Записки научных
семинаров ПОМИ
Том 552, 2026 г.

С. И. Николенко

МЕТОДЫ ФИЛЬТРАЦИИ И ОЦЕНКИ ДАННЫХ ДЛЯ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ

§1. Введение

Современные модели машинного обучения, особенно фундаментальные базовые модели и большие языковые модели (large language models, LLM), критически ограничены своими обучающими данными. Исследования законов масштабирования показывают, что при фиксированной архитектуре и алгоритме оптимизации улучшения в объёме, качестве и составе обучающего корпуса часто оказываются гораздо важнее преимуществ, связанных с размером модели. Тем не менее, большинство исследовательских и практических проектов в машинном обучении по-прежнему рассматривают выбор данных в основном как этап грубой предварительной фильтрации: эвристики на основе правил, уровня ошибки или неопределённости, списки разрешённых доменов и классификаторы качества применяются к огромным выборкам данных (например, к Common Crawl) перед предобучением.

Настоящий обзор посвящён систематическому рассмотрению методов фильтрации и оценки обучающих данных для больших языковых моделей. Мы охватываем широкий спектр подходов: от простых детерминированных правил до сложных методов на основе моделей, от статических оценок качества до адаптивных стратегий отбора. Эти методы полезны для удаления очевидного шума и повышения общего качества обучающих корпусов, однако они не отвечают на центральный причинно-следственный вопрос: *какие конкретные обучающие примеры важны для заданного поведения обученной модели в дальнейшем*?

Ключевые слова: большие языковые модели, фильтрация данных, подбор данных.

Работа выполнена при финансовой поддержке Министерства науки и высшего образования Российской Федерации (соглашение № 075-15-2025-344 от 29.04.2025 в Санкт-Петербургском международном математическом институте имени Леонарда Эйлера, ПОМИ РАН).

Для ответа на этот вопрос существует отдельный класс методов — функции влияния, которые присваивают значения обучающим примерам на основе их контрфактического влияния на параметры и выходы модели. Функциям влияния посвящена вторая часть настоящего обзора. В данной же части мы сосредоточимся на методах, которые оценивают данные без явного моделирования причинно-следственных связей между обучающими примерами и поведением модели.

Структура обзора. Настоящий обзор организован следующим образом. В разделе 2 мы рассматриваем методы фильтрации на основе правил — старейший и до сих пор широко используемый класс методов, применяющий детерминированные эвристики для грубой очистки данных. В разделе 3 мы переходим к методам фильтрации на основе данных-образцов, включая выбор по сходству с референтными корпусами, дообученные селекторы, методы на основе энтропии и функции потерь, а также подходы, основанные на оценках качества с использованием LLM-судей и классификаторов. В заключении мы синтезируем представленный материал, обсуждаем сильные стороны и ограничения различных подходов и намечаем связь с функциями влияния, рассматриваемыми во второй части обзора.

Наша цель — предоставить практикам систематический обзор доступных инструментов для работы с обучающими данными LLM, помогая выбрать подходящие методы для конкретных задач предварительной фильтрации, доменной адаптации и оценки качества данных.

§2. Фильтрация данных на основе правил

Методы фильтрации на основе правил представляют собой одно из старейших и наиболее широко используемых семейств методов отбора данных для предобучения больших языковых моделей (LLM). Основная идея заключается в применении детерминированных, разработанных человеком правил к необработанным веб-данным с целью удаления явно некачественных или вредоносных образцов, снижения уровня шума и увеличения доли релевантного, грамматически правильного и безопасного для обучения модели текста. Этот подход особенно привлекателен на ранних этапах разработки, когда огромный объем данных (сотни миллиардов токенов) делает более дорогостоящую оценку нецелесообразной.

2.1. Параметры фильтрации. Типичные системы на основе правил включают несколько параметров фильтрации, например:

(1) *источник данных* — ограничение входных данных определенным набором доменов (например, Википедия, новостные сайты) или исключение известных источников низкого качества (домены спама, фермы ссылок);
(2) *длина документа* — удаление очень коротких документов (например, содержащих менее 20 токенов) или слишком длинных документов, превышающих практический предел обработки;
(3) *идентификация языка* — сохранение только документов, чей показатель превышает пороговое значение вероятности $p_{\text{lang}} > \tau$ для целевого языка, где p_{lang} вычисляется с помощью быстрого классификатора, такого как `langid.py`;
(4) *удаление шаблонного кода* — удаление навигационных HTML-меню, повторяющихся колонтитулов, рекламных объявлений и шаблонов с использованием библиотек, таких как Boilerpipe;
(5) *дедупликация* — обнаружение почти дублирующихся фрагментов с помощью алгоритмов MinHash или SimHash; математически "почти дубликаты" определяются как пары со сходством по Жаккару по n-граммам токенов (shingles), превышающим пороговое значение θ;
(6) *токсичность и наличие ненормативной лексики* — удаление документов, чья оценка токсичности $t(x)$ из классификатора (например, Perspective API) превышает пороговое значение τ_t;
(7) *перплексия* — вычисление перплексии (функции ошибки предсказания) при помощи небольшой высококачественной LLM и отбрасывание документов с перплексией, выходящей за пределы желаемого диапазона, что позволяет отфильтровать как бессмысленный, так и тривиально предсказуемый шаблонный текст.

Несколько известных решений для подготовки корпусов для больших языковых моделей полагаются на обширные наборы правил фильтрации:

- C4 [25] применяет идентификатор языка, ограничения по длине и дедупликацию к *Common Crawl*, а также простую эвристику для удаления контента с высоким содержанием стоп-слов;

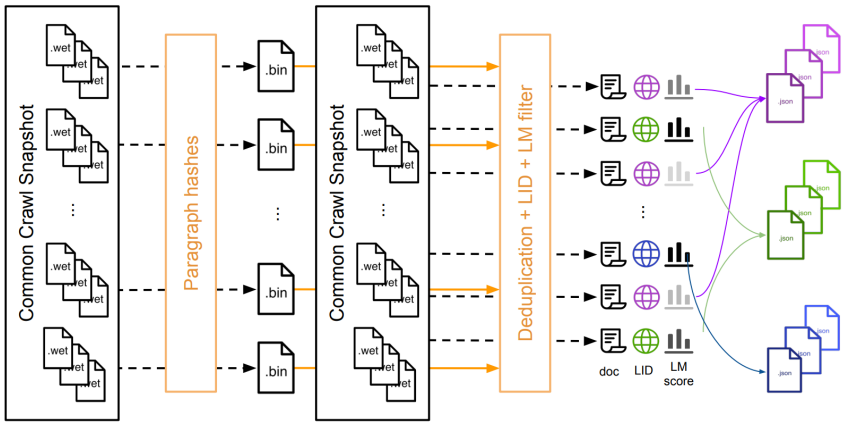

Рис. 1. Общий конвейер CCNet [37].

