлялись. Сценарии сравнивались через агрегированную оценку внутри сценария. Такой дизайн упрощает сравнительную оценку выравниваемости между разными сценариями, но не отвечает на вопрос, насколько модель из одного сценария совместима с моделью из другого.

Четвёртое ограничение связано с корпусом исходных документов. RFC-документы являются удобным корпусом технических спецификаций, но не покрывают все типы нормативных, юридических, промышленных и организационных документов. Следовательно, перенос результата на другие домены требует дополнительной проверки.

Пятое ограничение связано с C# как языком представления модели. Результаты относятся к моделям, выраженным через C#-типы, свойства, перечисления и отношения. Для других формализмов, например языкам с поддержкой ADT, онтологий OWL, DSL требований или логических правил, структура ошибок и метрики сценариев могут отличаться.

Заключение. Проведённый расширенный эксперимент показал, что сценарий задания инструкций существенно влияет на межмодельную выравниваемость C#-моделей, сгенерированных большими языковыми моделями по техническим спецификациям. Эксперимент выполнен на полном запланированном объёме: 32 RFC-документа, шесть генераторных моделей, три сценария генерации, пять независимых запусков и две СОМА-подобные реализации выравнивания. Всего получено 2880 C#-артефактов, извлечено 115 614 элементов схем, построено 14 400 направленных межмодельных пар и запланировано 86 400 операций выравнивания.

Основной результат состоит в том, что сценарий *guided* является наиболее эффективным сценарием для межмодельного выравнивания. По метрике *pair_alignment_f1* сценарий *guided* увеличивает результат относительно *control* на +0,1517, или +46,2%. Сценарий *feedback* также существенно превосходит *control*, давая прирост +0,1323, или +40,3%, однако не превосходит *guided*: разность *feedback* - *guided* составляет -0,0193, или -4,0%. На уровне документов оба структурированных сценария превосходят *control* во всех 32 документах, тогда как *feedback* превосходит *guided* только в 9 из 32 документов.

Таким образом, исходная гипотеза должна быть уточнена. Семантическая обратная связь полезна как механизм повышения однозначности формальных моделей и сохраняет значительную часть выигрыша при выравнивании относительно контрольного сценария, но в текущем эксперименте не является лучшим способом максимизации *pair_alignment_f1*.

Для массового межмодельного выравнивания наиболее эффективным предоставление инструкции о соблюдении структуры и семантике именования. Дальнейшая работа должна включать экспертную разметку качества выравнивания, проверку качества на связанных задачах, анализ выравнивания для разных сценариев и разработку процедур обратной связи, оптимизирующих не только внутреннюю явность схем, но и устойчивость автоматического сопоставления.

Исходные материалы, алгоритмы и результаты исследования опубликованы в репозитории проекта [18].

