

СОДЕРЖАНИЕ

Введение	3
1 ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ	6
1.1 Определение намерений пользователя.....	6
1.2 Методы автоматического машинного обучения	7
1.2.1 H2O.....	7
1.2.2 LightAutoML	9
1.2.3 AutoGluon	9
1.2.4 FEDOT.....	10
1.2.5 Сравнение алгоритмов	10
1.3 Нейросетевые методы представления текста	12
1.3.1 BERT.....	12
1.3.2 Sentence Transformers.....	12
1.4 Методы улучшения и адаптации моделей.....	12
1.4.1 P-Tuning	12
1.4.2 LoRA	12
1.5 Классические методы классификации текста.....	13
1.5.1 Logistic Regression.....	13
1.5.2 ML-KNN	13
1.5.3 DNNC	14
1.5.4 CatBoost	14
1.6 Методы поиска текста.....	14
1.7 Используемые метрики.....	14
1.7.1 Метрики поиска.....	14
1.7.2 Метрики классификации.....	14
1.8 Текстовые аугментации	14
2 ПРОЕКТИРОВАНИЕ.....	15

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

AutoML – автоматическое машинное обучение.

NLP – обработка естественного языка.

NLU – понимание естественного языка.

Эмбе́ддинг – .

Промпт – .

Введение

В последние годы наблюдается бурный рост интереса к диалоговым системам на основе искусственного интеллекта (чат-ботам, голосовым помощникам и так далее). Так, по данным трендов, интерес к голосовым технологиям AI вырос почти втрое за пять лет¹. Диалоговые системы внедряются в бизнес-процессы и повседневную жизнь, однако создание их интеллектуальной части – модели определения намерения пользователя (intent classification) – остается сложной задачей. Такой модуль является ключевым компонентом системы, позволяя автоматически выявлять цель запроса пользователя, но учет специфики различных доменов серьезно затрудняет разработку универсальной модели. Проектирование и тонкая настройка модели интенгов требуют значительных экспертных усилий в области NLP и ML. Поэтому актуальной представляется автоматизация данного процесса – создание универсальных решений, способных уменьшить долю ручной работы и упростить разработку моделей классификации интенгов. Автоматизированные подходы к машинному обучению (AutoML) обещают значительно сократить объем ручного труда за счет автоматического подбора оптимальных моделей и параметров, что особенно важно для быстро растущей области диалоговых систем.

На сегодняшний день для задачи классификации интенгов накоплен внушительный арсенал методов. Традиционно применяются алгоритмы классического машинного обучения, такие как наивный tf-idf[1], а также подходы на основе k-ближайших соседей и ансамблевые методы (например, градиентный бустинг). С развитием глубокого обучения все более широко используются нейросетевые модели, таких как BERT[2], которые достигают высоких показателей качества на задачах. Параллельно развиваются технологии AutoML, автоматизирующие выбор моделей и настройку гиперпараметров. Тем не менее, несмотря на прогресс отдельных компонентов, целостных универсальных AutoML-фреймворков, специально ориентированных на определение интенгов пользователя, предложено немного. Существующие решения зачастую требуют участия эксперта для каждой

¹<https://www.verloop.io/blog/100-best-chatbot-statistics>

новой предметной области, что указывает на необходимость разработать более обобщенный подход.

В связи с этим актуальной является проблема отсутствия универсального, масштабируемого и эффективного AutoML-решения для классификации интенгов, способного автоматически адаптироваться к разным доменам без глубокого участия человека-эксперта.

Цель исследования заключается в разработке такого универсального AutoML-фреймворка, который способен автоматически подбирать оптимальные модели и их конфигурации для классификации интенгов пользователя. Разработанное решение будет протестировано на различных корпусах данных (наборы пользовательских запросов), а его эффективность сопоставлена с результатами ручной настройки моделей, чтобы оценить выигрыш от автоматизации.