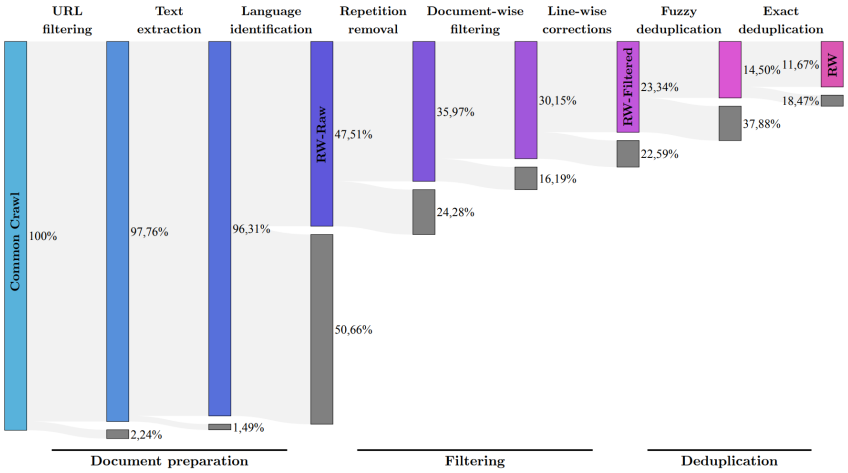

Рис. 2. Результаты фильтрации *RefinedWeb* [22].

- *CCNet* [37] кластеризует и дедуплицирует контент *Common Crawl*, используя алгоритм *MinHash*, и фильтрует по перплексии LLM и показателям сходства с *Wikipedia* (рисунок 1);

- *RefinedWeb* [22] реализует усовершенствованный вариант правил в стиле C4 с более агрессивной дедупликацией и фильтрацией спама (рисунок 2 показывает результаты применения их фильтров);
- ROOTS [11], многоязычный корпус *BigScience*, объединяет списки разрешённых доменов, проверки токенизации для конкретных языков и разработанные вручную правила для контента более чем для 50 языков;
- масштабные промышленные наборы правил, например *Gopher* [24], *Dolma* [31] или *FineWeb* [20], используют настраиваемые фильтры токсичности, детекторы шаблонов и классификаторы типов контента.

2.2. Преимущества и ограничения. Фильтрация на основе правил обеспечивает:

- *простоту и скорость* — правила быстро применяются и легко реализуются в больших масштабах;
- *прозрачность* — решения легко интерпретируются, ведь логика фильтра ясна;
- *значительный выигрыш на ранних этапах* — правила могут устранить заведомый шум до применения дорогостоящих методов оценки.

Однако у неё есть и существенные недостатки:

- *отсутствие адаптивности* — правила не адаптируются к реальным потребностям целевой модели или динамике обучения;
- *потеря информации* — агрессивные методы фильтрации рискуют отбросить редкие, но ценные данные;
- *отсутствие оценки полезности на основе примеров* — правила не могут количественно оценить влияние конкретного примера или подмножества данных на предсказания модели.

Таким образом, методы фильтрации на основе правил, несомненно, полезны для того, чтобы готовить наборы данных для обучения, и отказываться от них не нужно. Однако они не могут помочь с задачей *атрибуции данных* и в целом с более детальной оценкой влияния данных на результаты работы модели.

§3. Фильтрация на основе данных-образцов

3.1. Идея метода. Методы фильтрации на основе данных-образцов (reference data) выбирают обучающие данные, сравнивая их с одним или несколькими образцовыми корпусами (reference corpora). Образцовым корпусом может быть:

(1) или высококачественный *золотой корпус* (например, Википедия или проверенный датасет учебников),
(2) или *корпус целевой области*, представляющий предполагаемое распределение задач, которые в будущем будет решать обучаемая модель (например, корпус медицинских или юридических текстов).

Формально, пусть $\mathcal{X}_{\text{ref}}$ обозначает референтный набор, а $\mathcal{X}_{\text{cand}}$ — набор кандидатов в данные для обучения. Функция оценки $s : \mathcal{X}_{\text{cand}} \to \mathbb{R}$ присваивает каждому кандидату оценку релевантности на основе его сходства с $\mathcal{X}_{\text{ref}}$, и выбираются k лучших кандидатов.

3.2. Сходство с высококачественными корпусами. Первый вариант использует фиксированный статический корпус–образец (например, *Wikipedia*) в качестве ориентира для "хорошего" текста. Главный вопрос здесь — как именно считать похожесть между корпусами текстов. Распространённые методы оценки похожести включают в себя:

- *метрики на основе "мешка слов"* (bag-of-words), в частности метрики TF–IDF или BM25 между кандидатами и образцами данных [26, 27];
- *сходство вложений* (embeddings), т.е. косинусную похожесть между векторными представлениями предложений из различных современных моделей, например SBERT;
- *фильтрация по перплексии*: вычисление перплексии кандидатов небольшой LLM, обученной на образцовом наборе $\mathcal{X}_{\text{ref}}$, и выбор документов, перплексия которых попадает в заданный диапазон $[p_{\min}, p_{\max}]$.

Например, CCNet [37] сохраняет страницы *Common Crawl* с низким уровнем перплексии согласно предсказаниям LLM, обученной на основе *Wikipedia*, что зачастую означает стилистическое и тематическое сходство с документами из этого корпуса.

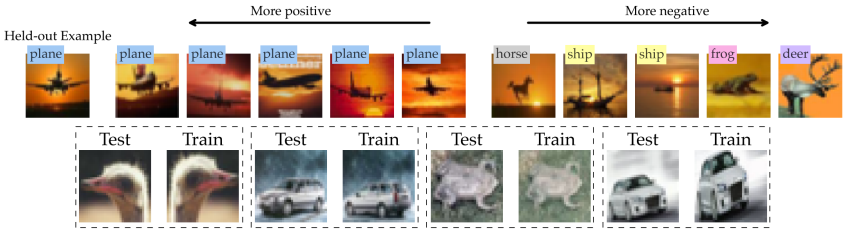

Рис. 3. Модели данных, используемые для поиска утечек данных [7].

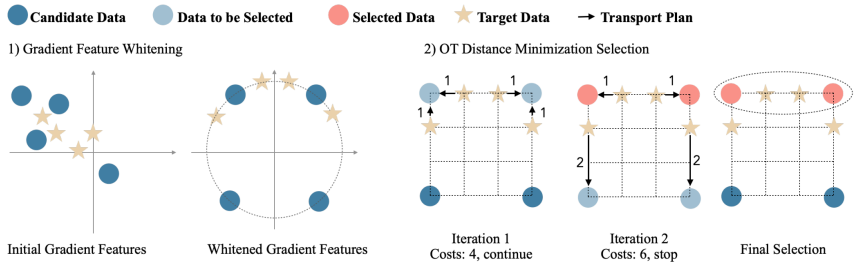

Рис. 4. Обзор TAPO [3].