Список литературы

1. Проказин М. Р. Гибридный формализм описания нормативных требований DDL-LTLf [Электронный ресурс]. — 2025. — URL: <https://www.cksit-rspp.ru/conferences/video-dokladov-it-standart-2025/> (дата обращения: 17.05.2026).
2. Nute D. Defeasible Deontic Logic. — Dordrecht: Kluwer Academic Publishers, 1997. — URL: <https://unesdoc.unesco.org/ark:/48223/pf0000109572> (дата обращения: 17.05.2026).
3. Ecma International. ECMA-334: C# Language Specification [Электронный ресурс]. — 2023. — URL: <https://ecma-international.org/publications-and-standards/standards/ecma-334/> (дата обращения: 17.05.2026).
4. Rahm E., Bernstein P. A. A Survey of Approaches to Automatic Schema Matching // *The VLDB Journal*. — 2001. — P. 334–350. — DOI: 10.1007/s007780100057.
5. Shvaiko P., Euzenat J. Ontology Matching: State of the Art and Future Challenges // *IEEE Transactions on Knowledge and Data Engineering*. — 2013. — P. 158–176. — DOI: 10.1109/TKDE.2011.253.
6. Object Management Group. Requirements Interchange Format (ReqIF) [Электронный ресурс]. — 2016. — URL: <https://www.omg.org/spec/ReqIF/1.2/> (дата обращения: 17.05.2026).
7. Проказин М. Р., Теремов И. А. Предложения к стандарту, описывающему архитектуру и формат данных умных стандартов // *ИТ-Стандарт*. — 2025. — URL: <https://itstd-journal.ru/wp-content/uploads/2025/02/PROPOSALS-FOR-A-STANDARD-DESCRIBING-THE-ARCHITECTURE.pdf> (дата обращения: 17.05.2026).
8. Проказин М. Р. Семантическая обратная связь при генерации формальных моделей как средство повышения качества межмодельного выравнивания // *Инновации в науке и технологиях: глобальные перспективы и локальные решения: материалы XIV Международной научно-практической конференции*. — 2026.
9. Microsoft. Language Server Protocol Specification [Электронный ресурс]. — URL: <https://microsoft.github.io/language-server-protocol/specifications/lsp/3.17/specification/> (дата обращения: 17.05.2026).
10. Aumuelier D., Do H.-H., Massmann S., Rahm E. Schema and Ontology Matching with COMA++ // *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*. — New York : ACM, 2005. — P. 906–908. — DOI: 10.1145/1066157.1066283.
11. Melnik S., Garcia-Molina H., Rahm E. Similarity Flooding: A Versatile Graph Matching Algorithm and Its Application to Schema Matching // *Proceedings of the 18th International Conference on Data Engineering*. — IEEE, 2002. — P. 117–128. — DOI: 10.1109/ICDE.2002.994702.
12. Madhavan J., Bernstein P. A., Rahm E. Generic Schema Matching with Cupid // *Proceedings of the 27th International Conference on Very Large Data Bases*. — VLDB Endowment, 2001. — P. 49–58. — URL: <https://www.vldb.org/conf/2001/P049.pdf> (дата обращения: 17.05.2026).
13. Koutras C., Siachamis G., Ionescu A., Psarakis K., Brons J., Fragkoulis M., Lofi C., Bonifati A., Katsifodimos A. Valentine: Evaluating Matching Techniques for Dataset Discovery // *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. — IEEE, 2021. — P. 468–479. — DOI: 10.1109/ICDE51399.2021.00047.
14. Lopez R., Liu Y., Koutras C., Freire J. BDI-Kit Demo: A Toolkit for Programmable and Conversational Data Harmonization // *Companion of the International Conference on Management of Data (SIGMOD Companion '26)*. — ACM, 2026. — DOI: 10.1145/3788853.3801608.
15. Liu Y., Pena E. H. M., Santos A., Wu E., Freire J. Magneto: Combining Small and Large Language Models for Schema Matching // *Proceedings of the VLDB Endowment*. — 2025. — P. 2681–2694. — DOI: 10.14778/3742728.3742757.
16. Madaan A., Tandon N., Gupta P., Hallinan L., Gao L., Wiegrefe S., Alon U., Dziri N., Prabhume S., Yang Y., Gupta S., Majumder B. P., Hermann K., Welleck S., Yazdanbakhsh A., Clark P. Self-Refine: Iterative Refinement with Self-Feedback. — *arXiv*, 2023. — DOI: 10.48550/arXiv.2303.17651.
17. Bradner S. The Internet Standards Process — Revision 3 [Электронный ресурс]. — RFC Editor, 1996. — URL: <https://www.rfc-editor.org/rfc/rfc2026> (дата обращения: 17.05.2026).
- Проказин М. Р. Публичный репозиторий с алгоритмами и артефактами эксперимента [Электронный ресурс]. — URL: <https://github.com/bearpro/semantic-feedback-model-alignment> (дата обращения: 17.05.2026).

References

1. Prokazin M. R. Hybrid Formalism for Describing Normative Requirements DDL-LTLf [Electronic resource]. — 2025. — URL: <https://www.cksit-rspp.ru/conferences/video-dokladov-it-standart-2025/> (accessed: 17.05.2026).

