Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»
Физтех-школа Прикладной Математики и Информатики
Научно-образовательный центр когнитивного моделирования

**Направление подготовки / специальность:** 09.04.01 Информатика и вычислительная техника
**Направленность (профиль) подготовки:** Методы и технологии искусственного интеллекта

# ГЕНЕРАЦИЯ СИНТЕТИЧЕСКОГО ДАТАСЕТА ДЛЯ ДЕТЕКЦИИ ИМЕНОВАННЫХ СУЩНОСТЕЙ С ПРИМЕНЕНИЕМ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ

(магистерская диссертация)

**Студент:**
Савкин Максим Константинович

_____
*(подпись студента)*

**Научный руководитель:**
Коновалов Василий Павлович,
канд. техн. наук

_____
*(подпись научного руководителя)*

**Консультант** *(при наличии)*:
_____
_____
*(подпись консультанта)*

Москва 2025

# Содержание

**1 Введение** **4**

**2 Обзорно-постановочный раздел работы** **8**

2.1 Основные понятия и определения . . . . . 8

2.2 Обзор современного состояния проблемы распознавания именованных сущностей . . . . . 10

2.2.1 Подзадача распознавания персональных данных . . . . . 13

2.3 Формальная постановка задачи . . . . . 15

2.3.1 Оценка качества разметки . . . . . 17

**3 Вычислительные эксперименты** **21**

3.1 Датасет для задачи распознавания персональных данных . . . . . 21

3.1.1 Методология генерации . . . . . 23

3.1.2 Анализ данных . . . . . 25

3.1.3 Постановка экспериментов . . . . . 28

3.1.4 Экспериментальные результаты . . . . . 29

3.2 Датасет PoPet для задачи распознавания именованных сущностей . . . 32

3.2.1 Методология генерации . . . . . 33

3.2.2 Анализ данных . . . . . 36

3.2.3 Постановка экспериментов . . . . . 36

3.2.4 Экспериментальные результаты . . . . . 38

**4 Заключение** **39**

**5 Приложения** **47**

5.1 Промпты . . . . . 47

**Аннотация**

Распознавание именованных сущностей (Named Entity Recognition, NER) остаётся актуальной задачей в области обработки естественного языка, особенно в условиях ограниченной доступности качественно размеченных данных. Ручная разметка требует значительных ресурсов и слабо масштабируется, в то время как автоматические методы разметки реальных текстов часто страдают от низкой точности и отсутствия контроля над составом данных.

В данной работе предлагается альтернативный подход, основанный на генерации синтетических наборов данных с использованием больших языковых моделей (LLM). Разработанные методики позволяют задавать структуру текста и точно контролировать распределение и типы внедряемых сущностей. Это обеспечивает высокий уровень гибкости при формировании обучающих выборок под конкретные сценарии.

В рамках исследования были созданы два синтетических корпуса: SPY — для задачи идентификации персональных данных, и PoPet — для решения общей задачи NER в открытом домене. Эксперименты с трансформерными моделями показали, что обучение на синтетических данных обеспечивает сопоставимую или превосходящую эффективность по сравнению с существующими few-shot подходами на основе LLM. Полученные результаты демонстрируют, что синтетические наборы могут служить полноценной заменой вручную размеченным данным в условиях их дефицита.

# 1 Введение

В последние десятилетия наблюдается стремительный рост объёма доступной текстовой информации – новостные публикации, научные статьи, пользовательский контент в социальных сетях и многое другое. Большая часть этих данных представлена в неструктурированном виде, что затрудняет их автоматический анализ. Одной из ключевых задач обработки таких данных является распознавание именованных сущностей (Named Entity Recognition, NER) – детекция и классификация фрагментов текста, содержащих имена людей, организаций, географических объектов и прочее. NER лежит в основе многих прикладных систем: от чат-ботов до систем защиты конфиденциальной информации.

В последние годы наблюдается рост интереса к задаче распознавания персональных данных (Personally Identifiable Information, PII), которая является частным случаем NER. В отличие от стандартных именованных сущностей, задача распознавания PII включает только те сущности, которые могут быть использованы для идентификации конкретного человека, такие как ФИО, контактные данные, номера документов и другие идентификаторы личности. Принципиальное различие между PII и общедоступными именованными сущностями заключается в том, что PII – информация, изначально не предназначенная для публичного доступа, но способная случайно попасть в открытые источники, например, через документы, изначально не предназначенные для публикации. Различие между PII и общедоступными именованными сущностями показано на примерах в таблице 1.

| **Общедоступные данные и Персональная информация** |
|---|
| a) Телефон технической поддержки Apple: 1-800-800-2775.<br>Сатья Наделла является генеральным директором компании Microsoft. |
| b) Лидия Михайловна живет по адресу г. Москва, пер. Металлургов, д. 2/5 стр. 17. |

Таблица 1: Примеры текстов с **a)** общедоступными именованными сущностями; **b)** сущностями содержащими персональные данные. Все примеры персональной информации были сгенерированы с использованием библиотеки Faker [5].

Обнаружение персональных данных имеет важное юридическое и этическое значение. Персональные данные подлежат особой защите в соответствии с законода-

тельством многих стран, и их утечка может повлечь серьёзные последствия как для организаций, так и для физических лиц. Задача автоматического распознавания PII требует специализированных методов и наборов данных, которые учитывают специфику защиты конфиденциальной информации и выходят за рамки классических NER-корпусов.

Формирование обучающих выборок для NER представляет собой серьёзную проблему. Большинство существующих корпусов создаётся путём ручной аннотации — трудоёмкого и дорогостоящего процесса, который плохо масштабируется и требует участия экспертов в целевой области. Создание выборок с разметкой PII является ещё более сложной задачей: из-за юридических ограничений реальные персональные данные практически недоступны.

