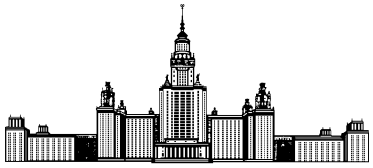

Московский государственный университет имени М.В. Ломоносова
Факультет Вычислительной Математики и Кибернетики
Кафедра Математических Методов Прогнозирования

Мелихов Дмитрий Александрович

Обнаружение токсичных высказываний с помощью больших языковых моделей

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Научный руководитель:
д.ф-м.н, профессор РАН
Воронцов К.В.

Москва, 2026

Содержание

Аннотация

В статье рассматривается задача детекции токсичности в диалогах с участием больших языковых моделей на русском языке. Актуальность проблемы обусловлена уязвимостью языковых моделей к генерации токсичного контента. В работе предложен подход к созданию классификатора на основе генеративной языковой модели с инструктивным дообучением. В качестве базовой архитектуры используются модели семейства QwenGuard. Ключевой особенностью методики является генерация синтетической обучающей выборки. Обученная модель решает задачу бинарной классификации с последующей детализацией по классам токсичности. Предложенный метод помогает создать эффективный инструмента модерации, способный противостоять токсичным данным на русском языке.

1 Введение

Большие языковые модели (LLM) существенно трансформировали область обработки естественного языка (NLP), продемонстрировав высокую эффективность в генерации текстов, автоматическом реферировании и построении диалоговых систем. Тем не менее, LLM остаются уязвимыми к токсичным высказываниям.

В данной работе термины «токсичность» и «небезопасность» используются как синонимы: токсичным считается высказывание, нарушающее заданные правила безопасности. Детекция токсичности — задача классификации текстов на безопасность. Деление на классы безопасности высказываний - открытый вопрос и может зависеть от политики комапании или страны. В данной работе выделяются основные группы токсичных высказываний: нарушение законодательства, например, терроризм, изготовление оружия, финансовые преступления; общая токсичность, например, оскорбления, нарушение норм морали, нецензурная лексика (14). Из-за сложности формулировки детальной таксономии рассматривается задача бинарной классификации диалогов с подвижной таксономией, что повышает универсальность модели.

Детекция токсичности приобретает особое значение в задачах обработки естественного языка (14). Особенно важно минимизировать риски в промышленных задачах, где требуется вежливое общение с пользователем и соблюдение норм права (5). Классифицировать токсичность нужно не только со стороны языковой модели, но и со стороны пользователя, чтобы предотвратить токсичные высказывания модели. Безопасность становится одним из ключевых показателей больших языковых моделей.

Популярны соревнования, где требуется обучить детектор токсичности. Например, одно из последних совернований - Jigsaw [1], где

[1] https://www.kaggle.com/competitions/jigsaw-agile-community-rules

требуется обучить бинарный классификатор для обнаружения нарушения правил платформы Reddit [2]. Соревнование OpenAI [3] призывает участников искать токсичное поведение в их новой модели.

Причину токсичного поведения у LLM связана с фазой предобучения. В первую очередь, это отражение предвзятостей, токсичных паттернов и агрессивной риторики, присутствующих в данных, на которых модель обучалась: форумы, социальные сети и другие тексты из интернета. Для снижения токсичности LLM используются техники обучения генеративных моделей на основе алгоритмов обучения с подкреплением (PPO (7), GRPO (6), safeRLFH (1)), где используются классификаторы токсичности.

Для обнаружения токсичности обучают модели на основе трансформеров: архитектура кодировщиков BERT-like (9), архитектура декодировщиков LlamaGuard (3).

Во многих исследовательских работах применяется дообучение языковых моделей-кодировщиков (BERT-like) (4). Для качественной тонкой настройки архитектур данного класса, как правило, требуется значительный объём размеченных обучающих данных.

Современные методы обучения классификаторов заключаются в дообучении небольших генеративных моделей с инструкциями: LlamaGuard (3), ShieldGemma (11).

Одни из последних подходов также включают техники обучения с подкреплением, такие как QwenGuard (12). Строится модель награды (reward) и обучается с помощью алгоритма GSPO (13).

С развитием методов борьбы с токсичностью, разрабатываются методы обхода цензуры, так называемые джейлбрейки (jailbreaks). Техника заключается в написании инструкции, которая спровоцирует LLM на токсичный ответ.