3.3. Сходство с корпусами из целевой области. Здесь $\mathcal{X}_{\text{ref}}$ – это набор данных, специфичный для конкретной задачи или предметной области (например, медицинские записи или юридические документы). Документы-кандидаты оцениваются по:

- *разнице кросс-энтропии* [1]: $s(x) = H_{\text{in}}(x) - H_{\text{out}}(x)$, где H_{in} – перекрёстная энтропия при внутридоменной языковой модели, а H_{out} – при LLM, обученной не на целевом домене (out of domain);
- *модели данных* (datamodels) [7]: этот метод обучает несколько линейных предикторов на случайных подмножествах обучающего набора для прогнозирования выходов целевой модели; затем обученная "модель данных" используется для присвоения оценок значений примерам; например, на рисунке 3 показано, как модели данных можно использовать для поиска утечек данных из обучающего набора в тестовый;

- *LLM-as-a-judge*, например метод *Ask-LLM* [28], используют мощную LLM, которой предлагается оценить полезность выборки для целевой задачи;
- *автоматизированные конвейеры поиска*, например TAROT [3], извлекают данные-кандидаты путём расширения запросов, начиная из некоторых запросов, заведомо принадлежащих целевой области, а затем осуществляют их переранжирование с использованием обученых моделей релевантности; см. иллюстрацию метода TAROT на рисунке 4.

3.4. Преимущества и ограничения. Методы, основанные на образцах данных, интуитивно понятны, не требуют вычисления градиентов и могут быть реализованы до обучения.

Однако:

- они оптимизируют *сходство*, а не *полезность для обучения*; выборка, очень похожая на $\mathcal{X}_{\text{ref}}$, может быть избыточной, если модель уже обучалась на этом домене или на этом корпусе-образце;
- они *не являются адаптивными*, т.е. игнорируют специфику модели; выборка статична при заданном $\mathcal{X}_{\text{ref}}$ и не зависит от модели;
- они могут *сместить модель* в сторону стиля образцовых данных, что потенциально может негативно сказаться на её обобщении за пределами корпуса-образца.

3.5. Выбор данных на основе моделей. Методы выбора данных на основе моделей оценивают кандидатов в обучающие данные с использованием *модели*, которой может быть сама целевая модель на ранней стадии обучения, дообученная прокси-модель или отдельная лёгкая модель-оценщик. В отличие от выбора на основе данных-образцов, эти подходы используют информацию, специфичную для модели (например, градиенты, внутренние оценки-логиты, внутренние состояния) для оценки полезности данных. В этом семействе можно выделить несколько весьма разнообразных подтипов.

Дообученные LLM-оценщики. Некоторые методы явным образом *обучают* отдельную модель для предсказания полезности примеров

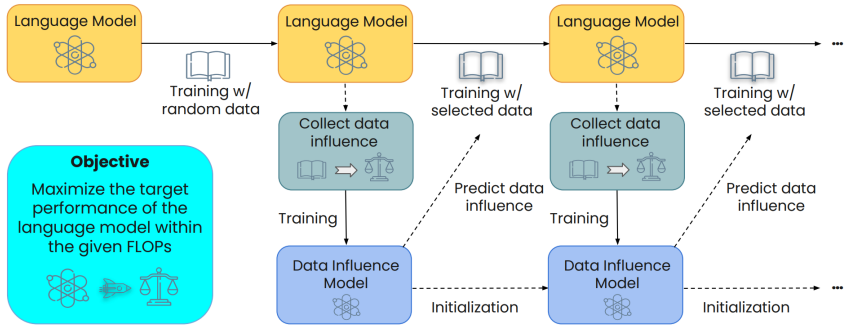

Рис. 5. Обзор подхода MATES [43].

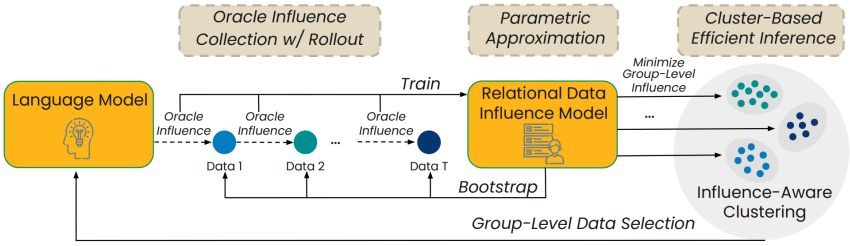

Рис. 6. Обзор подхода Group-MATES [44].

для целевой задачи. Например, MATES [43] и Group-MATES [44] обучают маленькую LLM для оценки примеров для предобучения, используя метки от внутренней метрики полезности модели-учителя; см. рисунки 5 и 6 для обзора этих методов.

Подход ALinFIK, обсуждаемый во второй части обзора, также принадлежит к этому классу подходов: небольшая модель быстро предсказывает оценки для всего пула кандидатов с минимальным использованием памяти GPU и хранилища. В общем случае эти методы могут амортизировать дорогостоящую оценку, обучаясь один раз на подвыборке, а затем применяя лёгкий оценщик ко всем данным.

Оценка на основе референтной модели. Вместо дистилляции из меток в стиле функций влияния некоторые методы оценивают кандидатов с помощью отдельной референтной модели:

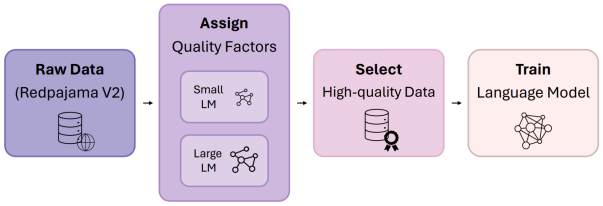

Рис. 7. Обзор подхода ScalingFilter [13].

- разреженные автокодировщики (sparse autoencoders, SAE) [10, 16, 18] обучают компактную специализированную сеть для аппроксимации функции предпочтений целевой модели;
- *ScalingFilter* [13] фильтрует данные на основе эвристик, выведенных из законов масштабирования, связывающих потери при обучении с размером датасета (рисунок 7);
- PDPC [45] использует предобученный классификатор доменов для оценки релевантности домена.

Хотя эти методы быстрее, чем использование полной целевой модели, их оценки могут не идеально согласовываться с внутренним ландшафтом полезности целевой модели.

Зондирование в контексте (in-context probing). Другие подходы опрашивают саму целевую модель в режиме zero-shot или few-shot для оценки пробелов в знаниях:

- *Nuggets* [15] определяет атомарные "наггеты" знаний и выбирает данные для покрытия отсутствующих наггетов, выявленных с помощью зондирования; рисунок 8 показывает их подход к обучению по одному примеру (one-shot; см. рисунок 8);
- RICo [40] нацелен на рассуждающие модели и зондирует отсутствующие подтипы задач (рисунок 9);
- *Relation to Influence Function* [8] связан с нашей основной темой и исследует концептуальные пересечения с IF, но остаётся основанным на зондировании, а не на градиентах, поэтому мы помещаем его сюда.

Такие методы обеспечивают интерпретируемое покрытие таксономий способностей, но сильно зависят от дизайна зондов.