В связи с трудностями, возникающими при ручной разметке, всё большую популярность приобретает генерация синтетических текстовых данных. Большие языковые модели используются не только для автоматической разметки данных, но и как источники новых обучающих текстов. Синтетические данные находят применение во многих задачах обработки текста, например в анализе тональности, классификации намерений и других задачах, где достаточно сгенерировать текст и сопоставить ему простую метку. Однако в таких задачах, как распознавание именованных сущностей, требуется не просто создать реалистичный текст, но и точно разметить отдельные фрагменты – например указать, какие отрывки текста являются именами, организациями, номерами документов или географическими объектами. Эти задачи предъявляют повышенные требования к точности генерации, внутренней согласованности и формату разметки.

На сегодняшний день можно выделить три основных подхода к созданию размеченных NER-датасетов:

**Ручная разметка датасета** Традиционный способ, предполагающий привлечение аннотаторов с профильной подготовкой. Такой подход обеспечивает высокое качество разметки, но требует значительных временных и финансовых затрат, плохо масштабируется и не подходит для быстрого переноса на новые предметные области.

**Автоматическая разметка** Набор текстов размечается с помощью больших языковых моделей. Этот метод позволяет быстро обрабатывать большие объёмы данных, но имеет ограниченную точность, особенно в специализированных доменах, и может

быть чувствителен к лексическим и стилистическим особенностям корпуса.

**Генерация синтетических текстов** Самый гибкий подход, при котором большие языковые модели используются не для разметки существующих текстов, а для создания новых примеров вместе с уже встроенной аннотацией. Такой метод особенно полезен в следующих случаях: (1) доступ к реальным текстам ограничен (например, из-за приватности); (2) необходимы редкие или специфические типы сущностей; (3) важно обеспечить разнообразие формулировок и сценариев.

Текущие исследования в области генерации синтетических текстов остаются ограниченными. Как правило, они фокусируются на отдельных предметных областях и уделяют недостаточное внимание разнообразию создаваемых текстов. В данной работе рассматриваются эти ограничения и предлагаются способы их преодоления.

В рамках магистерской диссертации были проведены следующие этапы исследования:

1. Проведен обзор существующих подходов и датасетов для задачи распознавания именованных сущностей, включая специализированные корпуса для обнаружения персональных данных. Выявлены их ограничения по разнообразию и масштабируемости.

2. Разработана и реализована многошаговая методология генерации синтетических NER-датасетов, позволяющая управлять распределением и типами сущностей в датасете.

3. Построен синтетический датасет SPY [37] для задачи обнаружения персональных данных, охватывающий несколько тематических доменов и обеспечивающий чёткое разграничение между приватными и общедоступными сущностями при сбалансированном распределении классов.

4. Разработан подход направленный на увеличение лексического и стилистического разнообразия синтетических текстов, на основе которого сгенерирован обобщённый NER-корпус PoPet, включающий более 50 классов сущностей из семи предметных областей.

5. Проведены вычислительные эксперименты по обучению энкодерных моделей на предложенных датасетах. Установлено, что такие модели превосходят суще-

ствующие решения на основе больших языковых моделей и сохраняют устойчивость при тестировании на новых доменах.

# 2 Обзорно-постановочный раздел работы

В этом разделе приводятся основные определения и рассматриваются существующие подходы, связанные с генерацией синтетических данных для задачи NER.

## 2.1 Основные понятия и определения

**Определение 2.1 (Датасет).** *Датасет — структурированный набор текстовых данных, предназначенный для обучения, валидации и тестирования моделей машинного обучения.*

**Определение 2.2 (Токен).** *Токен — текстовая единица, получаемая в результате алгоритма токенизации, которой разбивает исходный текст на фрагменты (слова, отрывки слов, символы). Токены используются для дальнейшего представления текста в векторном виде, который пригоден для распознавания языковой моделью.*

**Определение 2.3 (Словарь).** *Словарь — конечное множество токенов, каждый из которых имеет уникальный идентификатор. Словарь задаёт допустимые единицы текста, с которыми может работать языковая модель.*

**Определение 2.4 (Трансформер).** *Трансформер — архитектура нейронной сети, основанная на механизме внимания, который позволяет моделировать зависимости между элементами последовательности независимо от их расстояния и эффективно параллелизовать вычисления. Классическая трансформерная модель состоит из двух частей: энкодера и декодера. Энкодер преобразует входную последовательность токенов в набор контекстуализированных векторных представлений. Декодер работает в авторегрессионном режиме: на каждом шаге генерации он принимает ранее сгенерированные токены и предсказывает следующий.*

**Определение 2.5 (Большая языковая модель).** *Большая языковая модель (Large Language Model, LLM) — это нейросетевая модель, обученная аппроксимировать вероятностное распределение последовательностей токенов в тексте. Такие модели, как правило, содержат от сотен миллионов до сотен миллиардов параметров и демонстрируют способность решать широкий спектр задач без дополнительного дообучения.*

*На практике под большими языковыми моделями чаще всего понимаются модели основанные на трансформерной декодерной архитектуре, предобученные на больших текстовых корпусах на задаче предсказания следующего токена. В авторегрессионных LLM, таких как GPT-3 [10], предсказание каждого следующего токена $w_t$ осуществляется на основе всех предыдущих токенов:*

$$w_t \sim P_\theta(w_t \mid w_1, \ldots, w_{t-1})$$

*где $\theta$ — параметры модели, оптимизируемые в процессе обучения.*

**Определение 2.6 (Языковое моделирование).** *Языковое моделирование — задача предсказания следующего токена текста на основе предыдущих. Цель — максимизация правдоподобия наблюдаемых последовательностей, корректно предсказывая следующий токен $w_t$ по предыдущим:*

$$w_t \sim P_\theta(w_t \mid w_1, \ldots, w_{t-1}),$$

*где $\theta$ — параметры модели.*

*Такой подход позволяет учитывать статистические закономерности языка и формировать осмысленные продолжения текста.*

**Определение 2.7 (Промпт).** *Промпт — текстовая инструкция или запрос, который подаётся на вход языковой модели для генерации ответа или выполнения задачи.*

**Определение 2.8 (Zero-shot).** *Zero-shot — режим работы модели, при котором промпт не содержит примеров решения задачи.*

**Определение 2.9 (Few-shot).** *Few-shot — режим работы модели, при котором в промпт явно включается небольшое число примеров решения задачи, обычно от одного до нескольких десятков.*