2. Nute D. Defeasible Deontic Logic. — Dordrecht: Kluwer Academic Publishers, 1997. — URL: <https://unesdoc.unesco.org/ark:/48223/pf0000109572> (accessed: 17.05.2026).
3. Ecma International. ECMA-334: C# Language Specification [Electronic resource]. — 2023. — URL: <https://ecma-international.org/publications-and-standards/standards/ecma-334/> (accessed: 17.05.2026).
4. Rahm E., Bernstein P. A. A Survey of Approaches to Automatic Schema Matching // *The VLDB Journal*. — 2001. — P. 334–350. — DOI: 10.1007/s007780100057.
5. Shvaiko P., Euzenat J. Ontology Matching: State of the Art and Future Challenges // *IEEE Transactions on Knowledge and Data Engineering*. — 2013. — P. 158–176. — DOI: 10.1109/TKDE.2011.253.
6. Object Management Group. Requirements Interchange Format (ReqIF) [Electronic resource]. — 2016. — URL: <https://www.omg.org/spec/ReqIF/1.2/> (accessed: 17.05.2026).
7. Prokazin M. R., Teremov I. A. Proposals for a Standard Describing the Architecture and Data Format of Smart Standards // *IT-Standard*. — 2025. — URL: <https://itstd-journal.ru/wp-content/uploads/2025/02/PROPOSALS-FOR-A-STANDARD-DESCRIBING-THE-ARCHITECTURE.pdf> (accessed: 17.05.2026).
8. Prokazin M. R. Semantic Feedback in Formal Model Generation as a Means of Improving the Quality of Intermodel Alignment // *Innovations in Science and Technology: Global Perspectives and Local Solutions : Proceedings of the 14th International Scientific and Practical Conference*. — 2026.
9. Microsoft. Language Server Protocol Specification [Electronic resource]. — URL: <https://microsoft.github.io/language-server-protocol/specifications/lsp/3.17/specification/> (accessed: 17.05.2026).
10. Aumüller D., Do H.-H., Massmann S., Rahm E. Schema and Ontology Matching with COMA++ // *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*. — New York : ACM, 2005. — P. 906–908. — DOI: 10.1145/1066157.1066283.
11. Melnik S., Garcia-Molina H., Rahm E. Similarity Flooding: A Versatile Graph Matching Algorithm and Its Application to Schema Matching // *Proceedings of the 18th International Conference on Data Engineering*. — IEEE, 2002. — P. 117–128. — DOI: 10.1109/ICDE.2002.994702.
12. Madhavan J., Bernstein P. A., Rahm E. Generic Schema Matching with Cupid // *Proceedings of the 27th International Conference on Very Large Data Bases*. — VLDB Endowment, 2001. — P. 49–58. — URL: <https://www.vldb.org/conf/2001/P049.pdf> (accessed: 17.05.2026).
13. Koutras C., Siachamis G., Ionescu A., Psarakis K., Brons J., Fragkoulis M., Lofi C., Bonifati A., Katsifodimos A. Valentine: Evaluating Matching Techniques for Dataset Discovery // *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. — IEEE, 2021. — P. 468–479. — DOI: 10.1109/ICDE51399.2021.00047.
14. Lopez R., Liu Y., Koutras C., Freire J. BDI-Kit Demo: A Toolkit for Programmable and Conversational Data Harmonization // *Companion of the International Conference on Management of Data (SIGMOD Companion '26)*. — ACM, 2026. — DOI: 10.1145/3788853.3801608.
15. Liu Y., Pena E. H. M., Santos A., Wu E., Freire J. Magneto: Combining Small and Large Language Models for Schema Matching // *Proceedings of the VLDB Endowment*. — 2025. — P. 2681–2694. — DOI: 10.14778/3742728.3742757.
16. Madaan A., Tandon N., Gupta P., Hallinan S., Gao L., Wiegrefe S., Alon U., Dziri N., Prabhunoy S., Yang Y., Gupta S., Majumder B. P., Hermann K., Welleck S., Yazdanbakhsh A., Clark P. Self-Refine: Iterative Refinement with Self-Feedback. — arXiv, 2023. — DOI: 10.48550/arXiv.2303.17651.
17. Bradner S. The Internet Standards Process — Revision 3 [Electronic resource]. — RFC Editor, 1996. — URL: <https://www.rfc-editor.org/rfc/rfc2026> (accessed: 17.05.2026).
18. Prokazin M. R. Public Repository with Algorithms and Experimental Artifacts [Electronic resource]. — URL: <https://github.com/bearpro/semantic-feedback-model-alignment> (accessed: 17.05.2026).