Другие стратегии на основе обучения. Методы, не вошедшие в классификацию выше, включают, например:

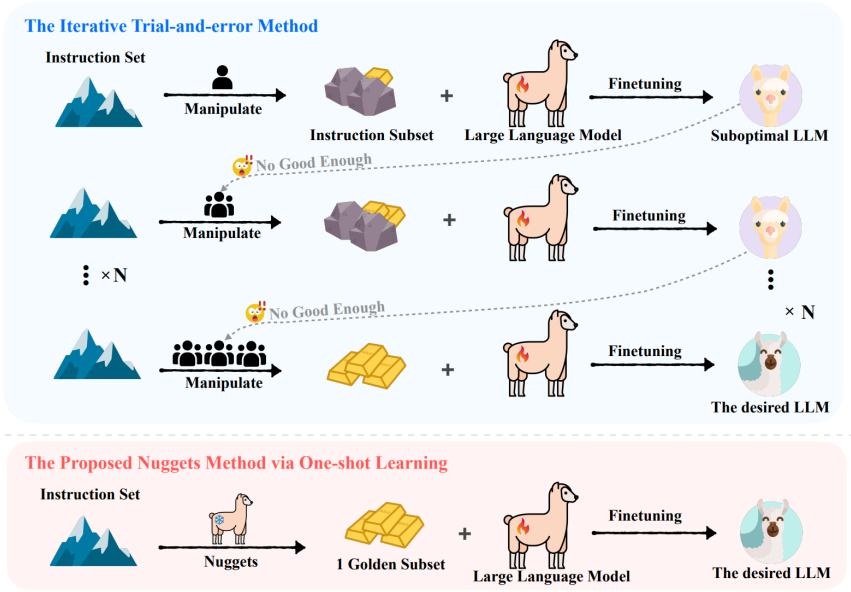

Рис. 8. Как *Nuggets* делает выбор данных более эффективным [15].

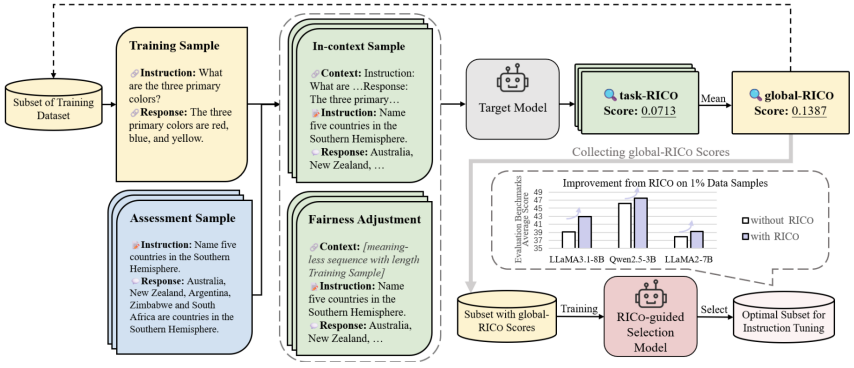

Рис. 9. Обзор подхода RICo [40].

- DSDM [2], который обучает “модели данных” (datamodels) [7] для выбора примеров, наиболее точно предсказывающих выходы модели;

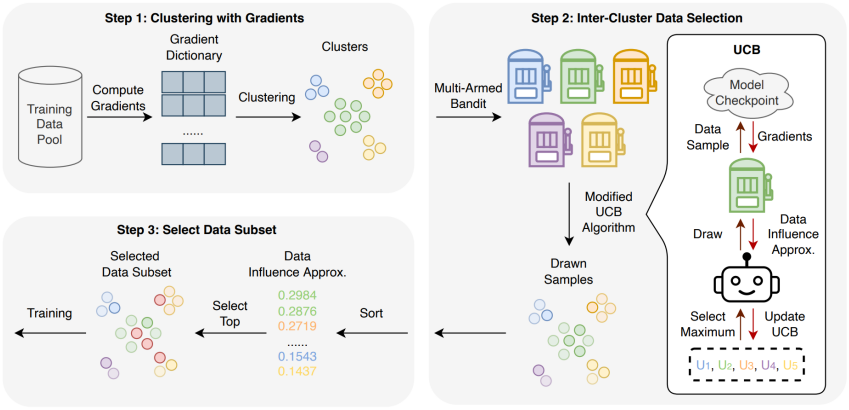

Рис. 10. Обзор метода *ClusterUCB* [36].

- *ClusterUCB* [36], где задача выбора данных формулируется как задача контекстуального бандита над кластерами примеров (рисунок 10);
- NICE [33], который предлагает эвристики нормализации и улучшения для оценки на уровне корпуса;
- *Extended Core-Set* [4], который расширяет выбор ключевых подмножеств данных (coresets), основанный на метриках разнообразия, дополнительными ограничениями.

Преимущества и недостатки. Методы выбора данных на основе моделей могут:

- использовать сигналы, специфичные для модели (значения функции потерь, градиенты, активации), что делает их адаптивными;
- интегрировать информацию о downstream-задаче непосредственно в оценку.

Однако:

- прокси-модели или дистиллированные модели-оценщики могут оказаться несогласованными с истинной целевой моделью;
- оценщики, обученные по данным одной стадии обучения, могут не обобщаться на другие фазы обучения;

- многие методы оптимизируют эффективность выбора, а не точность атрибуции.

Таким образом, дистиллированные оценщики отлично подходят для *быстрого выбора*, и такие методы, как ALinFIK, действительно представляются важными и в контексте функций значения. Но они полагаются на предвычисленные метки от более точного, но и более дорогого метода. Для атрибуции же оценку влияния было бы предпочтительно вычислять *напрямую* из градиентов и кривизны обученной модели, как в классических или масштабируемых функциях влияния [5, 9, 19] (см. вторую часть обзора), обеспечивая принципиальную связь с leave-one-out оценками и позволяя проводить анализ по каждому выходу и каждому примеру. Кроме того, масштабируемые функции влияния (например, проецированные градиенты, LOGIX/LoGra) могут превратить это в легко масштабируемую задачу поиска, сохраняя при этом чёткую статистическую интерпретацию, связанную с геометрией гессианов и матриц Фишера.

3.6. Методы отбора на основе энтропии и функции потерь. Методы, основанные на энтропии и функции потерь, отбирают примеры на основе *неопределённости* или *сложности* их предсказания моделью. Основное предположение заключается в том, что примеры с высокими значениями функции потерь или энтропии обеспечивают более сильный обучающий сигнал.

Базовая формулировка. Пусть $\ell(x;\theta)$ обозначает функцию потерь модели θ на примере x, а $p_\theta(y \mid x)$ – распределение предсказаний. Тогда основные оценки имеют вид:

- *на основе функции потерь* $s_{\text{loss}}(x) = \ell(x;\theta)$, часто (минус) логарифм правдоподобия;
- *на основе энтропии* $s_{\text{ent}}(x) = -\sum_y p_\theta(y \mid x) \log p_\theta(y \mid x)$.

Выбираются примеры, для которых оценка s превышает пороговое значение τ.

Характерные примеры методов. Во-первых, можно осуществлять *простую фильтрацию*, т.е. сохранять примеры с оценками ниже/выше (в зависимости от формулировки) порогового значения для удаления легко предсказуемых или вырожденных текстов. Однако существует несколько подходов, которые развивают эту идею дальше, в частности:

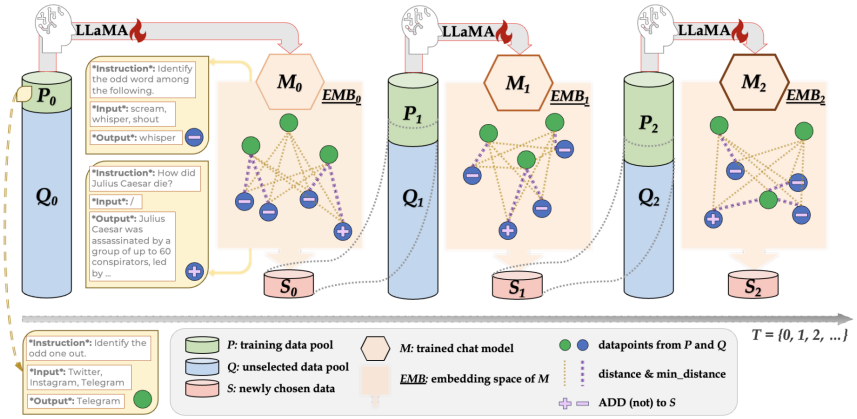

Рис. 11. Метод *DiverseEvol* [39].