**Определение 2.10 (Синтетический текстовый датасет).** *Синтетический текстовый датасет — это набор текстовых данных, сгенерированный полностью автоматически, без использования текстов написанных людьми.*

**Определение 2.11 (Гиперпараметр).** *Гиперпараметр — это параметр модели или алгоритма обучения, значение которого задаётся до начала обучения и не обновляется в процессе.*

**Определение 2.12 (Краудсорсинг).** *Краудсорсинг — это способ сбора данных, выполнения задач или генерации решений путём привлечения большого числа участников (обычно через онлайн-платформы). Участники выполняют задания добровольно или за вознаграждение. Краудсорсинг широко применяется для аннотирования данных, в том числе в задачах обработки естественного языка, таких как разметка сущностей, классификация текстов и оценка качества генерации.*

**Определение 2.13 (Маркер).** *Маркер — специальный набор символов, который заменяет настоящее значение содержимого в тексте (например,* `username`*).*

## 2.2 Обзор современного состояния проблемы распознавания именованных сущностей

В этой секции описываются наиболее известные датасеты для задачи рапознавания именованных сущностей. Полный список описываемых датасетов с указанием количества типов сущностей и источников разметки представлен в таблице 2.

| Датасет | Источник разметки | Домен | # Типов | Языки | Лицензия |
|---|---|---|---|---|---|
| CoNLL 2003 [1] | 👤 | Newswire | 4 | en, de | CC BY-NC-SA 4.0 |
| OntoNotes 5.0 [4] | 👤 | News, Web, etc | 18 | en, zh, ar | LDC |
| WikiGold [3] | 👤 | Wikipedia | 4 | en | CC-BY 4.0 |
| WNUT17 [6] | 👤 | Social Media | 6 | en | CC-BY 4.0 |
| MasakhaNER 2.0 [12] | 👤 | News | 8 | 20 afr | CC-BY 4.0 |
| MultiNERD [17] | 👤 | Wikipedia | 49 | 8 | CC BY-SA-NC 4.0 |
| HIPE-2022 [16] | 👤 | Historical | 5–16 | fr, de, en | CC BY-SA-NC 4.0 |
| Few-NERD [11] | 👤 | Wikipedia | 66 | en | CC BY 4.0 |
| WikiNEuRal [15] | 👤/🤖 | Wikipedia | 4–5 | 9 | CC BY-SA-NC 4.0 |
| Pile-NER [34] | 🤖 | Pile | 13,020 | en | CC BY 4.0 |
| NuNER [28] | 🤖 | C4 | 200k | en | CC BY 4.0 |
| GLiNER [27] | 🤖 | Pile | 13,000 | en | CC BY 4.0 |
| ProgGen [30] | 🤖 | 🤖 | adapt | en | CC BY 4.0 |
| **Popet(Ours)** | 🤖 | 🤖 | 54 | en | CC BY 4.0 |

Таблица 2: Обзор датасетов для задачи распознавания именованных сущностей. В столбце ***Источник разметки*** условными символами обозначены ручная разметка (👤) и автоматическая генерация (🤖). В датасетах `ProgGen` и `Popet` тексты и разметка были сгенерированы языковыми моделями.

Распознавание именованных сущностей – одна из ключевых задач обработки

естественного языка, применяемая для анализа медицинских и юридических документов, детекции персональных данных, анонимизации датасетов и в других прикладных сценариях. Исторически стандартом для оценки качества NER-систем служит датасет CoNLL-2003 [1], созданный и вручную размеченный ещё до появления трансформерных моделей. Этот датасет содержит короткие тексты из новостного домена, в которых размечены три типа сущностей: PER (люди), LOC (локации) и ORG (организации). Несмотря на свою популярность и историческую значимость, CoNLL-2003 сильно ограничен по ряду причин: мало типов сущностей, поддержка только двух языков и привязка к одному домену.

Большинство существующих NER-датасетов, таких как CoNLL-2003, были созданы для академических задач и содержат узкий набор сущностей, ограниченный стиль и тематический охват. Эти ограничения делают их непригодными для широкого спектра прикладных сценариев, где требуется поддержка специфичных типов сущностей, адаптация к конкретным предметным областям и устойчивость к разнообразию форматов текста. Ручная разметка новых корпусов для каждой задачи требует значительных затрат и плохо масштабируется. В связи с этим встает задача разработки методов, позволяющих формировать обучающие данные быстрее и дешевле.

С развитием больших языковых моделей появилась возможность автоматизировать процесс разметки текстов, что позволило существенно расширить и разнообразить NER-датасеты. Всё больше датасетов начали размечаться автоматически. Как показано в таблице 2 — многие из современных наборов данных в нижней части таблицы были получены именно таким способом. При этом наблюдается устойчивое расширение охвата предметных областей: тексты для этих датасетов заимствуются из крупных многодоменных корпусов. Например, датасет Pile-NER [34] был создан из датасета Pile [14]: сначала было отобрано 50 тысяч неразмеченных отрывков текста, далее при помощи ChatGPT (gpt-3.5-turbo-0301) из них извлекли все сущности и определили их типы, при этом модель не ограничивали фиксированным списком меток. На основе собранного датасета была обучена LLM модель UniversalNER [34] эффективная на множестве существующих NER-датасетов. Дальнейшие исследования улучшили эти результаты, представив более компактные и эффективные модели NuNER [28] и GliNER [27], способные работать с большим количеством типов сущностей и текстовых доменов. Однако датасеты с автоматической разметкой обладают рядом недостатков, к которым относятся низкое качество разметки и нечёткие опре-

деления типов сущностей. Это ограничивает их применение в реальных задачах, требующих четкой формализации классов сущностей и специализации на текстах фиксированного стиля и домена.