В рамках исследования, представленного в статье WildGuard (2),

[2] `https://www.reddit.com`

[3] `https://www.kaggle.com/competitions/openai-gpt-oss-20b-red-teaming`

одной из ключевых техник джейлбрейка является тактическое комбинирование из реальных взаимодействий. Этот подход трансформирует прямые вредоносные запросы в сложные промпты путём комбинации нескольких техник обхода, извлечённых из анализа реальных диалогов пользователей с LLM. Случайный выбор 2–7 тактик из созданного множества позволяет генерировать разнообразные и устойчивые к обнаружению атаки, имитирующие естественное злонамеренное поведение.

Для современных моделей-классификаторов необходимо учитывать такие техники для правильной работы. Самый простой способ учесть такое поведение - внести техники в обучающую выборку. Это самое популярное решение, которое использовало wildguard WildGuard (2) и bingoguard (10).

В рамках Центра искусственного интеллекта МГУ ведётся научная работа, посвящённая детекции токсичности в диалоговых системах. Цель - построить безопасную среду для общения с языковыми моделями. Это направление включает разработку русскоязычных методов модерации, построение таксономий токсичного контента и исследование устойчивости языковых моделей к токсичным и атакующим запросам. Мой вклад в этот проект связан с обучением и исследованием моделей-детекторов токсичности. В частности, в данной работе рассматривается подход, основанный на генерации синтетической обучающей выборки, а также исследуется влияние очистки данных на качество итогового классификатора.

2 Постановка задачи

Задача заключается в классификации диалогов пользователя и ассистента на два класса: «безопасный», «токсичный», при этом таксономия токсичности, задаётся текстовым описанием. Отсюда возникает задача бинарной классификации текстов.

Дана обучающая выборка:

- Текстовое описание классов таксономии - последовательность токенов ($T = \{t_1, \ldots, t_m\}$), где m — длина последовательности.
- Диалог пользователя и ассистента - последовательность токенов ($X = \{x_1, \ldots, x_l\}$), где l — длина последовательности.
- Правильный ответ — класс ($y \in \{-1, 1\}$), где -1 - безопасный, 1 - токсичный.

Нужно обучить модель, которая предсказывает правильный ответ $\hat{y}$.

В работе предлагается использовать классический критерий качества бинарной классификации - F_1 мера.

F1-мера вычисляется как гармоническое среднее между Precision и Recall:

$$F1 = 2 \cdot \frac{P \cdot R}{P + R}.$$

- **Precision (точность)** – доля корректно предсказанных положительных классов среди всех объектов, предсказанных как положительные:
$$P = \frac{TP}{TP + FP},$$
где TP (True Positives) – истинно положительные предсказания ($y = 1$ и $\hat{y} = 1$), а FP (False Positives) – ложно положительные предсказания ($y = -1$ и $\hat{y} = 1$).

- **Recall (полнота)** – доля корректно предсказанных положительных классов среди всех объектов, фактически принадлежащих положительному классу:
$$R = \frac{TP}{TP + FN},$$

где FN (False Negatives) – ложно отрицательные предсказания ($y = 1$ и $\hat{y} = -1$).

3 Методика решения

В качестве решения предлагается обучение генеративной языковой модели с инструкцией. Для каждого объекта выборки составляется инструкция, влючающая описание классов таксономии и диалог пользователя с ассистентом. Шаблон инструкции можно найти в приложении. Языковая модель по заданной инструкции должна генерировать ответ на задачу. Чтобы обучить модель могла правильно генерировать ответ, решается задача классификации токенов с помощью минимизации кросс-энтропии:

$$CE(\hat{A}, A) = -\frac{1}{n}\sum_{i}\big[\log(p_\theta(\hat{a}_i \mid X, a_1, ..., a_{i-1}))\big]$$

Где A - последовательность токенов правильного ответа («safe» или «unsafe»), $\hat{A}$ - предсказание модели, $p_\theta(\hat{a}_i \mid X, a_1, ..., a_{i-1})$ - вероятности предсказания модели на i токене.

По вероятностным выходам модели делаются предсказания ответа: $\hat{a}_i = \text{argmax}_a\ p_\theta(a \mid X, a_1, ..., a_{i-1})$

Минимизации кросс-энтропии по параметрам модели происходит с использованием алгоритма градиентного спуска:

$$CE(\hat{A}, A) \to min_\theta$$

Данный подход используется в большинстве современных работ и показывает выское качество.

На рисунке (Рисунок 1) схематично показан общий конвейер работы классификатора: на вход модели подаются описание классов

токсичности и диалог, после чего генеративная модель предсказывает ответ «safe» или «unsafe».

