DOI: 10.15514/ISPRAS-2025-37(6)-20

CC BY

Итеративное обучение со слабым контролем с уточнением функций разметки на основе больших языковых моделей

[1,2] *А.Д. Сосновиков, ORCID: 0009-0004-1447-532X <sosnovikov.artur@gmail.com>*
[2] *А.Д. Земеров, ORCID: 0009-0006-4832-7610 <zemerov@tochka.com>*
[1] *Д.Ю. Турдаков, ORCID: 0000-0001-8745-0984 <turdakov@ispras.ru>*

[1] *Институт системного программирования РАН,*
Россия, 109004, г. Москва, ул. А. Солженицына, д. 25.

[2] *Банк Точка,*
Россия, 109044, г. Москва, 3-й Крутицкий пер., д. 11.

Аннотация. Обучение высококачественных классификаторов в условиях ограниченного количества размеченных данных является одной из фундаментальных проблем машинного обучения. Несмотря на то, что большие языковые модели (LLM) демонстрируют впечатляющие результаты при решении задач классификации явного обучения (zero-shot), их прямое применение на практике затруднено из-за высокой вычислительной стоимости, чувствительности к формулировкам запросов (prompt engineering) и ограниченной интерпретируемости. В качестве масштабируемой альтернативы выступает обучение со слабым контролем, которое основано на объединении множества неточных функций разметки (labeling functions, LF). Однако создание и последующая настройка таких функций обычно требует существенных затрат ручного труда. В данной работе мы предлагаем подход LLM-Guided Iterative Weak Labeling (LGIWL), который сочетает генерацию функций разметки с помощью больших языковых моделей и методику обучения со слабым контролем в рамках итеративного цикла обратной связи. Вместо прямого использования LLM в качестве классификатора, мы применяем её для автоматического создания и постепенного уточнения функций разметки на основе ошибок промежуточного классификатора. Полученные функции фильтруются с использованием небольшого размеченного набора данных и затем применяются к неразмеченной выборке при помощи генеративной модели меток. Это позволяет обучить итоговый дискриминативный классификатор высокого качества при минимальных затратах на ручную аннотацию. Эффективность предложенного подхода продемонстрирована на реальной задаче классификации диалогов службы поддержки клиентов на русском языке. LGIWL существенно превосходит как классические эвристики на основе ключевых слов (Snorkel), так и подходы zero-shot на основе GPT-4, а также полностью контролируемый классификатор CatBoost, обученный на размеченных данных аналогичного размера. В частности, вариант LGIWL с моделью RuModernBERT достигает высокого показателя полноты при значительном улучшении точности, демонстрируя итоговый результат по метрике F1 = 0.863. Полученные результаты подтверждают как высокую устойчивость метода, так и его практическую применимость в условиях ограниченных ресурсов размеченных данных.

Ключевые слова: обучение со слабым контролем; финансовый сектор; модели LLM.

Для цитирования: Сосновиков А.Д., Земеров А.Д., Турдаков Д.Ю. Итеративное обучение со слабым контролем с уточнением функций разметки на основе больших языковых моделей. Труды ИСП РАН, том 37, вып. 6, часть 2, 2025 г., стр. 65–76. DOI: 10.15514/ISPRAS–2025–37(6)–20.

Iterative Weak Supervision with LLM-Guided Labeling Function Refinement

[1,2] *A.D. Sosnovikov, ORCID: 0009-0004-1447-532X <sosnovikov.artur@gmail.com>*
[2] A.D. Zemerov, *ORCID: 0009-0006-4832-7610 <zemerov@tochka.com>*
[1] *D.Y. Turdakov, ORCID: 0000-0001-8745-0984 <turdakov@ispras.ru>*

[1] *Institute for System Programming of the Russian Academy of Sciences, 25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.*