Следующей ступенью стало создание полностью синтетических NER-корпусов, в которых генерируется не только аннотация, но и сами тексты. В работе ProgGen [30] была предложена многошаговая методология генерации размеченных текстов для NER-датасета. Сначала LLM анализирует специфику домена и генерирует релевантные атрибуты текста (стиль, длину и другие). Затем генерируется набор именованны сущностей на основе комбинаций этих атрибутов. Далее генерируются текст, в который встраиваются полученные ранее комбинации именованных сущностей. Наконец применяется механизм самокоррекции для проверки и исправления аннотаций. Данная методология позволяет контроллировать формат и содержанием данных. Тем не менее, в полностью синтетических датасетах часто присутствует шаблонность и недостаток текстового разнообразия [26]. Для приложений NLP важно обеспечить максимальную вариативность, ограниченную рамками задачи — например, собрать корпус из разнообразных постов на тему здравоохранения, написанных в разных стилях и форматах, чтобы модель эффективно обрабатывала всё многообразие возможный вариаций входных текстов.

Разнообразие текстовых данных определяется множеством различных факторов начиная от лексики и семантики текста и заканчивая охватываемыми предметными областями. В частности под семантическим разнообразим текстов подразумевается мера, отражающая, насколько сильно различаются смыслы (темы, идеи, значения) между текстами внутри набора. Чем больше семантическое разнообразие, тем шире спектр представленных понятий и тем меньше тематическое или лексическое перекрытие между текстами.

Для усиления лексической и семантической вариативности синтетических текстов применяются методы изменяющие распределение вероятностей генерируемых токенов. К таким подходам относятся методы снижающие вероятность генерации частых токенов и отдающие больший приоритет менее распространённым, при этом сохраняя логику и структуру текста. Самый распространённый способ — изменение температуры генерации. Температура — это скалярный гиперпараметр, применяемый к логитам распределения вероятностей перед выбором следующего токена по следующей формуле:

$$p_i = \text{softmax}\left(\frac{l_i}{T}\right)$$

где:

- $l_i$ — логит для токена $i$,
- $T$ — гиперпараметр температуры,
- $p_i^{(T)}$ — итоговая вероятность токена $i$ после применения температуры.

**Замечание 2.1.** Температура регулирует энтропию выходного распределения. Ее можно интерпретировать следующим образом:

- $T \to 0$ - распределение становится острым и близким к детерминированному.
- $T = 1$ - распределение остаётся неизменным.
- $T \to \infty$ - распределение стремится к равномерному.

Однако при высокой температуре генерация может терять связность, поэтому на практике вводят более продвинутые методы, такие как diverse beam search: модифицированная версия beam search, при которой лучи (beams) делятся на группы, и вводится штраф за слишком сильное сходство между группами, что способствует генерации разнотипных вариантов последовательностей токенов. Другой подход - logit suppression: уменьшает логиты высокочастотных токенов, чтобы модель предпочитала менее вероятные варианты, без радикального искажения смысла.

Кроме техник влияющих на распределение вероятностей токенов, повышение разнообразия может достигаться сменой языковой модели. Модели могут существенно отличаться словарем, стилистикой текстов, охватываем доменам, языкам и синтаксическим структурам. Поэтому даже если используются одинаковые техники генерации, смена модели сама по себе вносит дополнительный вклад в разнообразие текстов.

### 2.2.1 Подзадача распознавания персональных данных

Распознавание персональных данных представляет собой частный и очень важный случай задачи извлечения именованных сущностей. На ранних этапах её решения преобладали подходы, основанные на использовании регулярных выражений и

разработанных вручную шаблонов. Эти методы демонстрировали удовлетворительную точность при выявлении строго формализованных типов PII, таких как адреса электронной почты, номера телефонов и IP-адреса. Однако при анализе неструктурированных текстов, содержащих более контекстно-зависимые сущности (например, имена или физические адреса), подобные эвристические подходы оказываются менее эффективными.

С развитием методов машинного обучения внимание исследователей сместилось в сторону использования обучаемых моделей, в частности систем, основанных на задаче NER. Такие модели обладают способностью учитывать контекст и демонстрируют высокую адаптивность при обнаружении разнообразных типов персональных данных в свободном тексте [9, 18, 22]. Однако ключевым ограничением при разработке таких систем остаётся недоступность датасетов с высоким качеством разметки.

Многие существующие датасеты, используемые для обучения и оценки NER-моделей, изначально создавались не для задач приватности и часто не отражают специфику персональной информации. Не все именованные сущности являются персональными данными, и попытки применить стандартные NER-корпусы к задаче обнаружения PII могут приводить к значительному числу ложноположительных результатов. Например, система Presidio от Microsoft [13], представляющая собой промышленный инструмент для детекции PII, сочетает в себе регулярные выражения, шаблоны и предобученные NER-модели. Несмотря на гибкость такого гибридного подхода, Presidio не учитывает контекст появления сущности: любое имя собственное автоматически классифицируется как PII, независимо от его характера.

Одной из ключевых проблем в данной области остаётся ограниченный доступ к открытым текстовым корпусам, содержащим реальные примеры персональных данных. Этические и юридические ограничения делают невозможным широкое распространение таких текстов. В связи с этим были предложены два подхода к формированию обучающих данных: (1) замена реальных персональных данных в исходных текстах синтетическими аналогами (анонимизация с сохранением структуры), и (2) генерация полностью синтетических корпусов.

Ниже представлены наиболее значимые датасеты, используемые в задачах детекции PII:

**BigCode PII** [21] — датасет, сформированный на основе корпуса The Stack [23], включающего исходный код из открытых репозиториев GitHub. Основное внимание

уделяется обнаружению PII в программном коде (например, ключей API, IP-адресов, email-адресов). Для предварительной фильтрации проводилась разметка с привлечением 1399 краудсорсеров через платформу Toloka. Хотя датасет полезен для анализа кода, он неприменим к обычным текстам, а ручная разметка ограничивает его масштабируемость.

**AI4Privacy** [35] представляет собой датасет, созданный с использованием проприетарных алгоритмов разметки. Корпус охватывает шесть языков, включает 63 класса PII, что делает его одним из наиболее универсальных. Однако закрытый программный код и отсутствие деталей генерации снижает прозрачность подхода, затрудняя оценку качества данных и их адаптацию под конкретные задачи.