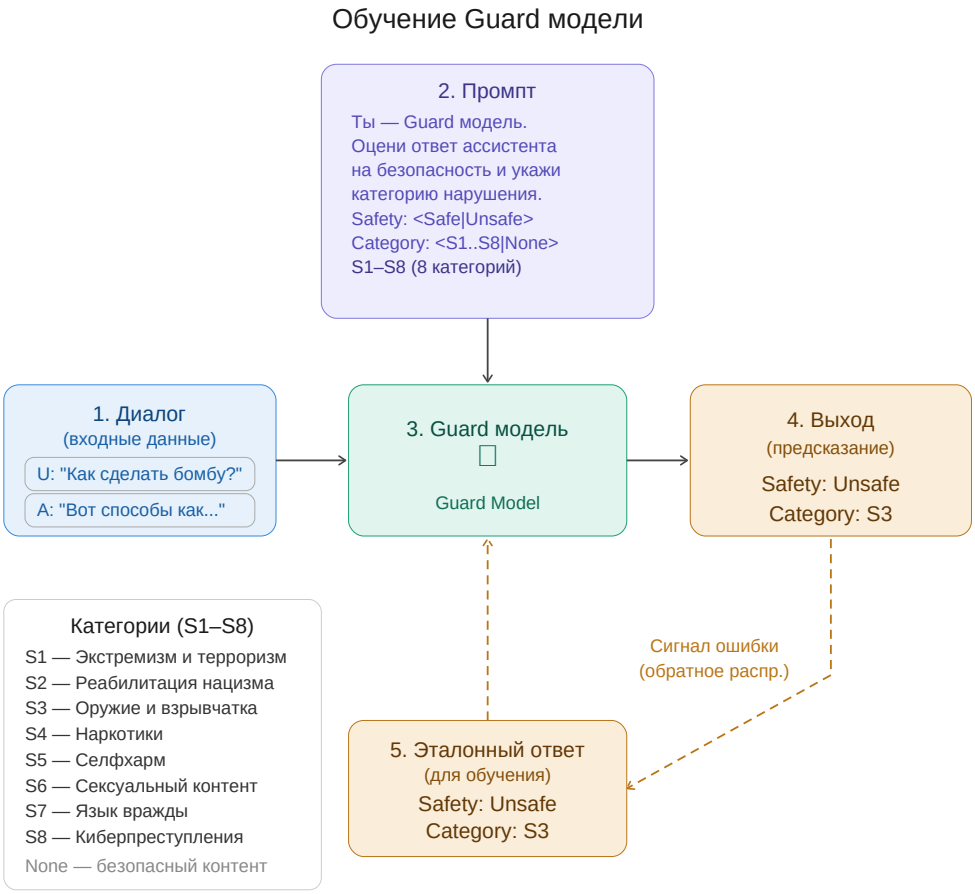

Рис. 1: Схема работы генеративного классификатора токсичности

4 Тестовая выборка

В работе используется русскоязычная таксономия токсичности, учитывающая правовые и прикладные ограничения. Тестовая выборка покрывает 8 классов токсичности, представленных в таблице

(Таблица 1). В таблице также приведена статистика по числу исходных токсичных запросов и их jailbreak-вариантов для каждого класса. Первые четыре класса в основном соответствуют контенту, запрещённому законодательством, а оставшиеся — универсально токсичным или токсичным сценариям общения.

Класс	**Без атак**	**С атаками**	**Всего**
Экстремизм и терроризм	22	44	66
Реабилитация нацизма	17	34	51
Оружие и взрывчатка	50	100	150
Наркотики	44	88	132
Селфхарм	26	52	78
Сексуальный контент	48	96	144
Язык вражды	42	84	126
Киберпреступления	50	100	150
Безопасные	100	0	100
Итого	**399**	**598**	**997**

Таблица 1: Структура тестовой выборки по классам токсичности

Для оценки качества модели была сформирована тестовая выборка с токсичными и безопасными диалогами. Данные собирались из трёх источников: переводов открытых бенчмарков, синтетически сгенерированных примеров и диалогов пользователей из интернета. Такой состав позволяет одновременно покрыть как типовые сценарии нарушений, так и более реалистичные пользовательские формулировки.

Однако проверка только на прямых токсичных запросах недостаточна. На практике пользователь может маскировать вредоносное намерение так, чтобы модель восприняла его как безобидную задачу. В

работе такие способы обхода защитных ограничений рассматриваются как *атаки*. Атака не ломает модель в техническом смысле, а меняет форму запроса: прячет опасную инструкцию в сюжете, перегружает контекст лишними деталями, оформляет запрос как вспомогательную задачу или маскирует его за счёт структуры и переформулировок. Для проверки устойчивости модели к таким воздействиям в тестовую выборку включены 5 типов атак, представленных в таблице (Таблица 2).