Для достижения поставленной цели в работе решены следующие задачи:

1. Провести обзор существующих фреймворков и библиотек, применяемых для построения моделей машинного обучения, включая решения для задач классификации пользовательских намерений;
2. Выполнить анализ современных алгоритмов и подходов к задаче определения пользовательских намерений, включая традиционные методы машинного обучения и нейросетевые архитектуры;
3. Разработать концепцию и архитектуру собственного фреймворка создания моделей машинного обучения для классификации пользовательских намерений;
4. Реализовать программную часть фреймворка с возможностью автоматизированной настройки моделей и выбора признаков;
5. Провести экспериментальное исследование эффективности фреймворка на нескольких датасетах из разных предметных областей;
6. Сравнить результаты, полученные с использованием разработанного фреймворка, с качеством моделей, настроенных вручную, и провести анализ полученных результатов.

Практическая значимость работы состоит в том, что созданный AutoML-фреймворк может быть непосредственно применен при разработке реальных диалоговых систем – чат-ботов, голосовых ассистентов, систем

клиентской поддержки – и других NLP-приложений. Использование такого инструмента позволит ускорить внедрение новых сервисов и снизить порог вхождения для разработчиков за счет автоматизации подбора оптимальной модели под конкретный набор интенгов.

Научная новизна исследования определяется интеграцией современных методов автоматизированного машинного обучения в единой специализированной архитектуре, ориентированной на задачу классификации интенгов. В предлагаемом решении объединяются передовые подходы, включая трансформерные модели и методы обучения с малым количеством примеров, в рамках одного AutoML-фреймворка. Такое сочетание технологий нацелено на достижение высокой точности и устойчивости модели при минимальном ручном вмешательстве, что ранее не было реализовано в полной мере для задачи определения интенгов пользователя.

1 ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Определение намерений пользователя

Классификация намерений – это задача сопоставления высказывания пользователя с предопределенной меткой намерения (семантической категорией цели пользователя). Например, запрос “Какая погода будет завтра?” может быть классифицирован как запрос погоды. Эта способность является ключевым компонентом понимания естественного языка (NLU) в диалоговых системах, позволяя чат-ботам, виртуальным помощникам и другим агентам искусственного интеллекта понимать, чего хочет пользователь, и соответствующим образом реагировать. Классификация намерений уходит корнями в ранние разговорные диалоговые системы (например, телефонное обслуживание клиентов) и с тех пор получила повсеместное распространение в самых разных областях - от личных помощников и ботов поддержки клиентов до систем медицинских и юридических консультаций.

Ранние методы были основаны на правилах, которые разрабатывались вручную, или на классическом машинном обучении с добавлением дополнительных функций. Однако с развитием области преобладать стали статистические методы, которые основываются на анализе данных. Сначала они использовали традиционные алгоритмы машинного обучения, а затем — методы глубокого обучения. Также мы наблюдаем расширение сферы применения: от простой классификации с закрытым набором параметров, когда каждый запрос должен относиться к одному из известных намерений, до более сложных сценариев. Например, к многоцелевой классификации, обнаружению намерений с открытым доменом или открытым набором параметров (когда запрос не соответствует ни одному из известных намерений), а также к распознаванию намерений с минимальным количеством попыток или вообще без них с помощью мощных генеративных моделей.

1.2 Методы автоматического машинного обучения

Автоматизированное машинное обучение (AutoML) относится к автоматизации полного процесса применения методов машинного обучения для решения реальных задач. Вместо того чтобы вручную выбирать алгоритмы, настраивать гиперпараметры, разрабатывать архитектуры моделей и создавать признаки, система AutoML автоматически принимает эти решения на основе данных. Мотивация для развития AutoML вытекает из бурного роста применения машинного обучения и стремления «демократизировать» машинное обучение – сделать современные техники доступными даже для неспециалистов. Модели машинного обучения зачастую чувствительны к множеству параметров (тип модели, архитектура, настройки гиперпараметров, предварительная обработка признаков и так далее), и нахождение оптимальной конфигурации часто требует кропотливого перебора даже для экспертов. Эта проблема особенно заметна в глубоком обучении, где выбор правильной архитектуры сети и стратегии обучения может определять конечное качество модели. Цель AutoML – автоматизировать принятие этих решений, позволяя пользователю просто предоставить данные, а система подбирает оптимальную модель. Данный обзор литературы предоставляет академический анализ AutoML с основным упором на его применение в обработке естественного языка (NLP), а также включает как фундаментальные работы, так и последние разработки. Мы рассмотрим историческую эволюцию и мотивации AutoML, ключевые технические компоненты, ведущие фреймворки и системы, особенности применения AutoML в задачах NLP (например, классификация текстов, маркировка последовательностей, языковое моделирование), сравнительный анализ производительности и существующие бенчмарки, а также новые тенденции и направления исследований (например, интеграция с фундаментальными моделями, обучение с малым количеством примеров, объяснимость моделей). Обзор ссылается на рецензируемые публикации и академические источники.