[2] *Tochka Bank, 11, 3-y Krutitskiy pereulok, Moscow, 109044, Russia.*

Abstract. Training high-quality classifiers in domains with limited labeled data remains a fundamental challenge in machine learning. While large language models (LLMs) have demonstrated strong zero-shot capabilities, their use as direct predictors suffers from high inference cost, prompt sensitivity, and limited interpretability. Weak supervision, in contrast, provides a scalable alternative through the aggregation of noisy labeling functions (LFs), but authoring and refining these rules traditionally requires significant manual effort. We introduce LLM-Guided Iterative Weak Labeling (LGIWL), a novel framework that integrates prompting with weak supervision in an iterative feedback loop. Rather than using an LLM for classification, we use it to synthesize and refine labeling functions based on downstream classifier errors. The generated rules are filtered using a small development set and applied to unlabeled data via a generative label model, enabling high-quality training of discriminative classifiers with minimal human annotation. We evaluate LGIWL on a real-world text classification task involving Russian-language customer service dialogues. Our method significantly outperforms keyword-based Snorkel heuristics, zero-shot prompting with GPT-4, and even a supervised CatBoost classifier trained on a full labeled dev set. In particular, LGIWL achieves strong recall while yielding a notable improvement in precision, resulting in a final F1 score of 0.863 with a RuModernBERT classifier–demonstrating both robustness and practical scalability.

Keywords: weak supervison; financial sector; LLM.

For citation: Sosnovikov A.D., Zemerov A.D., Turdakov D.Yu. Automating the process of responding to a tax request using weak supervision. Trudy ISP RAN/Proc. ISP RAS, vol. 37, issue 6, part 2, 2025. pp. 65-76 (in Russian). DOI: 10.15514/ISPRAS-2025-37(6)-20.

1. Введение

Обучение с учителем стало доминирующим подходом к обучению высокоэффективных классификаторов в задачах обработки естественного языка. Однако его успешность критически зависит от наличия большого объёма качественно размеченных данных – условия, которое редко выполняется на практике, особенно в специализированных областях, таких как финансы, здравоохранение и клиентская поддержка. Ручная аннотация является дорогостоящей, медленной и зачастую требует значительной предметной экспертизы.

Обучение со слабым контролем предлагает убедительную альтернативу, позволяя специалистам программно задавать эвристики разметки – так называемые функции разметки (labeling functions, LF), которые могут быть применены к большим неразмеченным корпусам. Эти слабые сигналы затем агрегируются с помощью вероятностных моделей для получения шумных, но масштабируемых обучающих меток. Фреймворки, такие как Snorkel, показывают, что при аккуратном проектировании функций разметки модели, обученные на слабо размеченных данных, могут приближаться по качеству к моделям, обученным на полностью ручной разметке. Однако разработка функций разметки остаётся нетривиальной задачей: она требует времени, экспертизы и итеративной доработки, что ограничивает доступность и масштабируемость обучения со слабым контролем на практике.

Тем временем, большие языковые модели (LLM), такие как GPT-4, продемонстрировали выдающиеся способности, позволяя решать новые задачи при минимальном (few-shot) или

полном отсутствии (zero-shot) размеченных данных. Однако использование LLM напрямую в качестве классификаторов порождает новые сложности: вывод (inference) требует значительных вычислительных ресурсов, предсказания чувствительны к формулировке подсказок, а отсутствие интерпретируемости мешает их интеграции в регламентированные или критически важные для производства системы.

В данной работе мы предлагаем новый подход, объединяющий семантическую гибкость больших языковых моделей LLM и надёжность с интерпретируемостью обучения со слабым контролем. Мы представляем метод *LLM-Guided Iterative Weak Labeling* (LGIWL) – фреймворк, в котором модели LLM рассматриваются не как чёрные ящики, а как генераторы функций разметки. LGIWL работает в итеративном цикле: ошибки классификатора используются для формирования подсказок LLM с целью генерации новых или уточнённых функций разметки; полученные функции фильтруются на небольшом валидационном наборе, и их выходы агрегируются для повторного обучения классификатора. Этот процесс, основанный на обратной связи, позволяет автоматически расширять и улучшать множество функций разметки, снижая необходимость в ручной настройке подсказок и разработке правил.

Мы применяем метод LGIWL к реальной задаче классификации обращений в службу поддержки в финансовой жносфере и демонстрируем, что он значительно превосходит метод запросов без явного обучения (zero-shot prompting), статическое обучение со слабым контролем и даже полностью супервизированные модели, обученные на таком же объёме размеченных данных. В частности, LGIWL достигает высокой полноты при существенном росте точности, демонстрируя свою эффективность как масштабируемого и интерпретируемого подхода к обучению в условиях ограниченных ресурсов.