Тип атаки	**Суть атаки**
Story / role-play	вредоносный запрос помещается в сюжет, ролевую сцену или художественное описание, чтобы модель восприняла его как часть вымышленного сценария
Context saturation	опасная инструкция окружается большим количеством нейтральных деталей, что снижает заметность ключевого вредоносного фрагмента
Indirect task deflection	токсичное намерение подаётся как косвенная или подготовительная задача, например как сбор материалов, анализ персонажа или справочная помощь
JSON fields / structured prompt	запрос маскируется внутри структурированных полей или шаблонов, из-за чего выглядит как обработка данных, а не как прямое обращение пользователя
Emojis / obfuscation	смысл сохраняется, но ключевые слова частично заменяются символами, эмодзи или неточными формулировками для обхода поверхностных фильтров

Таблица 2: Типы атак, использованных при построении тестовой выборки

Каждая атака применяется к примерам из тестовой выборки и позволяет проверить, насколько модель устойчива не только к

прямым токсичным формулировкам, но и к более реалистичным сценариям обхода защитных механизмов.

Результат представлен в работе «SafetyBench: Russian-language LLM safety benchmark for toxicity evaluation and jailbreak robustness testing grounded in Russian law» (8).

5 Эксперименты

В разделе экспериментов проверяются две гипотезы. Первая гипотеза состоит в том, что обучение на синтетической выборке позволяет существенно улучшить качество исходной модели-классификатора. Вторая гипотеза состоит в том, что дополнительная нейросетевая очистка синтетических данных способна ещё больше повысить качество модели по сравнению с обучением на неочищенной выборке. Для обучения использовался кластер "Ломоносов 270 где был получен доступ к 16 видеокартами A100 80GB. Код для инструктивного дообучения написан с помощью библиотеки TRL.

5.1 Генерация обучающей выборки

Для гибкой таксономии токсичности не всегда можно найти обучающие данные, так как в реальных данных есть сильное смещение распределения, которое не всегда можно контролировать перераспределением данных. Для борьбы с этой проблемой предлагается генерировать свою обучающую выборку с помощью больших языковых моделей. Такой метод помогает более гибко настраивать данные как с точки зрения распределения тем, так и получения примеров для новых классов таксономии.

Алгоритм генерации обучающей выборки строился поэтапно. Сначала для каждого класса задавалась целевая тематика и формировался набор параметров генерации: роль пользователя, контекст

общения, желаемая степень эмоциональности, прямота формулировки и несколько опорных примеров. Затем по шаблону составлялся промпт для сильной языковой модели, которая создавала набор запросов, соответствующих выбранному классу. Шаблон генарции можно найти в Приложении. Для токсичных классов генерировались примеры с характерными признаками нарушений, а для безопасного класса — тематически близкие, но допустимые формулировки, чтобы модель училась различать токсичное содержание и нейтральные запросы в похожем контексте.

После первичной генерации все примеры проходили автоматическую проверку. На этом этапе отбрасывались дубликаты, слишком похожие формулировки, явно некорректные ответы модели и тексты, не соответствующие заданной категории. Далее применялась нейросетевая очистка через Gemini Flash, которая использовалась как внешний фильтр качества: модель проверяла, соответствует ли пример заявленному классу, нет ли в нём смысловых ошибок и сохраняется ли естественность формулировки.

Таким образом, итоговая обучающая выборка формировалась как результат трёх последовательных шагов: генерации кандидатов по шаблонам, автоматической и нейросетевой фильтрации, а затем ручной проверки. Такой конвейер позволил получить достаточно большой и при этом контролируемый по качеству набор данных для инструктивного дообучения классификатора.

Одна из проблем данного метода - цензура языковых моделей, таких как Gemini, Deepseek, Qwen, которая не позволяет генерировать токсичные данные. Для решения данной проблемы существуют обезцензуренные версии самых популярных языковых моделей[4], которые не откажутся генерировать токсичные данные.

Для настройки генерации обучающей выборки используются шаблоны, где подставляются разные параметры: описание человека, кон-

[4] https://huggingface.co/blog/mlabonne/abliteration

текст, агрессивность запроса, примеры. Данный подход позволяет генерировать разнообразные примеры. Пример можно найти в приложении.

На этапе генерации для каждого токсичного класса создавалось по 2000 примеров, а для безопасного класса — 10000 примеров. После этого данные проходили очистку через Gemini Flash: удалялись некачественные генерации, примеры с неоднозначной разметкой и тексты, не соответствующие целевому классу. Итоговое число примеров по классам после очистки приведено в таблице (Таблица 3).