**Kaggle PII Detection** [33] — датасет, подготовленный в рамках соревнования на платформе Kaggle, включает около 22 тысяч эссе, написанных участниками онлайн-курса. В отличие от других рассмотренных корпусов, он содержит явное разделение между PII и общедоступными сущностями одного типа. Однако и у этого датасета имеются существенные ограничения: все тексты написаны в ответ на одно и то же задание, что ограничивает разнообразие лингвистических конструкций. Кроме того, в открытом доступе есть только 30% данных, среди которых большинство текстов не содержат персональных данных.

## 2.3 Формальная постановка задачи

Параграф содержит формальные определения модели, обучающей процедуры и синтетического датасета для задачи NER, а также постановку основной цели работы — оптимизацию генерации синтетических данных для достижения высокого качества разметки реальных тестов.

**Распознавание именованных сущностей**

Распознавание именованных сущностей — это задача, в которой по входной последовательности токенов $x = (x_1, x_2, \ldots, x_n)$, где $x_i \in \mathcal{V}$, требуется построить последовательность меток $y = (y_1, y_2, \ldots, y_n)$, таких что $y_i \in C = \{c_1, c_2, \ldots, c_K\}$, где $C$ — множество классов именованных сущностей (например, PERSON, LOCATION, ORGANIZATION и другие).

**Модель потокенной классификации** $M_W$ — трансформерная модель с параметрами $W$, отображающая последовательность токенов $x = (x_1, \ldots, x_n)$ в последовательность скрытых представлений:

$$M_W(x) = (l_1, l_2, \ldots, l_n) \in \mathbb{R}^{n \times d}, \tag{2.1}$$

где $l_i \in \mathbb{R}^d$ — скрытое представление токена $x_i$.

Для каждого $l_i$ применяется классификатор на основе softmax, который предсказывает распределение вероятностей над метками:

$$P(y_i \mid x) = \text{softmax}(W_{\text{clf}} l_i + b), \quad W_{\text{clf}} \in \mathbb{R}^{K \times d},\ b \in \mathbb{R}^K. \tag{2.2}$$

Предсказанная последовательность меток определяется как:

$$\widetilde{Y} = (\widetilde{y}_1, \widetilde{y}_2, \ldots, \widetilde{y}_n), \quad \widetilde{y}_i = \arg\max P(y_i \mid x). \tag{2.3}$$

Модель обучается на датасете $\mathcal{D}_{\text{train}} = \{(x^{(t)}, y^{(t)})\}_{t=1}^N$ с использованием кросс-энтропийной функции потерь:

$$\hat{W} = \arg\min_W \sum_{t=1}^{N} \sum_{i=1}^{n_t} \text{CE}(y_i^{(t)}, P(y_i^{(t)} \mid x^{(t)})). \tag{2.4}$$

**Синтетический датасет для задачи NER** $\mathcal{D}_{\text{syn}}$ определяется следующим образом:

Пусть

- $\mathcal{T} = \{x = [x_1, x_2, \ldots, x_n]$, где $x_j \in \mathcal{V} -$ словарь$\}$ — множество всевозможных входных текстов для заданного словаря.
- $p_{\text{real}}(D)$ — неизвестное распределение над последовательностями токенов $x = (x_1, x_2, \ldots, x_n)$, где $x_i \in \mathcal{V}$, распределение текстов в заданном домене $D$.
- $\mathcal{L}_\Theta(x, \alpha)$ - большая языковая модель с параметрами $\Theta$, гиперпараметрами генерации $\alpha$, аппроксимирующая реальное распределение текстов на заданном домене $D$: $\mathcal{L}_\Theta(x, \alpha) \approx p_{\text{real}}(x|D)$
- $\Phi(\mathbf{p}, x)$ - детерминированная функция извлекающая разметку из сгенерированного текста $x$ согласно заданному промту $\mathbf{p}$.

Тогда синтетический текстовый датасет $\mathcal{D}_{\text{syn}}$ для задачи классификации на уровне токенов определяется следующим образом:

$$\mathcal{D}_{\text{syn}}(\mathbf{p}, \Phi, \theta, \alpha) = \left\{(x_i, y_i) : \; x_i \sim \mathcal{L}_\Theta(\mathbf{p} \oplus x_{[0,i-1]}, \alpha); \; y = \Phi(\mathbf{p}, x)\right\}_{i=1}^{M} \tag{2.5}$$

**Цель работы** - построение такого набора синтетических текстовых данных $\mathcal{D}_{\text{syn}}$, что энкодерная модель $M_W$ обученная на нем показывает высокое значение метрики F1-score при переносе на бенчмарки с человеческой разметкой $\mathcal{D}_{\text{test}} = \{(X_{\text{test}}, Y_{\text{test}})\}$.

Оптимизационная задача формулируется следующим образом:

$$\gamma^* = \arg\max_{\gamma} \; F_1\left(M_{\hat{W}(\gamma)}(X_{\text{test}}), Y_{\text{test}}\right), \gamma = (\mathbf{p}, \Phi, \theta, \alpha) \tag{2.6}$$

Где $\hat{W}(\gamma)$ — параметры энкодерной модели $M_{\hat{W}}$, полученные при обучении на синтетическом датасете $\mathcal{D}_{\text{syn}}(\mathbf{p}, \Phi, \theta, \alpha)$.

**Замечание 2.2.** Данная постановка задачи поощряет:

- **Лингвистический реализм** текстов $x \sim p_{real}(D)$.
- **Соответствие аннотаций** $y$, принятым в корпусах $\mathcal{D}_{\text{test}}$ с ручной разметкой.

### 2.3.1 Оценка качества разметки

В большинстве наборов данных для задач извлечения именованных сущностей используется схема разметки BIO (*Begin, Inside, Outside*), которая позволяет однозначно определить границы и типы сущностей в тексте на уровне токенов. В этой схеме каждому токену присваивается одна из следующих меток:

- **B-<entity_type>** — токен является **первым** токеном некоторой сущности заданного типа (например, `B-PER` для начала имени человека).
- **I-<entity_type>** — токен является **внутренним** токеном сущности того же типа (например, `I-PER`).
- **O** — токен **не относится** ни к одной именованной сущности.