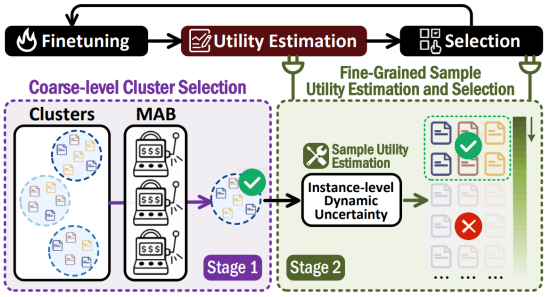

Рис. 12. Обзор метода LEAD [17].

- *DiverseEvol* [39] сочетает оценку на основе энтропии с эволюционным отбором подмножеств для обеспечения разнообразия (рисунок 11);
- ZIP [41] интегрирует априорные значимости без примеров (zero-shot) в фильтрацию на основе перплексии;
- LEAD [17] применяет многоэтапную фильтрацию, используя траектории потерь от раннего к позднему этапу обучения (рисунок 12);

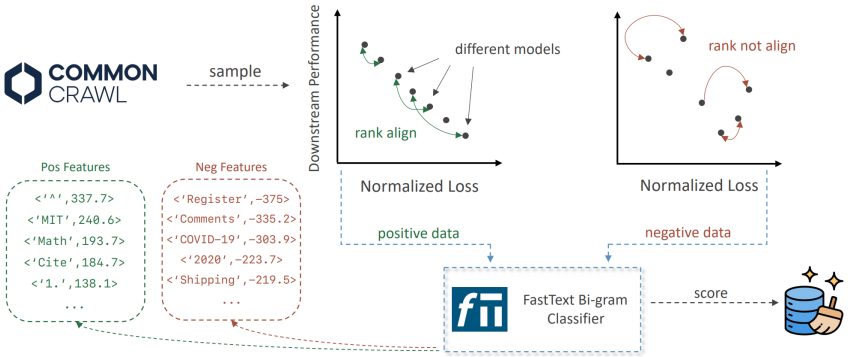

Рис. 13. Предсказательный выбор данных методом PreSelect [30].

- *PreSelect* [30] строит многокритериальный процесс отбора, сочетающий функцию потерь, нормы градиентов и ограничения на разнообразие (рисунок 13);
- *корреляции перплексии* [32] изучают, как перплексия коррелирует с качеством выполнения последующих задач, что помогает выбирать пороговые значения.

Преимущества и недостатки. Эти методы обладают следующими достоинствами:

- *простота и эффективность*, поскольку требуют только прямых проходов (или логированных значений функции потерь) по кандидатам;
- *адаптивность*, так как функция потерь вычисляется относительно конкретной модели, учитывая её текущие знания.

Однако они также имеют тенденцию к следующим проблемам:

- излишний отбор зашумленных или состязательно сложных примеров, которые могут навредить обучению;
- смешение понятий “сложный” и “полезный”; пример может иметь высокую функцию потерь из-за того, что он находится вне распределения данных и нерелевантен, но метрики на основе энтропии будут специально отбирать именно такой пример, тем самым усугубляя проблему;

- отсутствие контрфактических гарантий, т.е. такие методы не могут предсказать, как удаление примера изменит качество модели.

Стоит особо отметить разницу между этими подходами и методами на основе функций влияния, которые обсуждаются во второй части обзора. Эти методы полагаются исключительно на величину функции потерь или энтропии как на прокси для полезности примера. Высокая функция потерь может означать как действительно ценную информацию, которую модель ещё не усвоила, так и просто шум или нерелевантные данные.

Функции же влияния уточняют идею, основанную на функции потерь, взвешивая обучающий градиент g_z обратным гессианом, что даёт

$$\mathrm{IF}(z \to z_{\mathrm{test}}) = -g_{\mathrm{test}}^{\top} H^{-1} g_z,$$

которое дисконтирует точки с высокой функцией потерь, градиенты которых не совпадают с улучшением тестовой функции цели. Этот направленный сигнал позволяет избежать чрезмерного акцента на нерелевантных "сложных" примерах и обеспечивает принципиальную аппроксимацию оценок leave-one-out. Иными словами, функции влияния не просто измеряют, насколько сложен пример для модели, но оценивают, насколько его включение или исключение из обучающего набора повлияет на производительность модели на конкретных тестовых примерах, что даёт гораздо более точную и полезную информацию для атрибуции данных.

Таким образом, в отличие от методов на основе энтропии и значения функций потерь, методы на основе функций влияния используют не только величину градиента, но и его направление, а также кривизну пространства параметров через обратный гессиан. Это позволяет различать случаи, когда пример сложен, но полезен (его градиент совпадает с желаемым направлением обучения), и случаи, когда пример сложен, но вреден (его градиент направлен в нежелательном направлении).

3.7. Методы оценки на основе суждений на естественном языке и классификаторов качества. Это семейство методов использует либо написанные людьми рубрикаторы, либо основанные на

моделях "судьи" для явной оценки *качества*, *полезности*, *безвредности* или других атрибутов обучающих примеров-кандидатов. Функция оценки обычно обучается или промптируется для выдачи категориальной оценки (например, высокое/среднее/низкое качество) или числовой оценки в интервале $[0, 1]$.

Выбор данных на основе начального набора и вручную созданные конвейеры. Одна ветвь таких методов начинается с *начального набора* (seed set) проверенных высококачественных примеров вместе с явной политикой или алгоритмом для принятия решений. Этот алгоритм затем применяется – вручную или с простой автоматизацией – к новым данным-кандидатам. Важные примеры таких методов включают:

- ROOTS [11], который включает предоставленные сообществом высококачественные начальные данные для многоязычных веб-данных;
- системы рейтинга на основе правил, такие как DPP rule-based rating [14], в которых созданные вручную функции оценки (например, для длины, структуры и токсичности) ранжируют примеры.

Такие подходы часто требуют значительного человеческого надзора и экспертизы в предметной области, но могут установить высокую базовую планку качества. Преимущество таких подходов в том, что они опираются на человеческое понимание качества и могут учитывать тонкие нюансы, которые сложно формализовать. Однако их масштабирование ограничено доступностью экспертов и временем, необходимым для ручной проверки.