2. Обзор релевантных работ

Программируемое обучение со слабым контролем. Обучение со слабым контролем позволяет создавать размеченные наборы данных, заменяя ручную аннотацию эвристиками или правилами, известными как функции разметки LF. Фреймворк Snorkel ввёл формализованный подход к этому процессу, моделируя точность и корреляцию между функциями разметки с помощью генеративной модели меток [1]. Это позволило агрегировать шумные и потенциально конфликтующие слабые метки. Последующие работы расширили Snorkel для применения в крупномасштабных задачах [2] и разработали методы для выявления структуры между функциями разметки [3]. Позже набор WRENCH стандартизировал протоколы оценки для слабого обучения [4].

Однако, несмотря на существенную автоматизацию моделирования меток, написание функций разметки остаётся ручной и трудоёмкой задачей, часто требующей предметной экспертизы и итеративной доработки. Системы, такие как Snuba [5] и WITAN [6], стремились уменьшить нагрузку на разработку LF путём генерации правил из начальных примеров или обратной связи пользователя. Тем не менее, многие из этих систем по-прежнему требуют участия человека в цикле или зависят от жёстко заданной логики правил.

Разметка на основе подсказок с использованием LLM. Большие языковые модели (LLM) предлагают новый путь для генерации правил путём интерпретации подсказок на естественном языке. В работе [7] было предложено использовать LLM для создания слабых меток путём постановки структурированных вопросов, рассматривая выходные данные LLM как источник шумной супервизии. Система Alfred [8] развила эту идею, применяя подсказки на естественном языке в качестве интерфейса для генерации и управления функциями разметки. Другие системы, такие как ScriptoriumWS [9], ещё больше автоматизируют процесс, преобразуя подсказки в исполняемые функции разметки на языке Python.

Недавно появились полностью итеративные фреймворки. Например, DataSculpt [10] использует LLM для предложения новых функций разметки на основе few-shot подсказок и

уточняет набор правил в ходе нескольких раундов обучения и оценки. PRBoost [11] предложил цикл в стиле бустинга, где сгенерированные правила оцениваются и постепенно добавляются для повышения точности классификации.

Вклад данной работы. Наш метод LGIWL опирается на эти разработки, но вводит ключевое новшество: итеративное уточнение функций разметки, сгенерированных LLM, на основе сигналов об ошибках классификатора, встроенное в структурированную архитектуру обучения со слабым контролем. В отличие от предыдущих методов, которые рассматривают генерацию подсказок или правил как одноразовые шаги, LGIWL замыкает цикл между подсказками LLM, агрегацией меток и обучением дискриминативной модели. Это обеспечивает масштабируемую и адаптивную слабую разметку в реалистичных условиях с ограниченными ресурсами и минимальным участием человека.

3. Постановка задачи

Рассмотрим стандартную задачу бинарной классификации, в которой требуется построить функцию $f_\theta : X \to \{0,1\}$, предсказывающую принадлежность текстового объекта $x \in X$ к заданной семантической категории (например, обращению, связанному с тарифом). При типичном подходе с учителем для обучения модели необходим полностью размеченный набор данных $\mathcal{D}_{\text{sup}} = \{(x_i, y_i)\}^N_{i=1}$. Однако на практике получение такого набора часто затруднено из-за высокой стоимости ручной аннотации, нехватки квалифицированных специалистов или малого количества примеров целевого класса.

Вместо этого будем предполагать, что нам доступен большой неразмеченный корпус текстов $\mathcal{D}_U = \{(x_i)\}^N_{i=1}$ и небольшой размеченный валидационный набор $\mathcal{D}_{\text{dev}} = \{(x_j, y_j)\}^M_{j=1}$, причём $M \ll N$. Валидационный набор используется исключительно для отбора и фильтрации шумных эвристик и не применяется напрямую для обучения итогового классификатора.

Для создания слабых (неточных) меток на корпусе $\mathcal{D}_U$ введём набор функций разметки $\Lambda = \{\lambda_k\}^K_{i=1}$, где каждая функция $\lambda_k : X \to \{0,1,\emptyset\}$, либо сопоставляет входному объекту неточную метку класса, либо воздерживается от разметки. Такие функции могут представлять собой простые правила на основе ключевых слов, регулярные выражения или более сложные семантические конструкции. В рассматриваемой постановке функции разметки LF генерируются путём запроса к большой языковой модели с использованием инструкций и примеров на естественном языке.