1.2.1 H2O

H2O[3] – является платформой машинного обучения с открытым исходным кодом, разработанной для автоматизации процесса контролируе-

мого обучения. Она ориентирована на выполнение таких задач, как бинарная классификация, многоклассовая классификация и регрессия. Основная функция AutoML в H2O заключается в обучении широкого спектра алгоритмов, включая градиентные повышающие машины, случайные леса, глубокие нейронные сети и обобщенные линейные модели. Затем эти модели объединяются в ансамбль для получения наилучших предсказаний. Результатом работы AutoML является таблица лидеров — ранжированный список моделей по показателям производительности, из которого можно выбрать оптимальную модель для развертывания.

Процесс автоматизации в H2O ограничен по времени, что позволяет пользователю задать максимальное время выполнения или количество моделей. Система обучает столько моделей, сколько возможно в рамках установленных ограничений. В отличие от более сложных методов оптимизации гиперпараметров (например, байесовской оптимизации), H2O использует случайный перебор моделей, полагаясь на разнообразие моделей и ансамблевую технику для достижения высокой производительности. Особенностью платформы является автоматическое создание двух сложенных ансамблей: один включает все обученные модели, другой — только лучшие модели каждого семейства алгоритмов. Этот подход позволяет повысить точность предсказаний без ручной настройки.

H2O эффективно обрабатывает большие массивы данных за счет распределения вычислений по нескольким ядрам или узлам кластера. Платформа реализована на языке Java и предоставляет API для Python, R и других языков, что позволяет интегрировать её в различные среды. Результаты работы легко интерпретируемы: на выходе формируется ранжированный список моделей с указанием показателей производительности и времени обучения. Кроме того, встроенные инструменты объяснения моделей позволяют пользователям получать такие пояснения, как важность переменных, графики частичной зависимости и значения SHAP для лучших моделей. Таким образом, H2O обеспечивает возможность построения множества моделей за короткий промежуток времени, что особенно актуально при работе с большими объемами данных.

1.2.2 LightAutoML

LightAutoML[4] (LAMA) — это облегчённый фреймворк AutoML с открытым исходным кодом, предназначенный для моделирования табличных данных. Основное назначение LAMA — автоматическая генерация конвейеров для структурированных данных с акцентом на скорость и эффективность использования вычислительных ресурсов. Он поддерживает задачи бинарной и многоклассовой классификации, а также регрессию. Изначально ориентированный на работу с табличными данными, LightAutoML расширил свои возможности и теперь поддерживает текстовые признаки.

Фреймворк автоматически выполняет предварительную обработку данных, включая очистку и кодирование отсутствующих значений, вывод типов признаков и их отбор в рамках конвейера. Гиперпараметры моделей настраиваются автоматически. LightAutoML предоставляет готовые пресеты конвейеров (например, «TabularAutoML»), которые обеспечивают быстрое развертывание моделей с минимальным вмешательством пользователя. Также доступны гибкие настройки для создания собственных конвейеров с учетом специфики задачи.

Отличительной чертой LightAutoML является параллельное обучение нескольких конвейеров, результаты которых объединяются с использованием ансамблевых методов. Это может быть простое усреднение или более сложное построение, при котором модели одного уровня используют предсказания предыдущего уровня в качестве входных данных. Также важной концепцией является разделение этапов чтения и предварительной обработки: компонент «Reader» проверяет исходный набор данных и определяет необходимые преобразования для различных типов признаков. Это гарантирует надежность и автоматизацию оценки модели.

1.2.3 AutoGluon

AutoGluon – комплексный инструментарий AutoML с открытым исходным кодом, который поддерживает широкий спектр задач машинного обучения, включая прогнозирование табличных данных, компьютерное зрение, обработку естественного языка и прогнозирование временных рядов. Фреймворк предоставляет специализированные API для каждой задачи, например, TabularPredictor, TextPredictor, ImagePredictor и TimeSeriesPredictor, что упрощает использование в различных областях.