Такая разметка позволяет моделям однозначно определять, где начинаются и заканчиваются сущности, а также различать соседние сущности одного типа. Например, последовательность токенов [‘‘John’’, ‘‘Smith’’, ‘‘met’’, ‘‘Anna’’, ‘‘Brown’’] может быть размечена как [‘‘B-PER’’, ‘‘I-PER’’, ‘‘O’’, ‘‘B-PER’’, ‘‘I-PER’’], что указывает на наличие двух отдельных сущностей типа PER. BIO-разметка широко используется в популярных датасетах для NER, таких как CoNLL-2003 [2], OntoNotes [4] и других.

Существуют и другие схемы, такие как например BIOES, которая обозначает сущности состоящие из одного токена как **S-<entity_type>**), но BIO остаётся наиболее распространённой благодаря своей универсальности и поддержке большинством инструментов и библиотек. Поэтому во всех экспериментах, описанных в данной работе, используется именно BIO-схема разметки.

**Метрики качества разметки** Для оценки качества разметки текстов содержащих именованные сущности использовались метрики точности (precision), полноты (recall) и F1-меры (F1-score), которые являются стандартными для задач классификации токенов [2]. Далее представлены формулы для расчета указанных метрик:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.7}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.8}$$

$$\text{F1-score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.9}$$

Где $TP$, $FP$, $FN$ — соответственно число истинно положительных, ложно положительных и ложно отрицательных предсказаний.

**Span-level метрики** При подсчёт метрик на уровне сущностей (span-level, entity-level), величины TP, FP и FN подсчитываются на уровне полных сущностей (отрезков текста с соответствующим классом). Формально:

**Определение 2.14.** *Обозначим предсказанный набор сущностей как* $\hat{S} = \{(\hat{s}_i, \hat{e}_i, \hat{c}_i)\}_{i=1}^{n}$, *где* $\hat{s}_i$, $\hat{e}_i$ *— начальный и конечный индексы сущности, а* $\hat{c}_i$ *— её класс. Истинная разметка —* $S = \{(s_j, e_j, c_j)\}_{j=1}^{m}$.

*Тогда величины TP, FP и FN для span-level метрик определяются следующим образом:*

- $TP = |\hat{S} \cap S|$ — *количество сущностей, которые в точности совпадают по начальной и конечной позиции, а также по метке класса.*
- $FP = |\hat{S} \setminus S|$ — *количество предсказанных сущностей, не имеющие точного совпадения в оригинальной разметке.*
- $FN = |S \setminus \hat{S}|$ — *количество сущностей в оригинальной разметке, не предсказанных моделью.*

**Token-level метрики** В потокенной (token-level) оценке TP, FP, FN подсчитываются по отдельным токенам, а не по целым сущностям. Для каждого токена сравниваются предсказанная и истинная метки.

**Определение 2.15.** *Рассмотрим токенизированную последовательность длины $T$. Обозначим предсказанные метки для каждого токена как $\hat{y} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_T)$, а истинные метки — как $y = (y_1, y_2, \ldots, y_T)$, где $\hat{y}_t, y_t \in \mathcal{C}$ — метки из набора допустимых меток $\mathcal{C}$. Тогда величины TP, FP и FN для token-level метрик определяются следующим образом:*

- $TP = |\{t \in [1, T] \mid \hat{y}_t = y_t \wedge y_t \neq \texttt{O}\}|$ — *количество токенов, для которых предсказанная метка совпадает с истинной и не является пустой меткой* `O`*;*
- $FP = |\{t \in [1, T] \mid \hat{y}_t \neq y_t \wedge \hat{y}_t \neq \texttt{O}\}|$ — *количество токенов, ошибочно размеченных как сущности;*
- $FN = |\{t \in [1, T] \mid \hat{y}_t \neq y_t \wedge y_t \neq \texttt{O}\}|$ — *количество токенов, которые должны были быть размечены как сущности, но не были.*

Потокенные метрики менее строгие, поскольку частичное совпадение (например, один токен из трёх угадан) увеличивает значение TP, в отличие от span-level, где требуется полное совпадение сущности.

**Выводы** В задаче извлечения именованных сущностей возможна оценка качества модели как на уровне токенов (token-level), так и на уровне сущностей (span-level). Однако token-level метрики имеют ряд ограничений: частичное совпадение сущности может необоснованно увеличивать значение $TP$, тем самым искажая общую картину качества модели. Такие метрики могут переоценивать модель, особенно в случае длинных сущностей. Span-level метрики более строго оценивают качество извлечения сущностей, требуя точного совпадения как границ, так и типа сущности. Это соответствует прикладным требованиям к системам NER, где частично извлечённые сущности зачастую не имеют практической ценности (например, при сборе персональных данных или юридически значимой информации). Чтобы корректно отразить способность модели извлекать целостные сущности, в работе используются именно span-level метрики. Расчет метрик осуществляется с использованием библиотеки Seqeval [7].

# 3 Вычислительные эксперименты

## 3.1 Датасет для задачи распознавания персональных данных

Применение больших языковых моделей для аннотирования текстов оказалось эффективным при работе с корпусами, содержащими большое количество именованных сущностей [34, 29, 27]. Однако при недостатке данных этот подход работает значительно хуже. Даже при использовании качественной ручной разметки, в условиях дефицита данных итоговый датасет может оказаться ограниченным по количеству и балансу присутствующих в нем именованных сущностей. Проблема особенно остро проявляется в задачах, связанных с детекцией сущностей содержащих персональные данные: такие сущности крайне редко встречаются в открытых датасетах, поскольку обычно предварительно удаляются их авторами. Кроме того, аннотации в таких текстах часто оказываются сильно несбалансированными. Например, в датасете из соревнования Kaggle PII Detection [33] количество примеров по классам варьируется от всего лишь 2 до 1200 (см. рис. 1), что приводит к выраженному дисбалансу.