Каждый элемент выборки $x_i \in \mathcal{D}_U$ может быть размечен несколькими функциями разметки одновременно, в результате чего формируется матрица меток $L \in \{0, 1, \emptyset\}^{N \times K}$. Вероятностная модель меток $p(y_i \mid L_{i,:})$ агрегирует полученные шумные разметки в виде мягких (вероятностных) меток $\hat{y}_i \in [0, 1]$, на которых затем обучается классификатор f_θ.

Центральная задача состоит в том, чтобы, используя обратную связь от модели f_θ, итеративно улучшать набор функций разметки Λ таким образом, чтобы каждое новое поколение функций LF точнее отражало семантику задачи, обеспечивало большее покрытие выборки и уменьшало уровень шума. Данный подход лежит в основе предлагаемого нами фреймворка LGIWL, который превращает процесс проектирования функций разметки в адаптивный цикл обратной связи, управляемый ошибками обучаемой модели.

4. Методология: итеративное обучение со слабым контролем на основе LLM

Фреймворк LGIWL основан на идее использования больших языковых моделей не в роли непосредственного классификатора, а в качестве генератора семантических правил, встроенного в итеративный процесс обучения со слабым контролем. Целью такого подхода является создание тщательно подобранного набора функций разметки, позволяющих

формировать информативные вероятностные метки для больших неразмеченных наборов данных при минимальном участии человека.

Пусть имеется неразмеченный корпус текстов и небольшой размеченный валидационный набор, где $M \ll N$.

Валидационный набор разделяется на две части: $\mathcal{D}_{\text{dev}}^{\text{prompt}}$, используемую для выявления ошибок текущего классификатора, и $\mathcal{D}_{\text{dev}}^{\text{filter}}$, предназначенную для количественной оценки и фильтрации кандидатных функций разметки.

Каждая итерация $t \in \{1,...,T\}$ цикла LGIWL включает следующие шаги:

i. **Формирование подсказок (Prompt Construction).** Используя примеры из набора $\mathcal{D}_{\text{dev}}^{\text{prompt}}$, на которых текущий классификатор ошибается или демонстрирует неопределённость, формируется текстовая подсказка $\pi^{(t)}$. Примеры группируются по пакетам и передаются в большую языковую модель (например, модель класса GPT), которая генерирует набор кандидатных функций разметки в виде семантических правил, шаблонов или групп ключевых слов. Обозначим полученный набор как $\Lambda_{\text{cand}}^{(t)}$.

ii. **Нормализация и дедупликация (Normalization and Deduplication).** Далее, вновь обращаясь к языковой модели с набором $\Lambda_{\text{cand}}^{(t)}$, происходит слияние семантически эквивалентных правил и удаление зашумлённых или избыточных формулировок, в результате чего получается очищенный набор кандидатных функций.

iii. **Фильтрация на валидационных данных (Filtering on Development Data).** Каждая кандидатная функция разметки $\lambda_k \in \Lambda_{\text{cand}}^{(t)}$ оценивается на наборе $\mathcal{D}_{\text{dev}}^{\text{filter}}$. Функции, для которых показатели оценки точности (precision) и покрытия (coverage) превышают заданные пороги δ_p и δ_c соответственно, включаются в общий накопленный набор функций разметки $\Lambda^{(t)}$.

iv. **Разметка и агрегация (Labeling and Aggregation).** Все накопленные функции разметки из множества $\Lambda^{(t)}$ применяются к неразмеченному корпусу D*U* при помощи лёгкого движка правил (например, t-lite-it-1.0), что приводит к формированию матрицы меток:

$$L \in \{0,1,\emptyset\}^{N \times |\Lambda^{(t)}|}.$$

Затем генеративная модель меток ϕ (например, Snorkel) агрегирует полученные сигналы в вероятностные метки:

$$\hat{y}_i = \phi(L_{i,:}).$$

v. **Цикл обратной связи (Feedback Loop).** На полученных метках $\{(x_i, \hat{y}_i)\}$ обучается временный рабочий классификатор. Ошибки классификации на наборе $\mathcal{D}_{\text{dev}}^{\text{prompt}}$ фиксируются и используются для формирования подсказки следующей итерации $\pi^{(t+1)}$.