AutoGluon поддерживает обучение различных моделей: от древовидных алгоритмов (LightGBM, XGBoost, CatBoost) до нейронных сетей (например, трансформеров для текста и сверточных сетей для изображений), а также простых моделей, таких как k-ближайших соседей и линейные модели. Пользователь может настроить гиперпараметры, выбрать конкретные модели для обучения и использовать предустановленные конфигурации. Таким образом, AutoGluon предоставляет гибкие возможности для настройки обучения с минимальным объемом кода.

1.2.4 FEDOT

FEDOT[5; 6](Flexible Evolutionary Design of Optimal Trees) – фреймворк AutoML с акцентом на оптимизацию конвейеров с помощью эволюционных алгоритмов. Разработанный лабораторией моделирования природных систем Университета ИТМО, он предназначен для автоматизации полного жизненного цикла машинного обучения: от предварительной обработки данных до построения и оптимизации моделей.

Основная идея FEDOT – создание составных конвейеров с помощью генетических алгоритмов. Конвейер представлен в виде направленного ациклического графа (DAG), узлы которого могут быть как преобразованиями данных, так и моделями. Эволюционный оптимизатор, известный как «GOLEM», генерирует начальную популяцию случайных конвейеров и затем улучшает их с помощью мутаций и скрещивания. В результате создаются оптимальные конвейеры, адаптированные к конкретной задаче.

FEDOT поддерживает работу с различными типами данных (табличные, текстовые, графовые) и обеспечивает гибкость настройки. Инструменты анализа позволяют исследовать чувствительность компонентов конвейера, а также оценивать влияние отдельных моделей на итоговую производительность. Фреймворк поддерживает экспорт оптимальных конвейеров в формате JSON и позволяет интеграцию в производственные среды.

1.2.5 Сравнение алгоритмов

Классификация намерений пользователя представляет собой важную задачу в области обработки естественного языка, требующую использования современных методов машинного обучения и автоматизированных инструментов для построения эффективных моделей. Для реализации данной

задачи важно учитывать несколько ключевых критериев: способы обработки текста, поддержка работы с малым набором данных, поддержка выявления намерений вне области определения (Out-Of-Scope, OOS), гибкость настройки параметров, поддержка логирования и возможность использования промптов для энкодеров.

Первым важным критерием является обработка текста, поскольку текстовые данные являются основным источником информации при классификации намерений. Современные модели, такие как трансформеры, демонстрируют высокую точность в задачах NLP благодаря обучению на больших объемах текстов. Поэтому наличие встроенной поддержки текстовых признаков, включая возможность применения эмбеддингов и трансформерных архитектур, является важным аспектом при создании фреймворка.

Вторым значимым критерием является поддержка работы с малым набором данных. В прикладных задачах, связанных с классификацией намерений, часто возникает ситуация, когда количество размеченных данных ограничено. Это особенно актуально при адаптации моделей к новым доменам или редким языковым конструкциям. Поэтому важной характеристикой фреймворка является его способность эффективно работать с малыми наборами данных, например, за счет использования регуляризации или предварительно обученных эмбеддингов.

Не менее важной является поддержка Out-Of-Scope (OOS) – задачи, заключающейся в выявлении запросов пользователя, выходящих за рамки известных классов намерений. Выявление OOS-классов критично для обеспечения надежности и безопасности систем, поскольку позволяет корректно обрабатывать неизвестные или неподдерживаемые запросы. Фреймворки, реализующие данную функциональность, позволяют обучать модели, способные распознавать не только заданные классы, но и детектировать аномальные данные.

Следующим важным аспектом является изменение параметров запуска. В зависимости от задачи, объема данных и доступных вычислительных ресурсов, может потребоваться гибкая настройка процесса обучения. Это особенно актуально при разработке моделей для различных доменов или на основе разнородных данных. Возможность адаптировать парамет-

ры позволяет оптимизировать модель как по точности, так и по времени выполнения.