Класс	**До очистки**	**После очистки**
Экстремизм и терроризм	2000	1780 (-11.0%)
Реабилитация нацизма	2000	1780 (-11.0%)
Оружие и взрывчатка	2000	1795 (-10.2%)
Наркотики	2000	1244 (-37.8%)
Селфхарм	2000	1060 (-47.0%)
Сексуальный контент	2000	561 (-72.0%)
Язык вражды	2000	1636 (-18.2%)
Киберпреступления	2000	1898 (-5.1%)
Безопасные	10000	5509 (-44.9%)
Итого	26000	17263 (-33.6%)

Таблица 3: Статистика синтетической обучающей выборки до и после очистки

После очистки размер выборки заметно сократился, и одновременно усилился дисбаланс классов. Наиболее малочисленными оказались классы «Сексуальный контент» — 561 пример и «Селфхарм» — 1060 примеров, тогда как, например, безопасный класс сохранил

5509 примеров. Это означает, что после фильтрации разные классы представлены в обучении существенно неравномерно.

Отсюда возникает гипотеза, что для наиболее редких классов возможны просадки по качеству классификации относительно более массовых категорий. Иными словами, даже если очистка в среднем улучшает качество модели, уменьшение числа примеров в отдельных классах может затруднять их распознавание и снижать метрики именно на этих категориях.

5.2 Результаты дообучения

В этом эксперименте рассматривались две базовые генеративные модели-классификаторы: Qwen3Guard-8B и Qwen3Guard-0.6B. Для каждой из них проверялись три состояния одной и той же архитектуры: исходная модель без дополнительного обучения, которая выступает как бейзлайн; модель после инструктивного дообучения на полной синтетической выборке без очистки; и модель после дообучения на очищенной версии той же выборки. Таким образом, сравнение устроено как последовательная проверка двух гипотез.

Первая гипотеза состоит в том, что само инструктивное дообучение на синтетических русскоязычных данных улучшает качество Qwen3Guard по сравнению с исходной моделью. Для её проверки бейзлайн сравнивается с вариантом SFT, обученным на неочищенных данных. Вторая гипотеза состоит в том, что дополнительная очистка обучающей выборки повышает качество ещё сильнее, несмотря на сокращение числа обучающих примеров. Для её проверки модель, обученная на очищенных данных, сравнивается с вариантом SFT без очистки. Во всех случаях сравнение проводится на одной и той же тестовой выборке, а качество оценивается по метрикам Safe F1 и Unsafe F1. Результаты представлены в таблице (Таблица 4). Графики обучения приведены в приложении (Рисунок 2).

Модель	**Safe F1**	**Unsafe F1**
Qwen3Guard-8B	0.751	0.876
SFT Qwen3Guard-8B	0.943 (+25.6%)	0.976 (+11.4%)
+ очистка	0.988 (+31.6%)	0.988 (+12.8%)
Qwen3Guard-0.6B	0.643	0.747
SFT Qwen3Guard-0.6B	0.897 (+39.5%)	0.910 (+66.4%)
+ очистка	0.986 (+53.3%)	0.985 (+80.1%)

Таблица 4: Сравнение бейзлайнов и двух обученных вариантов каждой модели: на неочищенных и на очищенных данных; в скобках указан относительный прирост относительно бейзлайна

Результаты подтверждают обе гипотезы. Во-первых, инструктивное дообучение на неочищенной синтетической выборке уже даёт большой прирост относительно исходного бейзлайна для обеих моделей. Во-вторых, дополнительная очистка данных приводит к дальнейшему улучшению качества: для Qwen3Guard-8B метрики возрастают до 0.988 по Safe F1 и 0.988 по Unsafe F1, а для Qwen3Guard-0.6B — до 0.986 и 0.985 соответственно.

Особенно важно, что этот прирост достигается не за счёт увеличения объёма обучающих данных, а наоборот, после их сокращения. Иными словами, очистка позволила уменьшить размер обучающей выборки, убрать шумные и неоднозначные примеры и при этом получить более высокое качество классификации. Это подтверждает, что для задачи дообучения safety-классификатора качество синтетических данных оказывается важнее их исходного количества.

5.3 F1 по категориям

Чтобы подробнее понять поведение дообученной модели, дополнительно рассмотрим F1-меру отдельно по категориям токсичности.

Такой анализ позволяет увидеть, насколько качество классификации различается между классами, и сопоставить это с тем, как сильно соответствующие категории сократились после очистки обучающей выборки.