Итерации продолжаются до тех пор, пока качество слабых меток на наборе $\mathcal{D}_{\text{dev}}^{\text{filter}}$ не стабилизируется. После достижения стабилизации производится обучение финальной прогнозной модели на полученных слабых метках $\{(x_i, \hat{y}_i)\}$. Таким образом, высокопроизводительный классификатор оказывается полностью отделён от цикла проектирования функций разметки.

Разделяя стадии генерации правил, их применения и финального обучения, метод LGIWL сочетает семантическую гибкость языковых моделей с эффективностью и интерпретируемостью систем, основанных на правилах.

5. Экспериментальная часть

5.1 Задача и набор данных

Мы оцениваем предложенный подход на задаче бинарной текстовой классификации обращений клиентов в службу поддержки на русском языке. Цель задачи – определить, относится ли многоходовой диалог к вопросам, связанным с тарифами. Полный набор данных содержит 10,000 неразмеченных диалогов, собранных с платформы цифрового банкинга. Целевой класс представлен достаточно редко и составляет примерно 8.2%

Для разработки и оценки метода мы дополнительно разметили два подмножества данных. Валидационный набор из 1,000 размеченных диалогов используется в процессе обучения исключительно для генерации и фильтрации функций разметки, но не применяется напрямую для обучения итоговой модели. В рамках предлагаемого фреймворка LGIWL этот набор делится на две равные части: первая используется для формирования подсказок языковой модели на основе ошибок текущего классификатора, а вторая – для оценки качества кандидатных функций разметки перед их включением в общий набор LF.

Отдельный тестовый набор из тысячи диалогов с ручной разметкой оставлен для финальной оценки. Метки из тестового набора не используются при обучении или подборе моделей ни в одном из рассматриваемых подходов со слабым контролем.

5.2 Обзор сравниваемых методов

В экспериментальном сравнении мы используем следующие методы, отражающие различные сценарии обучения (см. табл. 1):

- **Zero-Shot Prompting (LLM).** Базовый zero-shot подход, в котором модель GPT-4 напрямую классифицирует диалоги на тестовом наборе, получая описание задачи и несколько примеров (zero-shot). Данный подход представляет собой классификацию без явного обучения на размеченных данных.
- **Snorkel (Keywords).** Классический подход обучения со слабым контролем, при котором функции разметки создаются вручную или автоматически на основе TF–IDF ассоциаций с целевым классом. Полученные функции агрегируются с помощью Snorkel для генерации вероятностных меток.
- **CatBoost + Snorkel (Keywords).** Дискриминативный классификатор, обученный на слабых метках, полученных с помощью описанного выше подхода Snorkel (Keywords). Этот метод демонстрирует прирост качества от применения дискриминативной модели поверх статических слабых меток.
- **CatBoost (Supervised, Small Data).** Классификатор на основе градиентного бустинга CatBoost, обученный напрямую на 1,000 размеченных примерах из валидационного набора. Этот подход отражает реалистичную ситуацию, когда небольшая ручная разметка данных доступна, и позволяет оценить достаточность простой супервизии в условиях ограниченных данных.
- **Snorkel (Default LLM).** Функции разметки генерируются с помощью общей языковой модели по определению целевого класса и инструкциям, затем агрегируются через Snorkel. Данный подход не использует итеративную обратную связь и представляет собой одношаговый вариант разметки с помощью LLM с базовым заданным вручную описанием целевого класса.
- **LGIWL (CatBoost и RuModernBERT).** Предлагаемый нами метод, комбинирующий ключевые слова, функции разметки на основе LLM и итеративную доработку, основанную на обратной связи от классификатора. Новые LF итеративно

генерируются, оцениваются по точности и покрытию на валидационном наборе и добавляются в агрегируемый набор функций. Итоговые слабые метки используются для обучения классификатора CatBoost или тонко настроенного трансформера RuModernBERT. Этот метод отражает полный подход к адаптивной, масштабируемой разметке.

Табл. 1. Сводка сравниваемых методов и источников разметки.
Table 1. Summary of Compared Methods and Annotation Sources.