Поддержка логирования является важным компонентом автоматизации машинного обучения, поскольку позволяет отслеживать процесс обучения, хранить промежуточные результаты и проводить анализ моделей. В контексте классификации намерений важно иметь возможность анализировать ошибки и проверять гипотезы о моделях на каждом этапе обучения. Логирование помогает выявлять причины ухудшения качества моделей и отслеживать процессы настройки гиперпараметров, что критично для обеспечения повторяемости экспериментов и объяснимости конечных результатов.

Последним критерием является поддержка промптов для энкодеров, что особенно важно при использовании моделей на основе трансформеров. В последнее время появляются модели, которые поддерживают промпты в зависимости от задачи, которые улучшают качество ее работы. Например, модель e5[7] использует `query:` и `passage:` для разных частей создания эмбединга для поиска.

1.3 Нейросетевые методы представления текста

1.3.1 BERT

Описание модели и её применения для задачи классификации текста. [2; 8; 9]

1.3.2 Sentence Transformers

Применение и преимущества этой модели для задачи классификации.

1.4 Методы улучшения и адаптации моделей

1.4.1 P-Tuning

Метод для улучшения производительности трансформеров с помощью эффективного использования параметров. [10]

1.4.2 LoRA

Подход к адаптации трансформеров с использованием низкоресурсных параметров. [11]

Таблица 1.1 — Сравнение AutoML фреймворков

Критерий	H2O	LightAutoML	AutoGluon	FEDOT
Способы обработки текста	Нет поддержки из коробки	TF-IDF[1] и эмбединг	Эмбединг	TF-IDF, эмбединг
Поддержка работы с малым набором данных	Не оптимизирован для малых данных	Имеет режимы, позволяющие работать с небольшими наборами данных	Нет поддержки	Может адаптироваться к малым данным
Изменение параметров запуска	Гибкая настройка через API	Настройка через пресеты и конфигурацию, плохо документировано	Можно передавать свой конфиг, плохо документировано	Ограниченная настройка
Поддержка логирования во внешние системы	Логирование результатов через интеграцию с H2O Flow	Нет поддержки	Нет поддержки	Нет поддержки
Поддержка промптов для энкодеров	Нет поддержки	Нет поддержки	Нет поддержки	Нет поддержки
Поддержка OOS (out of scope)	Нет встроенной поддержки	Нет встроенной поддержки	Нет поддержки	Нет поддержки

1.5 Классические методы классификации текста

1.5.1 Logistic Regression

Обзор логистической регрессии в контексте классификации текста.

1.5.2 ML-KNN

Описание метода и его применение в задачах многоклассовой классификации. [12]

1.5.3 DNNC

Описание метода DNNC для классификации, особенности работы и применения. [13]

1.5.4 CatBoost

Преимущества использования CatBoost для задач классификации текста. [14; 15]

1.6 Методы поиска текста

- Faiss[16]
- **Hierarchical Small Navigable Worlds (HSNW)** Применение метода HSNW в контексте классификации текста. [17]

1.7 Используемые метрики

1.7.1 Метрики поиска

- Pk
- NDCG
- MAP
- MRR
- marco

1.7.2 Метрики классификации

- Presicion
- Recall
- Accuracy
- F1
- RocAuc

1.8 Текстовые аугментации

[18]

- Intent-augmentation [19]
- Few-shot detection [20]
- Dspy [21]