Подходы "LLM-как-судья" (LLM-as-a-judge). Другое подсемейство использует большие языковые модели, которым даются инструкции и примеры для оценки текстов-кандидатов согласно рубрикатору качества. Вот примеры важных работ в этом направлении:

- **QuRating** [38] промптирует модель класса GPT для присвоения холистических оценок качества фрагментам текста (рисунок 14);
- **OpenCSG** [42] применяет LLM-судью для оценки качества фрагментов кода;
- **CCI3.0-HQ** [34] использует проверенные людьми высококачественные примеры для дообучения LLM-судьи;
- **FineWeb-Edu** [21] строит рубрикатор специально для образовательного контента;

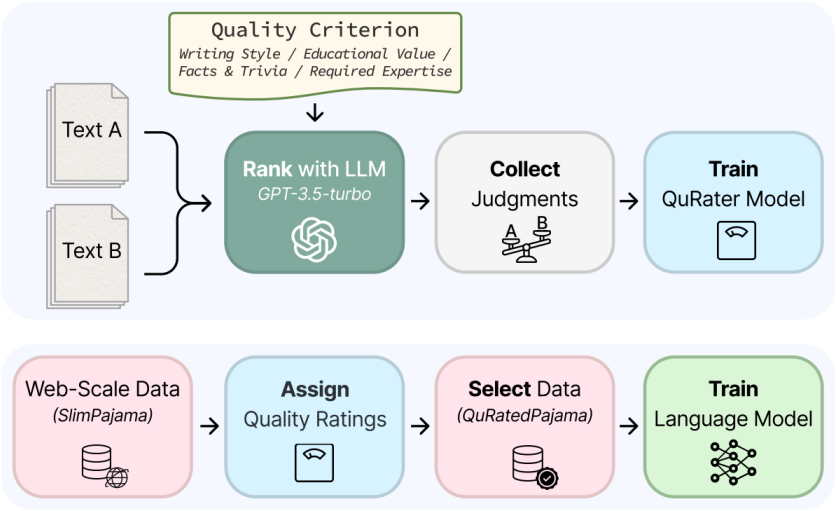

Рис. 14. Как *QuRating* сравнивает качество текстовых фрагментов [38].

- **Ask-LLM** [28] напрямую спрашивает LLM, принесёт ли пример пользу целевой задаче.

Такие методы могут учитывать тонкие, высокоуровневые критерии (стиль, фактическая точность), которые сложно закодировать в статических правилах. Ключевое преимущество здесь в том, что LLM могут обобщать понятие качества на основе обширных предобучающих данных и улавливать паттерны, недоступные простым эвристикам. Однако стоимость использования крупных LLM для оценки миллиардов примеров может быть запретительно высокой, что приводит к необходимости поиска более эффективных альтернатив.

Классификаторы на основе небольших моделей и программные метки. Когда стоимость или масштаб не позволяют использовать LLM-судью, можно обучить меньшие классификаторы (например fastText, линейные модели, BERT-base) на размеченных примерах для имитации человеческих или порождённых LLM суждений:

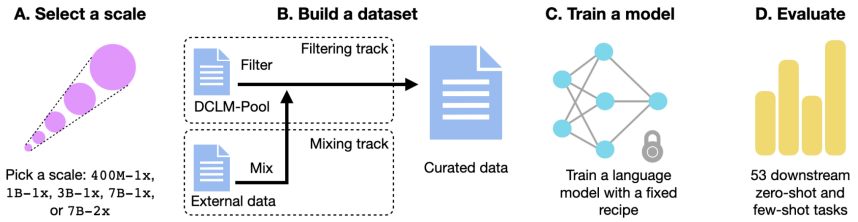

Рис. 15. Конвейер *DataComp-LM* [12]

- *DataComp-LM* [12] – это представительный бенчмарк, где различные классификаторы соревнуются на стандартных задачах отбора данных (рисунок 15);
- существуют специализированные конвейеры для предметных областей, такие как *DeepSeekMath* [29] и *DeepSeek-Coder* [6], которые отбирают высококачественные данные для математических рассуждений и программирования соответственно;
- *Ultra-FineWeb* [35] использует слабый сигнал обучения и эвристики для обучения масштабируемых классификаторов качества;
- фреймворки автоматизации, такие как *AutoDS* [46] и *DataMan* [23], оптимизируют цикл определения стратегии, разметки и дообучения моделей качества.

Такие системы могут обрабатывать миллиарды документов после обучения, при этом стоимость составляет лишь малую долю от стоимости использования LLM-судьи. Важно отметить, что небольшие классификаторы представляют собой компромисс между точностью оценки и вычислительной эффективностью: они не могут уловить все нюансы, которые видит человек или большая LLM, но их можно применять к огромным объёмам данных с приемлемыми затратами.

Преимущества и недостатки. Конвейеры оценки качества на основе LLM или более эффективных аппроксимаций LLM-суждений превосходно справляются с:

- улавливанием сложных, субъективных представлений о качестве за пределами поверхностного сходства или перплексии;
- обеспечением согласования с человеческими предпочтениями и руководствами по безопасности.

Однако они:

- производят статические оценки качества, которые не зависят от текущих знаний целевой модели;
- не предоставляют контрфактической связи между конкретным обучающим примером и конкретным выходом модели;
- могут вносить смещение (bias) из-за собственных ограничений судьи, формулировки промпта или обучающих данных.

Первый пункт особенно важен: оценка качества, полученная в начале обучения, может существенно отличаться от того, что действительно нужно модели на более поздних этапах. Пример, который выглядит высококачественным сам по себе, может оказаться избыточным, если модель уже усвоила подобную информацию. Аналогично, пример с умеренной оценкой качества может быть чрезвычайно ценным, если он заполняет пробел в знаниях модели. Кроме того, эти методы не могут ответить на главный каузальный вопрос: "Что произойдёт с производительностью модели, если мы удалим этот конкретный пример из обучающего набора?"

Наша основная цель в этом обзоре – *атрибуция данных*, т.е. ответ на вопрос, какие обучающие примеры больше всего повлияли на данный конкретный выход модели. Суждения о качестве, будь то от людей, LLM или небольших классификаторов, не оценивают *предельную полезность* примера в контексте фактических параметров модели. Функции влияния, напротив, вычисляют аппроксимацию первого порядка эффекта от увеличения веса примера на выбранную тестовую функцию потерь, давая принципиальный, специфичный для модели и выхода сигнал атрибуции.

Иными словами, методы оценки качества отвечают на вопрос "хорош ли этот пример сам по себе?", в то время как функции влияния отвечают на вопрос "как этот пример повлиял на способность модели обрабатывать данный конкретный тестовый случай?". Это фундаментальное различие делает функции влияния незаменимым инструментом для задач атрибуции данных, отладки моделей и понимания того, какие именно части обучающего набора ответственны за конкретное поведение обученной модели.

§4. Заключение

В настоящем обзоре мы систематически рассмотрели методы фильтрации и оценки обучающих данных для больших языковых моделей,

охватив широкий спектр подходов от простых эвристик до сложных систем на основе моделей.

Основные выводы. Методы фильтрации на основе правил остаются незаменимым инструментом для первичной обработки огромных веб-корпусов. Их простота, скорость и прозрачность делают их идеальным выбором для удаления очевидного шума – слишком коротких или длинных документов, контента на нецелевых языках, дубликатов и токсичных материалов. Такие системы, как C4, CCNet или RefinedWeb показывают, что грамотно настроенные конвейеры на основе правил могут существенно повысить качество обучающих данных при минимальных вычислительных затратах.