Метод	Описание	Источник разметки
Zero-Shot Prompting (LLM)	Прямая Zero-shot классификация с помощью LLM	Только промпты
Snorkel (Keywords)	Статические функции разметки на основе ключевых слов, агрегированные через Snorkel	Эвристика (TF–IDF)
CatBoost + Snorkel (Keywords)	CatBoost, обученный на слабых метках от Snorkel (Keywords)	Слабый контроль (статический)
CatBoost (Supervised, Small Data)	CatBoost, обученный на 1000 размеченных примерах	Ручная разметка (ограниченная)
Snorkel (Default LLM)	LFs, сгенерированные из LLM и агрегированные через Snorkel	Слабый контроль (LLM)
LGIWL (CatBoost / RuModernBERT)	Итеративная генерация LF через LLM, фильтрация и обучение	Слабый контроль (адаптивный)

5.3 Детали реализации

Все сравниваемые методы реализованы с использованием единой инфраструктуры и единых процедур обучения. В качестве дискриминативных моделей используются CatBoost (градиентный бустинг, эффективный на небольших структурированных данных) и RuModernBERT-base (предобученная трансформерная модель для русского языка). Слабые метки генерируются через Snorkel и используются как мягкие целевые метки для обучения моделей. При этом размеченные вручную метки во время обучения дискриминативных моделей не используются.

Функции разметки создаются как комбинация ключевых слов, семантических шаблонов и условий, сгенерированных с помощью LLM. В методе LGIWL новые LF итеративно предлагаются языковой моделью по ошибочным примерам и принимаются в общий набор только в случае превышения порогов покрытия и точности, оцениваемых на выделенной части валидационного набора.

5.4 Протокол оценки

Результаты классификации на тестовом наборе оцениваются по пяти метрикам: площадь под ROC-кривой (AUC), средняя точность (AP), точность (precision), полнота (recall) и F1-мера. Пороги классификации выбираются по максимуму F1-меры на валидационном наборе. Помимо точности классификации, мы анализируем покрытие и разнообразие правил, калибровку моделей и эффективность использования LLM.

6. Результаты и анализ

В данном разделе представлены результаты сравнительного анализа предложенного фреймворка LGIWL и базовых методов, описанных в разделе 5. Оценка проводилась на тестовом наборе из тысячи вручную размеченных диалогов с использованием стандартных метрик качества классификации: площади под ROC-кривой (AUC), средней точности (AP), точности (precision), полноты (recall) и F1-меры. Пороги для классификации подбирались по максимуму F1-меры на валидационном наборе и опущены в описании для удобства чтения.

6.1 Основные результаты

В табл. 2 представлены результаты оценки всех рассматриваемых методов. LGIWL, особенно в реализации на основе RuModernBERT, демонстрирует наилучшие результаты по всем метрикам, превосходя как традиционные подходы со слабым контролем, так и zero-shot классификацию с помощью LLM. Также LGIWL превосходит супервизированную модель CatBoost, обученную на тысяче размеченных примерах.

6.2 Сравнение супервизированного обучения и адаптивного слабого контроля

Хотя супервизированная модель CatBoost, обученная на 1,000 вручную размеченных примерах, показывает стабильные результаты (F1-мера: 0.816), она не превосходит метод LGIWL. Важно подчеркнуть, что LGIWL достигает более высоких результатов, не используя напрямую вручную размеченные данные при обучении, а расходуя аналогичный объём ручной разметки лишь на фильтрацию правил и уточнение подсказок. Это демонстрирует преимущество адаптивного обучения со слабым контролем по сравнению с прямым супервизированным обучением при ограниченном бюджете разметки.

Табл. 2. Сравнение качества моделей на тестовом наборе.
Table 2: Comparison of Model Performance on the Test Set.