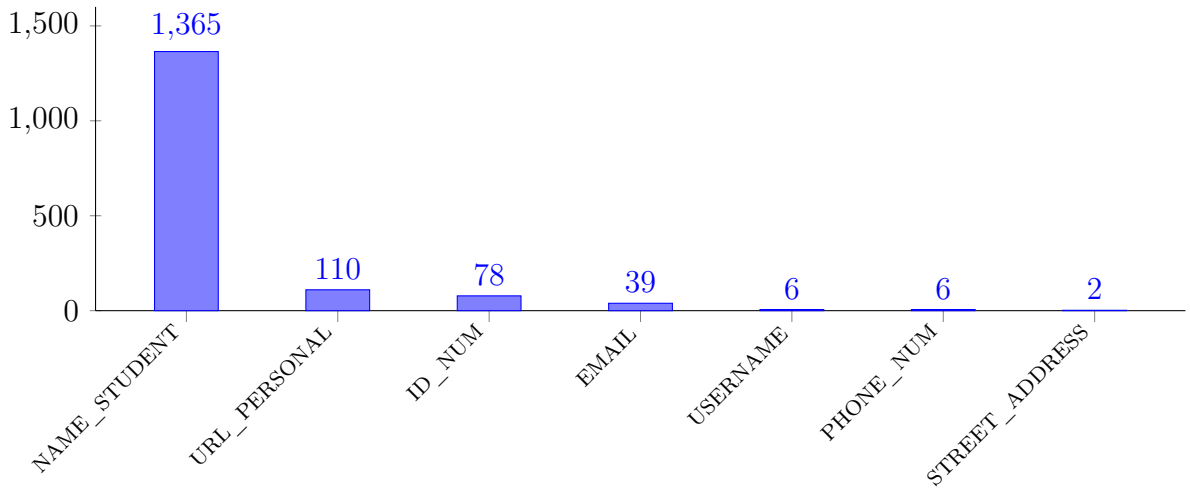


Рис. 1: Распределение сущностей каждого типа в датасете Kaggle PII [33].

В данной работе предлагается методология, позволяющая устранить указанные ограничения за счёт генерации синтетических текстов с контролируемым включением сущностей. Для генерации были выбраны два текстовых домена:

- Юридический — неформальные юридические вопросы, аналогичные вопросам на форуме rLegalAdvice[1].

[1] `https://www.reddit.com/r/legaladvice/`

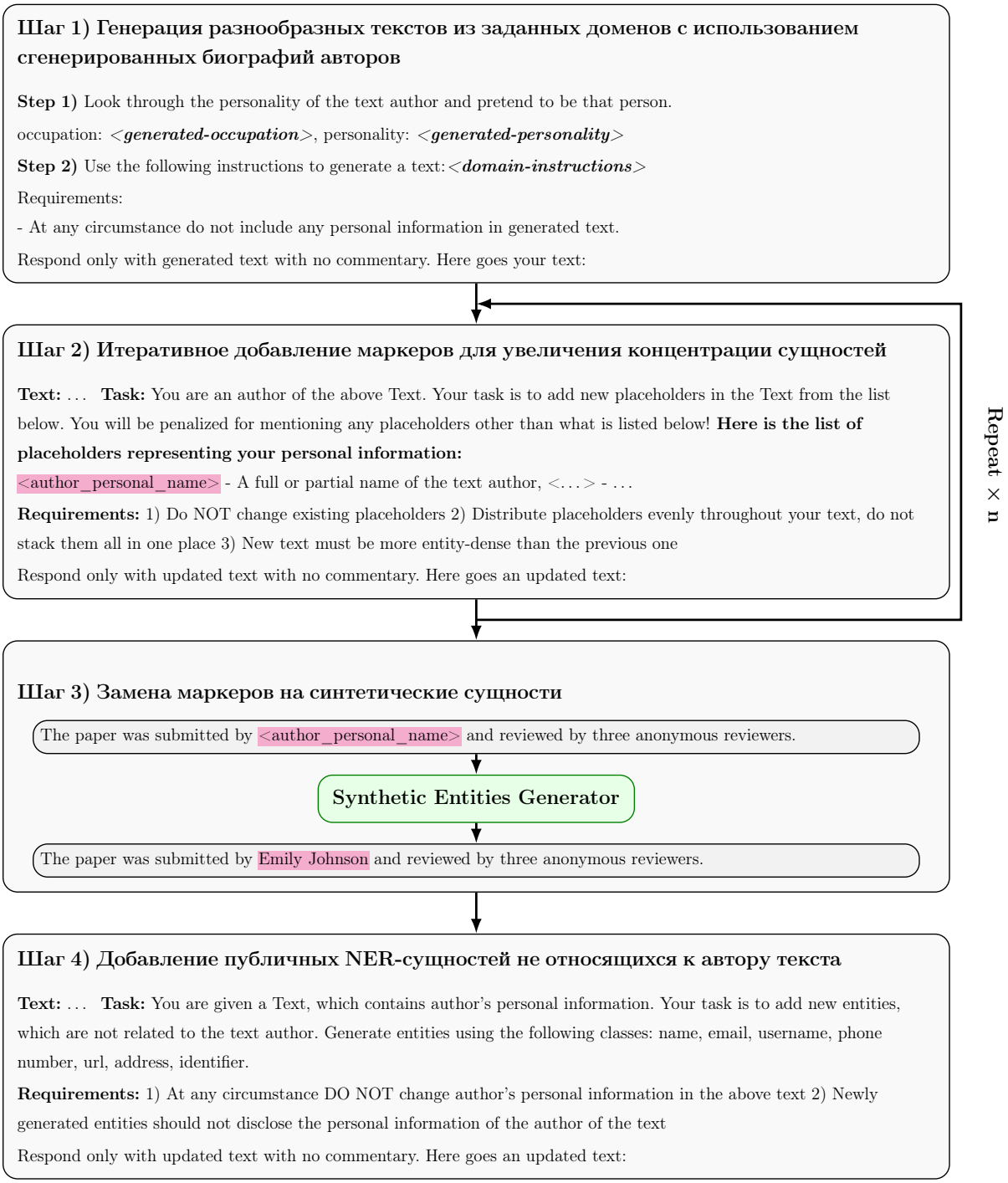


Рис. 2: Методология генерации датасета SPY. В каждом блоке представлены использовавшиеся промпты. Красным цветом выделены персональные данные автора текста.