2 ПРОЕКТИРОВАНИЕ

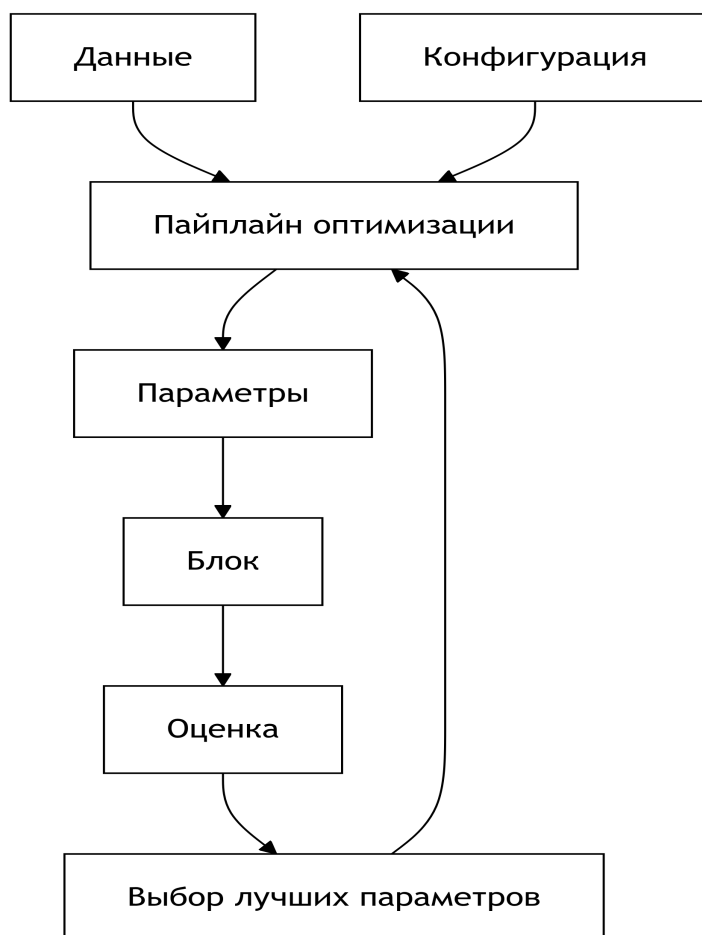


Рисунок 2.1 — Схема фреймворка

Конфигурация



Оптимизатор



Параметры



Блок¹⁶

Заключение

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Jones, Karen Sparck*. A Statistical Interpretation of Term Specificity and Its Application in Retrieval // Journal of Documentation. — 1972. — 1 янв. — Т. 28, № 1. — С. 11—21. — ISSN 0022-0418. — DOI: 10.1108/eb026526. — URL: <https://doi.org/10.1108/eb026526>.
2. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding / J. Devlin [и др.]. — 24.05.2019. — DOI: 10.48550/arXiv.1810.04805. — arXiv: 1810.04805 [cs]. — URL: <http://arxiv.org/abs/1810.04805> (дата обр. 02.08.2023). — Пред. пуб.
3. *LeDell E., Poirier S*. H2O AutoML: Scalable Automatic Machine Learning. — 2020.
4. LightAutoML: AutoML Solution for a Large Financial Services Ecosystem / A. Vakhrushev [и др.]. — 05.04.2022. — DOI: 10.48550/arXiv.2109.01528. — arXiv: 2109.01528 [cs]. — URL: <http://arxiv.org/abs/2109.01528> (дата обр. 04.04.2025). — Пред. пуб.
5. Automated Evolutionary Approach for the Design of Composite Machine Learning Pipelines / N. O. Nikitin [и др.] // Future Generation Computer Systems. — 2022. — Февр. — Т. 127. — С. 109—125. — ISSN 0167739X. — DOI: 10.1016/j.future.2021.08.022. — arXiv: 2106.15397 [cs]. — URL: <http://arxiv.org/abs/2106.15397> (дата обр. 18.05.2025).
6. Multi-Objective Evolutionary Design of Composite Data-Driven Models / I. S. Polonskaia [и др.]. — 17.05.2021. — DOI: 10.48550/arXiv.2103.01301. — arXiv: 2103.01301 [cs]. — URL: <http://arxiv.org/abs/2103.01301> (дата обр. 18.05.2025). — Пред. пуб.
7. Multilingual E5 Text Embeddings: A Technical Report / L. Wang [и др.]. — 08.02.2024. — DOI: 10.48550/arXiv.2402.05672. — arXiv: 2402.05672 [cs]. — URL: <http://arxiv.org/abs/2402.05672> (дата обр. 15.03.2025). — Пред. пуб.