Метод	AUC	AP	Точность	Полнота	F1-мера
Zero-Shot Prompting (LLM)	0,926	0,612	0,677	0,890	0,769
Snorkel (Keywords)	0,975	0,821	0,765	0,712	0,738
CatBoost + Snorkel (Keywords)	0,979	0,828	0,836	0,699	0,761
Snorkel (Default LLM)	0,732	0,323	0,800	0,219	0,344
CatBoost (ручная разметка, 1000 примеров)	0,977	0,877	0,800	0,833	0,816
LGIWL + CatBoost	0,982	0,905	0,851	0,792	0,820
LGIWL + RuModernBERT	**0,987**	**0,913**	**0,908**	**0,822**	**0,863**

6.3 Повышение точности как основной фактор роста F1-меры

Ключевым фактором более высокой F1-меры при использовании метода LGIWL является существенное улучшение точности по сравнению с другими подходами. Например, при запуске модели LGIWL на нейронной сети RuModernBERT достигается точность 0.908 при сохранении полноты 0.822. Для сравнения, метод запросов без явного обучения обеспечивает высокую полноту (0.890), но значительно проигрывает в точности (0.677), ограничивая тем самым итоговую F1-меру на уровне 0.769. Это указывает на то, что метод LGIWL не только

эффективно выявляет истинные положительные примеры, но и успешно избегает переизбыточной разметки, благодаря более точным и семантически осмысленным правилам.

6.4 Запросы без явного обучения: высокая полнота, низкая эффективность

Метод запросов без явного обучения (zero-shot prompting) на основе нейронной сети GPT-4 показывает относительно высокую полноту, однако требует значительных трудозатрат на ручную настройку. Разработка качественных подсказок потребовала множества итераций с участием человека, а также специфической настройки под задачу. Несмотря на отсутствие явного обучения, данный подход требует значительных ресурсов человека и вычислительных затрат, при этом обеспечивая лишь среднее итоговое качество. Его низкая точность и ограниченная адаптивность значительно проигрывают автоматизированному, основанному на обратной связи процессу LGIWL.

6.5 Эффект от итеративного уточнения функций разметки

Сравнение одношагового подхода Snorkel (Default LLM) с методом LGIWL иллюстрирует преимущество генерации подсказок по ошибкам классификатора. Функции разметки, сгенерированные за один шаг моделью LLM, показывают значение F1-меры, равное 0.344, тогда как итеративное уточнение правил в LGIWL повышает этот показатель до 0.820 (на модели CatBoost) и 0.863 (на сети RuModernBERT). Улучшение обусловлено не просто количеством правил, а их целенаправленной настройкой под ошибки классификации.

6.6 Ограничения статической разметки по ключевым словам

Статические методы на основе ключевых слов, включая Snorkel (Keywords) и CatBoost на Snorkel (Keywords), уступают методу LGIWL. Даже с использованием дискриминативного классификатора поверх слабых меток, лучшая достигнутая F1-мера не превышает 0.761. Подобные эвристики плохо справляются с перефразированием, многоходовыми контекстами и отрицаниями – ситуациями, которые LGIWL успешно обрабатывает за счёт семантической генерации правил через LLM.

6.7 Выводы

Метод LGIWL представляет собой надёжный и экономичный подход к обучению со слабым контролем. Он превосходит традиционные эвристики и методы запросов без явного обучения LLM по точности, полноте и F1-мере. Итеративная структура с обратной связью позволяет целенаправленно синтезировать и отбирать правила, улучшая обобщающую способность при сниженных затратах. Благодаря высокой точности и полноте, LGIWL обладает значимыми преимуществами для задач реальной классификации, где важны как корректность, так и полнота разметки.

7. Заключение

В данной работе мы представили LGIWL – итеративный подход к обучению со слабым контролем, основанный на генерации, уточнении и фильтрации функций разметки с помощью больших языковых моделей (LLM). Предложенный метод позволяет добиться высокого качества классификации в условиях ограниченного объёма размеченных данных, не прибегая к масштабной ручной аннотации. В отличие от подходов без явного обучения, требующих экспертной настройки инструкций и при этом страдающих от недостаточной точности, метод LGIWL использует большие языковые модели не напрямую как классификаторы, а как генераторы семантических правил, интегрированных в итеративный цикл с обратной связью по результатам работы классификатора.

Эксперименты на реальной задаче классификации банковских диалогов показали, что LGIWL существенно превосходит статические подходы обучения со слабым контролем и методы без явного обучения, достигая F1-меры 0,863 в реализации на основе сети RuModernBERT при высокой полноте и заметном росте точности. Полученные результаты сопоставимы с полностью супервизированными моделями, обученными на таком же количестве размеченных данных, и в некоторых случаях превосходят их. Это подтверждает, что стратегическое использование ограниченного набора размеченных данных для уточнения правил может быть эффективнее прямого обучения модели.