Методы на основе данных-образцов расширяют возможности фильтрации, позволяя отбирать данные по сходству с высококачественными референтными корпусами или целевыми доменами. Метрики на основе перплексии, вложений и кросс-энтропии обеспечивают более тонкий отбор, чем детерминированные правила, хотя и требуют наличия подходящего референтного корпуса.

Методы на основе моделей – дообученные LLM-оценщики, разреженные автокодировщики, системы зондирования в контексте – представляют следующий уровень сложности, используя информацию, специфичную для модели, для оценки полезности данных. Эти методы могут быть адаптивными и учитывать текущее состояние знаний модели, однако их оценки могут не идеально согласовываться с истинной полезностью данных для обучения.

Методы на основе энтропии и функции потерь отбирают примеры по их сложности для модели, предполагая, что трудные примеры обеспечивают более сильный обучающий сигнал. Однако эти методы не различают “сложные, но полезные” и “сложные, но вредные” примеры – пример может иметь высокую функцию потерь просто потому, что он находится вне распределения данных и нерелевантен.

Наконец, методы оценки качества на основе LLM-судей и классификаторов позволяют учитывать сложные, субъективные критерии качества, недоступные для формализации в виде правил. Однако они производят статические оценки, не зависящие от текущих знаний целевой модели.

Ограничения рассмотренных методов. При всём разнообразии рассмотренных подходов все они имеют общее ограничение: они не обеспечивают принципиальной связи с контрфактическими оценками. Иными словами, эти методы не могут ответить на ключевой вопрос: как изменится поведение модели, если мы удалим или изменим конкретный обучающий пример?

Этот вопрос имеет фундаментальное значение для задач атрибуции данных, отладки моделей, аудита и соответствия регуляторным требованиям. Для его решения существует отдельный класс методов – *функции влияния* (influence functions), которые аппроксимируют истинные leave-one-out оценки через локальную линеаризацию и учитывают геометрию пространства параметров через обратный гессиан.

Практические рекомендации. На основе проведённого анализа мы можем дать следующие рекомендации.

(1) Для грубой предварительной фильтрации больших веб-корпусов лучше начать с эвристик на основе правил и простых методов на основе перплексии. Эти методы эффективны, не требуют сложной настройки и позволяют быстро отфильтровать явно низкокачественные данные.
(2) Для доменной адаптации можно использовать методы на основе данных-образцов с референтным корпусом из целевой области. Разница кросс-энтропии и сходство вложений хорошо работают для отбора релевантного контента.
(3) Для оценки качества контента при отсутствии чётких формальных критериев лучше подходят методы на основе LLM-судей. Здесь, однако, следует помнить о возможных смещениях в оценках судьи и о том, что эти оценки статичны.
(4) Для задач, требующих понимания причинно-следственных связей между данными и поведением модели, лучше обратиться к функциям влияния, рассматриваемым во второй части настоящего обзора.

Рассмотренные в настоящей части методы и функции влияния не являются взаимоисключающими – напротив, они естественно дополняют друг друга.

Список литературы

1. A. Axelrod, *Cynical Selection of Language Model Training Data*, ArXiv preprint arXiv:1709.02279 (2017).

2. L. Engstrom, A. Feldmann, and A. Mądry, *DSDM: Model-Aware Dataset Selection with Datamodels*, in: Proceedings of the 41st International Conference on Machine Learning (ICML'24), JMLR.org, 2024.
3. L. Feng, F. Nie, Y. Liu, and A. Alahi, *TAROT: Targeted Data Selection via Optimal Transport*, in: Forty-Second International Conference on Machine Learning, 2025.
4. Y. Feng, P. Xia, B. Van Durme, and J. Sedoc, *Automatic Document Selection for Efficient Encoder Pretraining*, 2022.
5. R. Grosse et al., *Studying Large Language Model Generalization with Influence Functions*, 2023.
6. D. Guo, Q. Zhu, D. Yang, Z. Xie, K. Dong, W. Zhang, G. Chen, X. Bi, Y. Wu, Y. K. Li, F. Luo, Y. Xiong, and W. Liang, *DeepSeek-Coder: When the Large Language Model Meets Programming – The Rise of Code Intelligence*, 2024.
7. A. Ilyas, S. M. Park, L. Engstrom, G. Leclerc, and A. Madry, *Datamodels: Predicting Predictions from Training Data*, 2022.
8. C. Jiao, G. Gao, A. Raghunathan, and C. Xiong, *On the Feasibility of In-Context Probing for Data Attribution*, 2025.
9. P. W. Koh and P. Liang, *Understanding Black-Box Predictions via Influence Functions*, in: Proceedings of the 34th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol. 70, PMLR, 06–11 Aug 2017, pp. 1885–1894.
10. M. Lan, P. Torr, A. Meek, D. Krueger, and F. Barez, *Sparse Autoencoders Reveal Universal Feature Spaces Across Large Language Models*, 2025.
11. H. Laurençon et al., *The BigScience ROOTS Corpus: A 1.6TB Composite Multilingual Dataset*, in: Proceedings of the 36th International Conference on Neural Information Processing Systems (NIPS '22), Curran Associates Inc., 2022.
12. J. Li et al., *DataComp-LM: In Search of the Next Generation of Training Sets for Language Models*, NeurIPS 2024 Datasets and Benchmarks Track, 2024.
13. R. Li, Y. Wei, M. Zhang, N. Yu, H. Hu, and H. Peng, *ScalingFilter: Assessing Data Quality Through Inverse Utilization of Scaling Laws*, 2024.
14. X. Li, M. Gao, Z. Zhang, C. Yue, and H. Hu, *Rule-Based Rating and Selection of LLM Training Data*, ICLR 2025 Workshop on Navigating and Addressing Data Problems for Foundation Models, 2025.
15. Y. Li, B. Hui, X. Xia, J. Yang, M. Yang, L. Zhang, S. Si, L.-H. Chen, J. Liu, T. Liu, F. Huang, and Y. Li, *One-Shot Learning as Instruction Data Prospector for Large Language Models*, in: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Aug. 2024, pp. 4586–4601.
16. T. Lieberum, S. Rajamanoharan, A. Conmy, L. Smith, N. Sonnerat, V. Varma, J. Kramar, A. Dragan, R. Shah, and N. Nanda, *Gemma Scope: Open Sparse Autoencoders Everywhere All at Once on Gemma 2*, in: Proceedings of the 7th BlackboxNLP Workshop, Association for Computational Linguistics, Nov. 2024, pp. 278–300.
17. X. Lin, Y. Qi, Y. Zhu, T. Palpanas, C. Chai, N. Tang, and Y. Luo, *LEAD: Iterative Data Selection for Efficient LLM Instruction Tuning*, 2025.