8. *Reimers N., Gurevych I.* Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks. — 27.08.2019. — URL: <http://arxiv.org/abs/1908.10084> (дата обр. 20.12.2022).
9. Attention Is All You Need / A. Vaswani [и др.] // Advances in Neural Information Processing Systems. Т. 30. — 2017. — URL: <https://papers.nips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html> (дата обр. 28.02.2023).
10. Parameter-Efficient Fine-Tuning for Large Models: A Comprehensive Survey / Z. Han [и др.]. — 29.04.2024. — DOI: 10.48550/arXiv.2403.14608. — arXiv: 2403.14608 [cs]. — URL: <http://arxiv.org/abs/2403.14608> (дата обр. 13.05.2024). — Пред. пуб.
11. LoRA: Low-Rank Adaptation of Large Language Models / E. J. Hu [и др.]. — 16.10.2021. — DOI: 10.48550/arXiv.2106.09685. — arXiv: 2106.09685 [cs]. — URL: <http://arxiv.org/abs/2106.09685> (дата обр. 22.11.2023). — Пред. пуб.
12. *Zhang M.-L., Zhou Z.-H.* ML-KNN: A Lazy Learning Approach to Multi-Label Learning // Pattern Recognition. — 2007. — 1 июля. — Т. 40, № 7. — С. 2038—2048. — ISSN 0031-3203. — DOI: 10.1016/j.patcog.2006.12.019. — URL: <https://www.sciencedirect.com/science/article/pii/S0031320307000027> (дата обр. 04.04.2025).
13. Discriminative Nearest Neighbor Few-Shot Intent Detection by Transferring Natural Language Inference / J.-G. Zhang [и др.]. — 25.10.2020. — DOI: 10.48550/arXiv.2010.13009. — arXiv: 2010.13009 [cs]. — URL: <http://arxiv.org/abs/2010.13009> (дата обр. 04.04.2025). — Пред. пуб.
14. *Dorogush A. V., Ershov V., Gulin A.* CatBoost: Gradient Boosting with Categorical Features Support. — 24.10.2018. — URL: <http://arxiv.org/abs/1810.11363> (дата обр. 28.05.2023).
15. CatBoost: Unbiased Boosting with Categorical Features / L. Prokhorenkova [и др.] // Advances in Neural Information Processing Systems. Т. 31 / под ред. S. Bengio [и др.]. — Curran Associates, Inc., 2018. — URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/14491b756b3a51daac41c24863285549-Paper.pdf.

16. The Faiss Library / M. Douze [и др.]. — 11.02.2025. — DOI: 10.48550/arXiv.2401.08281. — arXiv: 2401.08281 [cs]. — URL: <http://arxiv.org/abs/2401.08281> (дата обр. 04.04.2025). — Пред. пуб.
17. *Malkov Y. A., Yashunin D. A.* Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs. — 14.08.2018. — DOI: 10.48550/arXiv.1603.09320. — arXiv: 1603.09320 [cs]. — URL: <http://arxiv.org/abs/1603.09320> (дата обр. 04.04.2025). — Пред. пуб.
18. *Li Z., Larson S., Leach K.* Generating Hard-Negative Out-of-Scope Data with ChatGPT for Intent Classification. — 08.03.2024. — DOI: 10.48550/arXiv.2403.05640. — arXiv: 2403.05640 [cs]. — URL: <http://arxiv.org/abs/2403.05640> (дата обр. 18.05.2025). — Пред. пуб.
19. *Hu R., Khosmood F., Edalat A.* Exploring Description-Augmented Dataless Intent Classification. — 25.07.2024. — DOI: 10.48550/arXiv.2407.17862. — arXiv: 2407.17862 [cs]. — URL: <http://arxiv.org/abs/2407.17862> (дата обр. 04.04.2025). — Пред. пуб.
20. Few-Shot Learning for Multi-label Intent Detection / Y. Hou [и др.] // Proceedings of the AAAI Conference on Artificial Intelligence. — 2021. — 18 мая. — Т. 35, вып. 14, № 14. — С. 13036–13044. — ISSN 2374-3468. — DOI: 10.1609/aaai.v35i14.17541. — URL: <https://ojs.aaai.org/index.php/AAAI/article/view/17541> (дата обр. 04.04.2025).
21. DSPy: Compiling Declarative Language Model Calls into Self-Improving Pipelines / O. Khattab [и др.]. — 05.10.2023. — DOI: 10.48550/arXiv.2310.03714. — arXiv: 2310.03714 [cs]. — URL: <http://arxiv.org/abs/2310.03714> (дата обр. 04.04.2025). — Пред. пуб.