В отличие от полностью ручных подходов, таких как подбор ключевых слов или разработка подсказок для языковых моделей, LGIWL требует минимальной аннотации, направленной исключительно на фильтрацию функций разметки по ошибкам классификатора. Метод автоматизирует выявление выразительных и высокопокрывающих правил, многие из которых отражают нетривиальные семантические закономерности, трудные для ручного кодирования.

В будущем планируется расширение LGIWL на многоклассовые задачи и сценарии междоменного переноса знаний, а также интеграция более продвинутых стратегий генерации подсказок (например, цепочек рассуждений и саморефлексии) в цикл создания функций разметки. Также планируется формализация критериев сходимости при итеративном слабом контроле с использованием LLM и изучение возможностей интеграции сигналов активного обучения в контур обратной связи.

Таким образом, LGIWL представляет собой масштабируемую, интерпретируемую и экономичную альтернативу как ручной разметке данных, так и непрозрачным zero-shot подходам, двигаясь в направлении автоматизированного, но управляемого человеком применения языковых моделей в реальных задачах машинного обучения.

Список литературы / References

[1]. Stephen H Bach, Ben He, Alexander Ratner, and Christopher Ré. Learning the structure of generative models without labeled data. In International Conference on Machine Learning, pages 273–282, 2017.
[2]. Stephen H Bach, Daniel Rodriguez, Yintao Liu, et al. Snorkel drybell: A case study in deploying weak supervision at industrial scale. In ACM SIGMOD, pages 362–375, 2019.
[3]. Bradley Denham et al. Witan: Unsupervised labeling function generation for assisted data programming. Proceedings of the VLDB Endowment, 15(11): 2334–2347, 2022.
[4]. Nan Guan et al. Datasculpt: Cost-efficient label function design via prompting large language models. In EDBT, pages 226–237, 2025.
[5]. Tai-Hsuan Huang et al. Scriptoriumws: A code generation assistant for weak supervision. In ICLR Workshop, 2023. arXiv:2301.01229.
[6]. Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision.
[7]. Robert Smith et al. Language models in the loop: Incorporating prompting into weak supervision. Journal of Data Science, 1(2):1–30, 2022.
[8]. Paroma Varma and Christopher Ré. Snuba: Automating weak supervision to label training data. In VLDB Endowment, volume 12, pages 223–236, 2018.
[9]. Peng Yu and Stephen H Bach. Alfred: A system for prompted weak supervision. In ACL System Demonstrations, pages 479–488, 2023.
[10]. Jialu Zhang et al. Wrench: A comprehensive benchmark for weak supervision. NeurIPS Datasets and Benchmarks, 2021.
[11]. Ruixiang Zhang et al. Prboost: Prompt-based rule discovery and boosting for interactive weakly-supervised learning. In ACL, pages 745–758, 2022.

Информация об авторах / Information about authors

Артур Дмитриевич СОСНОВИКОВ – аспирант Института системного программирования с 2023 года. Сфера научных интересов: методы машинного обучения, обучение со слабым контролем.

Artur Dmitrievich SOSNOVIKOV – graduate student at the Institute of System Programming since 2023. Research interests: machine learning methods, weakly supervised learning.

Антон Дмитриевич ЗЕМЕРОВ – старший ML-инженер в банке «Точка». Выпускник Физтех-Школы Прикладной Математики и Информатики МФТИ. Сфера научных интересов: методы машинного обучения, обработка естественного языка, большие языковые модели.

Anton Dmitrievich ZEMEROV – Senior ML Engineer at Tochka Bank. Graduate of the PhysTech School of Applied Mathematics and Informatics at MIPT. Research interests: machine learning methods, natural language processing, large language models.

Денис Юрьевич ТУРДАКОВ – кандидат физико-математических наук, заведующий отделом ИСП РАН, доцент кафедры системного программирования факультета ВМК МГУ. Научные интересы: анализ естественного языка, извлечение информации, обработка больших данных, анализ социальных сетей.

Denis Yurievich TURDAKOV – Cand. Sci. (Phys.-Math.), Head of Department at ISP RAS, associate professor of the Department of System Programming at MSU. Research interests: natural language processing, information extraction, big data analysis, social network analysis.