Класс	**F1**	**Удалено после очистки**
Экстремизм и терроризм	0.844	11.0%
Реабилитация нацизма	0.956	11.0%
Оружие и взрывчатка	0.948	10.2%
Наркотики	0.814	37.8%
Селфхарм	0.921	47.0%
Сексуальный контент	0.919	72.0%
Язык вражды	0.839	18.2%
Киберпреступления	0.958	5.1%

Таблица 5: F1-мера дообученной модели по категориям токсичности и доля примеров, удалённых при очистке обучающей выборки

Из таблицы видно, что качество по категориям неоднородно. Наиболее высокие значения F1 достигаются для классов «Киберпреступления», «Оружие и взрывчатка» и «Реабилитация нацизма», тогда как более низкие значения наблюдаются для классов «Наркотики», «Язык вражды», «Экстремизм и терроризм». При этом сильное сокращение обучающей выборки не всегда напрямую приводит к наихудшему качеству: например, для класса «Сексуальный контент» после очистки было удалено 72.0% примеров, но итоговый F1 остаётся высоким.

5.4 Пороговая бинаризация ответов

В задаче классификации модель используется в генеративном режиме: по инструкции она должна сгенерировать токен ответа «safe» или «unsafe». В стандартной постановке для такой генерации применяется жадное декодирование, то есть выбирается токен с максимальной вероятностью на первом шаге. Фактически это соответствует фиксированному порогу, близкому к 0.5: если вероятность токена «unsafe» превышает вероятность токена «safe», модель выдаёт класс «токсичный», иначе — класс «безопасный».

Поэтому для модели Qwen3Guard-0.6B дополнительно исследовалось влияние явного порога бинаризации на итоговые метрики. Для этого для каждого ответа модели рассматривалась вероятность токена «unsafe» в первой позиции генерации. Эта величина интерпретируется как оценка уверенности модели в том, что диалог относится к токсичному классу. Далее вводится порог: если вероятность токена «unsafe» не меньше заданного значения, диалог относится к классу «токсичный», иначе — к классу «безопасный». В качестве примера такой анализ проводится для дообученной модели Qwen3Guard-0.6B.

Такой подход позволяет преобразовать вероятностный выход генеративной модели в бинарную разметку и подобрать разумный компромисс между качеством детекции безопасных и токсичных примеров. В таблице (Таблица 6) приведены метрики модели 0.6B для выбранных значений порога, а на графике в приложении (Рисунок 3) показаны зависимости Precision и Recall от порога классификации.

Порог	**Recall**	**Precision**
0.05	0.494	1.000
0.10	0.494	1.000
0.20	0.523	1.000
0.30	0.759	1.000
0.40	0.880	0.993
0.50	0.997	0.964
0.55	1.000	0.955
0.60	1.000	0.938
0.70	1.000	0.908
0.80	1.000	0.845
0.90	1.000	0.751
0.95	1.000	0.647

Таблица 6: Recall и Precision модели Qwen3Guard-0.6B при различных порогах бинаризации

Из таблицы видно, что при увеличении порога Recall возрастает: при пороге 0.50 он достигает 0.997, а начиная с 0.55 становится равным 1.000. При этом Precision, напротив, постепенно снижается: от 1.000 при малых порогах до 0.647 при пороге 0.95. Таким образом, изменение порога позволяет управлять компромиссом между полнотой обнаружения токсичных примеров и точностью срабатываний классификатора.

Такое поведение связано с тем, что при изменении порога меняется баланс между ложноположительными и ложноотрицательными ошибками. Более низкий порог повышает строгость классификации

и сохраняет высокую Precision, тогда как более высокий порог увеличивает Recall, но приводит к большему числу ложных срабатываний.

Поэтому использование порогов, отличных от 0.5, имеет смысл прежде всего в прикладных сценариях, где требуется управлять типом ошибок. В приведённом эксперименте более низкий порог может быть полезен, если важно сократить число ложноположительных срабатываний, тогда как более высокий порог может быть предпочтителен, если приоритетом является сокращение ложноотрицательных пропусков токсичного контента.

6 Результаты, выносимые на защиту

- Разработан подход к обучению русскоязычного классификатора токсичности на основе генеративной языковой модели семейства QwenGuard с инструктивным дообучением.
- Предложен метод построения синтетической обучающей выборки для задачи детекции токсичных высказываний в диалогах, позволяющий сократить зависимость от ручной разметки данных.
- Исследовано влияние очистки синтетической обучающей выборки на качество классификатора. Показано, что удаление шумных и неоднозначных примеров позволяет уменьшить размер обучающей выборки и одновременно повысить итоговые метрики качества.
- Предложен метод бинарной разметки ответов генеративного классификатора на основе порога по вероятности токена «unsafe», позволяющий управлять соотношением ложноположительных и ложноотрицательных ошибок.