18. D. Ma, G. Shang, Z. Chen, L. Qin, Y. Luo, L. Pan, S. Fan, L. Chen, and K. Yu, *Task-Specific Data Selection for Instruction Tuning via Monosemantic Neuronal Activations*, 2025.
19. S. M. Park, K. Georgiev, A. Ilyas, G. Leclerc, and A. Madry, *TRAK: Attributing Model Behavior at Scale*, in: Proceedings of the 40th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol. 202, PMLR, 23–29 Jul 2023, pp. 27074–27113.
20. G. Penedo, H. Kydlíček, L. Ben Allal, A. Lozhkov, M. Mitchell, C. Raffel, L. Von Werra, and T. Wolf, *The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale*, NeurIPS 2024 Datasets and Benchmarks Track, 2024.
21. G. Penedo, H. Kydlíček, L. Ben Allal, A. Lozhkov, M. Mitchell, C. Raffel, L. Von Werra, and T. Wolf, *The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale*, in: Proceedings of the 38th International Conference on Neural Information Processing Systems (NIPS '24), Curran Associates Inc., 2025.
22. G. Penedo, Q. Malartic, D. Hesslow, R. Cojocaru, H. Alobeidli, A. Cappelli, B. Pannier, E. Almazrouei, and J. Launay, *The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data Only*, in: Proceedings of the 37th International Conference on Neural Information Processing Systems (NIPS '23), Curran Associates Inc., 2023.
23. R. Peng, K. Yang, Y. Zeng, J. Lin, D. Liu, and J. Zhao, *DataMan: Data Manager for Pre-Training Large Language Models*, 2025.
24. J. W. Rae et al., *Scaling Language Models: Methods, Analysis & Insights from Training Gopher*, ArXiv preprint arXiv:2112.11446 (2021).
25. C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. — J. Mach. Learn. Res. **21** (2020), no. 1.
26. S. Robertson and H. Zaragoza, *The Probabilistic Relevance Framework: BM25 and Beyond*. — Foundations and Trends in Information Retrieval **3** (2009), no. 4, 333–389.
27. S. E. Robertson, S. Walker, M. Hancock-Beaulieu, M. Gatford, and A. Payne, *Okapi at TREC-4*, in: Proceedings of The Fourth Text REtrieval Conference (TREC 1995), NIST Special Publication 500-236, National Institute of Standards and Technology (NIST), 1995.
28. N. Sachdeva, B. Coleman, W.-C. Kang, J. Ni, L. Hong, E. H. Chi, J. Caverlee, J. McAuley, and D. Z. Cheng, *How to Train Data-Efficient LLMs*, 2024.
29. Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. K. Li, Y. Wu, and D. Guo, *DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models*, 2024.
30. K. Shum, Y. Huang, H. Zou, Q. Ding, Y. Liao, X. Chen, Q. Liu, and J. He, *Predictive Data Selection: The Data That Predicts Is the Data That Teaches*, 2025.
31. L. Soldaini et al., *Dolma: An Open Corpus of Three Trillion Tokens for Language Model Pretraining Research*, in: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Aug. 2024, pp. 15725–15788.
32. T. Thrush, C. Potts, and T. Hashimoto, *Improving Pretraining Data Using Perplexity Correlations*, 2025.

33. J. Wang, X. Lin, R. Qiao, P. W. Koh, C.-S. Foo, and B. K. H. Low, *NICE: Non-Differentiable Evaluation Metric-Based Data Selection for Instruction Tuning*, ICLR 2025 Workshop on Navigating and Addressing Data Problems for Foundation Models, 2025.
34. L. Wang, B.-W. Zhang, C. Wu, H. Zhao, X. Shi, S. Gu, J. Li, Q. Ma, T. Pan, and G. Liu, *CCI3.0-HQ: A Large-Scale Chinese Dataset of High Quality Designed for Pre-Training Large Language Models*, 2024.
35. Y. Wang, Z. Fu, J. Cai, P. Tang, H. Lyu, Y. Fang, Z. Zheng, J. Zhou, G. Zeng, C. Xiao, X. Han, and Z. Liu, *Ultra-FineWeb: Efficient Data Filtering and Verification for High-Quality LLM Training Data*, 2025.
36. Z. Wang, Q. Zhu, F. Mi, M. Xu, R. Jin, and W. Yang, *ClusterUCB: Efficient Gradient-Based Data Selection for Targeted Fine-Tuning of LLMs*, 2025.
37. G. Wenzek, M.-A. Lachaux, A. Conneau, V. Chaudhary, F. Guzmán, A. Joulin, and E. Grave, *CCNet: Extracting High Quality Monolingual Datasets from Web Crawl Data*, in: Proceedings of the Twelfth Language Resources and Evaluation Conference, European Language Resources Association, May 2020, pp. 4003–4012.
38. A. Wettig, A. Gupta, S. Malik, and D. Chen, *Qurating: Selecting High-Quality Data for Training Language Models*, ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models, 2024.
39. S. Wu, K. Lu, B. Xu, J. Lin, Q. Su, and C. Zhou, *Self-Evolved Diverse Data Sampling for Efficient Instruction Tuning*, 2023.
40. Y. Yang, Q. Dong, L. Yao, F. Zhu, and Z. Sui, *Rico: Refined In-Context Contribution for Automatic Instruction-Tuning Data Selection*, 2025.
41. M. Yin, C. Wu, Y. Wang, H. Wang, W. Guo, Y. Wang, Y. Liu, R. Tang, D. Lian, and E. Chen, *Entropy Law: The Story Behind Data Compression and LLM Performance*, 2024.
42. Y. Yu, Z. Dai, Z. Wang, W. Wang, R. Chen, and J. Pei, *OpenCSG Chinese Corpus: A Series of High-Quality Chinese Datasets for LLM Training*, 2025.
43. Z. Yu, S. Das, and C. Xiong, *MATES: Model-Aware Data Selection for Efficient Pretraining with Data Influence Models*, NeurIPS 2024, 2024.
44. Z. Yu, F. Peng, J. Lei, A. Overwijk, W.-t. Yih, and C. Xiong, *Group-Level Data Selection for Efficient Pretraining*, 2025.
45. X. Zhang, L. Xu, F. Duan, Y. Zhou, S. Wang, R. Weng, J. Wang, and X. Cai, *Preference Curriculum: LLMs Should Always Be Pretrained on Their Preferred Data*, 2025.
46. Y. Zhang, Y. Luo, Y. Yuan, and A. C. Yao, *Autonomous Data Selection with Language Models for Mathematical Texts*, ICLR 2024 Workshop on Navigating and Addressing Data Problems for Foundation Models, 2024.

Nikolenko S. I. Data filtering and evaluation methods for large language models.

Modern large language models critically depend on the quality and composition of their training data. This survey systematically reviews methods for filtering and evaluating training data for LLMs, covering

a broad range of approaches: from deterministic filtering rules (length, language, toxicity constraints, deduplication) to reference-data-based methods (similarity to reference corpora, cross-entropy difference), model-based methods (fine-tuned LLM scorers, sparse autoencoders, in-context probing), entropy- and loss-based methods, as well as approaches relying on LLM judges and quality classifiers. We analyze the strengths and limitations of each class of methods and show that, for all their diversity, they do not provide a principled connection to counterfactual estimates of the influence of individual examples on model behavior. For data attribution tasks, there exists a separate class of methods, namely *influence functions* that we consider in the second part of this survey.

С.Петербургское отделение
Математического института
им. В. А. Стеклова РАН,
наб. р. Фонтанки, 27, 191023,
Санкт-Петербург, Россия
E-mail: sergey@logic.pdmi.ras.ru

Поступило 10 января 2026 г.