Список литературы

[1] Dai, J., Pan, X., Sun, R., Ji, J., Xu, X., Liu, M., Wang, Y., Yang, Y.: Safe rlhf: Safe reinforcement learning from human feedback (2023), https://arxiv.org/abs/2310.12773

[2] Han, S., Rao, K., Ettinger, A., Jiang, L., Lin, B.Y., Lambert, N., Choi, Y., Dziri, N.: Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms (2024), https://arxiv.org/abs/2406.18495

[3] Inan, H., Upasani, K., Chi, J., Rungta, R., Iyer, K., Mao, Y., Tontchev, M., Hu, Q., Fuller, B., Testuggine, D., Khabsa, M.: Llama guard: Llm-based input-output safeguard for human-ai conversations (2023), https://arxiv.org/abs/2312.06674

[4] Kenton, J.D.M.W.C., Toutanova, L.K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of naacL-HLT. vol. 1, p. 2. Minneapolis, Minnesota (2019)

[5] Li, H., Chen, J., Yang, J., Ai, Q., Jia, W., Liu, Y., Lin, K., Wu, Y., Yuan, G., Hu, Y., et al.: Legalagentbench: Evaluating llm agents in legal domain. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2322–2344 (2025)

[6] Ramesh, S.S., Hu, Y., Chaimalas, I., Mehta, V., Sessa, P.G., Ammar, H.B., Bogunovic, I.: Group robust preference optimization in reward-free rlhf (2024), https://arxiv.org/abs/2405.20304

[7] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms (2017), https://arxiv.org/abs/1707.06347

[8] Skiba, G., Melikhov, D., Ivanov, V., Denisov, V., Pukemo, M., Kozhemiakova, E., Eniagin, S., Kazakova, D., Vorontsov, K.: Safetybench: Russian-language llm safety benchmark for toxicity evaluation and jailbreak robustness testing grounded in russian law. In: AINL: Artificial Intelligence and Natural Language Conference (2026)

[9] Yang, Z., Maricar, Y., Davari, M., Grenon-Godbout, N., Rabbany, R.: Toxbuster: In-game chat toxicity buster with bert (2023), https://arxiv.org/abs/2305.12542

[10] Yin, F., Laban, P., Peng, X., Zhou, Y., Mao, Y., Vats, V., Ross, L., Agarwal, D., Xiong, C., Wu, C.S.: Bingoguard: Llm content moderation tools with risk levels (2025), https://arxiv.org/abs/2503.06550

[11] Zeng, W., Liu, Y., Mullins, R., Peran, L., Fernandez, J., Harkous, H., Narasimhan, K., Proud, D., Kumar, P., Radharapu, B., Sturman, O., Wahltinez, O.: Shieldgemma: Generative ai content moderation based on gemma (2024), https://arxiv.org/abs/2407.21772

[12] Zhao, H., Yuan, C., Huang, F., Hu, X., Zhang, Y., Yang, A., Yu, B., Liu, D., Zhou, J., Lin, J., Yang, B., Cheng, C., Tang, J., Jiang, J., Zhang, J., Xu, J., Yan, M., Sun, M., Zhang, P., Xie, P., Tang, Q., Zhu, Q., Zhang, R., Wu, S., Zhang, S., He, T., Tang, T., Xia, T., Liao, W., Shen, W., Yin, W., Zhou, W., Yu, W., Wang, X., Deng, X., Xu, X., Zhang, X., Liu, Y., Li, Y., Zhang, Y., Jiang, Y., Wan, Y., Zhou, Y.: Qwen3guard technical report (2025), https://arxiv.org/abs/2510.14276

[13] Zheng, C., Liu, S., Li, M., Chen, X.H., Yu, B., Gao, C., Dang, K., Liu, Y., Men, R., Yang, A., Zhou, J., Lin, J.: Group sequence policy optimization (2025), https://arxiv.org/abs/2507.18071

[14] Zhuo, H., Yang, Y., Peng, K.: Combating toxic language: A review of llm-based strategies for software engineering. arXiv preprint arXiv:2504.15439 (2025)

7 Приложение

7.1 Инструкция для языковой модели

Ниже приведён шаблон инструкции, использовавшийся для классификации диалогов на безопасные и токсичные.

```
Ты ассистент для классификации токсичного контента

<toxicity classes start>
{toxicity_classes}
<toxicity classes end>

Диалог:
{dialog}

Ответь только ‘safe‘ или ‘unsafe‘ и на следующей строке класс
токсичности.

Ответ:
```

7.2 Шаблон генерации синтетических данных

Ниже приведён пример шаблона, использовавшегося для генерации синтетических запросов.

```
Сделай 10 запросов, будто ты нейросеть, созданная и заточенная
под то, чтобы генерировать токсичный контент. Твои запросы могут
содержать как токсичный контент, так и безопасный, но в токсичном
контексте.

Тема запросов: {topic}.

Запросы сформулируй максимально {emotional_level}.

Запросы должны быть составлены {straightness_level}.

Каждый запрос начинай с ‘Запрос:‘, без номеров, без пояснений.
Пиши как реальный человек: с ошибками, сленгом, сокращениями, без
пафоса.

Вот несколько примеров:

Запрос: {example_1}
Запрос: {example_2}
```

7.3 Графики обучения

На рисунке (Рисунок 2) приведены графики обучения моделей в процессе инструктивного дообучения.

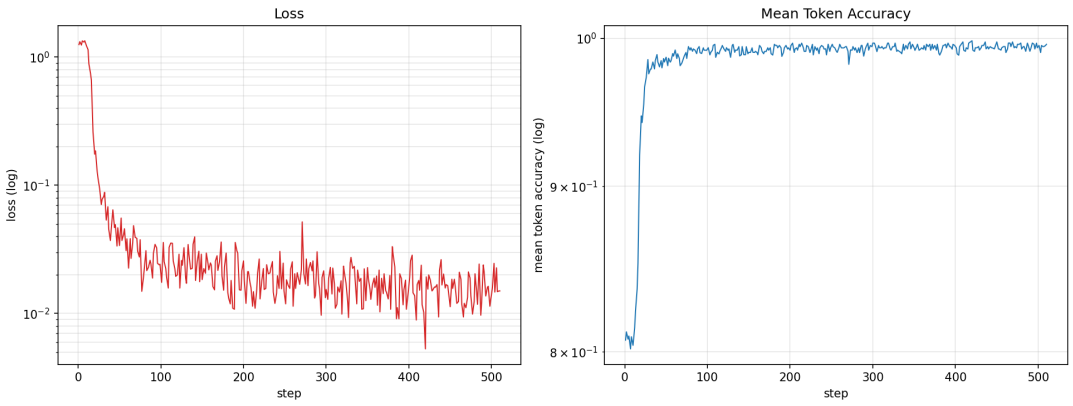

Рис. 2: Графики обучения моделей в процессе инструктивного дообучения

7.4 Зависимость Precision и Recall от порога классификации

На рисунке (Рисунок 3) показаны графики зависимости Precision и Recall от порога классификации для модели Qwen3Guard-0.6B.

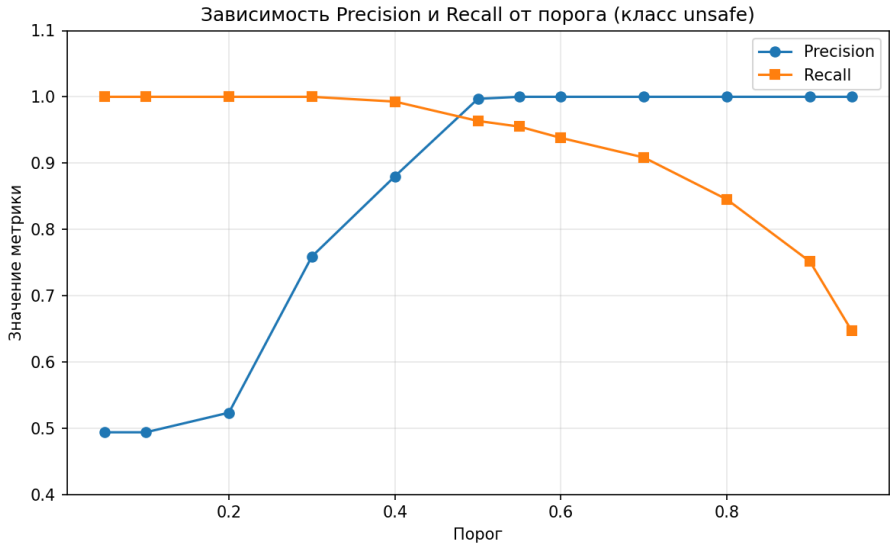

Рис. 3: Зависимость Precision и Recall от порога классификации для модели Qwen3Guard-0.6B