- Медицинский — формы, заполняемые пациентами для онлайн-консультации.

Для аннотации были выбраны следующие классы PII: `name`, `email`, `phone number`, `personal url`, `personal identifier`, `username`, `personal address`.

### 3.1.1 Методология генерации

Одной из ключевых проблем при разработке методологии генерации текстов с персональными данными является отсутствие чёткой границы между PII-сущностями и общедоступной информацией. В ряде случаев определить, является ли конкретная сущность персональной, возможно лишь путём дополнительного поиска в интернете, что приводит к низкому качеству автоматизированной аннотации. Однако для задач, связанных с защитой конфиденциальной информации, принципиально важно различать эти категории. Чтобы сформировать однозначное и воспроизводимое разделение между персональными и неперсональными сущностями, было введено ограничение: персональной информацией считаются только сущности содержащие данные автора текста, в то время как все прочие сущности считаются общедоступными и не размечаются. Такое допущение позволяет добиться формализации границы между PII и общей NER-разметкой, но сужает спектр возможных случаев появления персональных данных.

Методология для датасета SPY (synthetic PII dection dataset) была разработана с учетом следующих критериев: (1) добавление особенностей домена и естественная интеграция PII-сущностей, (2) включение как персональных, так и общедоступных сущностей из заранее определенных категорий и (3) сохранение четкого разделения между персональными и общедоступными данными. Для достижения этих целей была реализована многошаговая методология проиллюстрированная на рисунке 2.

**Шаг 1. Генерация текстов в заданном домене** Первый шаг — генерация разнообразных текстов в заданном домене. Для генерации использовалась модель Llama-3-70B [25] с промптом представленном на рисунке 10. На момент проведения основных экспериментов Llama-3-70B была одной из лучших инструктивных моделей с 70 миллиардами параметров. Она показала хорошие результаты при выполнении всех необходимых манипуляций с данными. Тестирование проприетарных моделей не проводилось из-за бюджетных ограничений.

Одним из серьезных ограничений полностью сгенерированных текстовых данных является их семантическое однообразие. Для повышения разнообразия генерируемых текстов в промт была добавлена информацию о профессии и личности автора, что расширило спектр тематик текстов в каждом из доменов. Информация об авторе текста была сгенерирована с помощью промпта, представленного на ри-

сунке 9.

**Шаг 2. Итеративное добавление маркеров** Изначально предполагалось, что в тексты полученные на первом шаге будут сразу добавляться конкретные PII-сущности вместо маркеров, однако из-за частых искажений сущностей при генерации текста пришлось отказаться от этого варианта. Вместо этого на втором шаге осуществляется интеграция маркеров вида $<author_personal_name>$, обозначающих PII-сущности, что позволяет минимизировать нежелательные перефразирования.

Еще одна проблема, возникшая в ходе разработки пайплайна - одновременное добавление нескольких маркеров. Модель не справлялась с тем чтобы добавить сразу все маркеры указанные в промпте. Для решения этой проблемы использовался итеративный подход, в котором модель постепенно обновляла предыдущую версию текста, добавляя новые сущности, как показано на рисунке 12.

Также проводились эксперименты с техникой Chain-of-Density [19], которая позволяет выполнять несколько обновлений текста в одном запросе. Ограничивающим фактором стала большая длина сгенерированных текстов, которые не помещались в контекстное окно модели при множественном переписывании.

**Шаг 3. Замена маркеров на синтетические сущности** Третий этап — замена маркеров на соответствующие им синтетические сущности с помощью библиотеки `Faker` [5]. Faker — открытый инструмент для создания реалистичных примеров персональных данных. Он поддерживает выбор локализации и позволяет генерировать широкий спектр сущностей, включая имена, адреса, email, номера телефонов и другие типы данных.

С помощью Faker были сгенерированы разнообразные синтетические PII-сущности, которые были подставлены вместо маркеров. После замены был получен набор текстов с персональной информацией привязанной исключительно к автору текста.

**Шаг 4. Добавление общедоступных именованных сущностей** На последнем этапе были добавлены публичные именованные сущности, промпт представлен на рисунке 11.

### 3.1.2 Анализ данных

Согласно предложенной методологии было сгенерировано 8688 текстов в юридическом и медицинском доменах, содержащие 58794 примеров сущностей.

Гибкая структура генерации синтетических текстов имеет несколько ключевых преимуществ, позволяющих не только контролировать состав и плотность персональной информации, но и устранять типичные ограничения классических подходов к созданию NER-корпусов.

**Равномерное распределение PII-сущностей** Одним из критических требований к датасетам, содержащим персональные данные, является равномерное распределение сущностей по всему тексту. Это позволяет моделям обучаться в условиях, приближенных к реальным, когда персональные данные могут встречаться в любой части текста.

На рисунке 3 представлена тепловая карта распределения PII-сущностей по четырём условным зонам текста: (0–25%), (25–50%), (50–75%) и (75–100%). По вертикальной оси указаны типы сущностей, а по горизонтальной — сегменты текста. Цвет ячеек и числовые значения внутри показывают количество примеров соответствующего типа сущности в данном участке текста. Из графика видно, что:

- `Name` чаще всего упоминается в самом начале — что соответствует частому использованию имени в приветствиях или вводных фразах.
- `Phone num` и `id_num` чаще встречаются ближе к концу текста, когда люди оставляют свои контакты для дальнейшей связи.
- Представленный метод генерации не привязывает PII-сущности к жёстко фиксированным позициям, что критично для тренировки универсальных моделей.

**Баланс классов сущностей** Для построения эффективных систем извлечения сущностей важно не только равномерное распределение сущностей, но и сбалансированное количество примеров каждого типа.

Таблица 3 демонстрирует общее количество вхождений каждого типа PII-сущностей в двух доменах: `legal question` и `medical consultations`. Видно, что

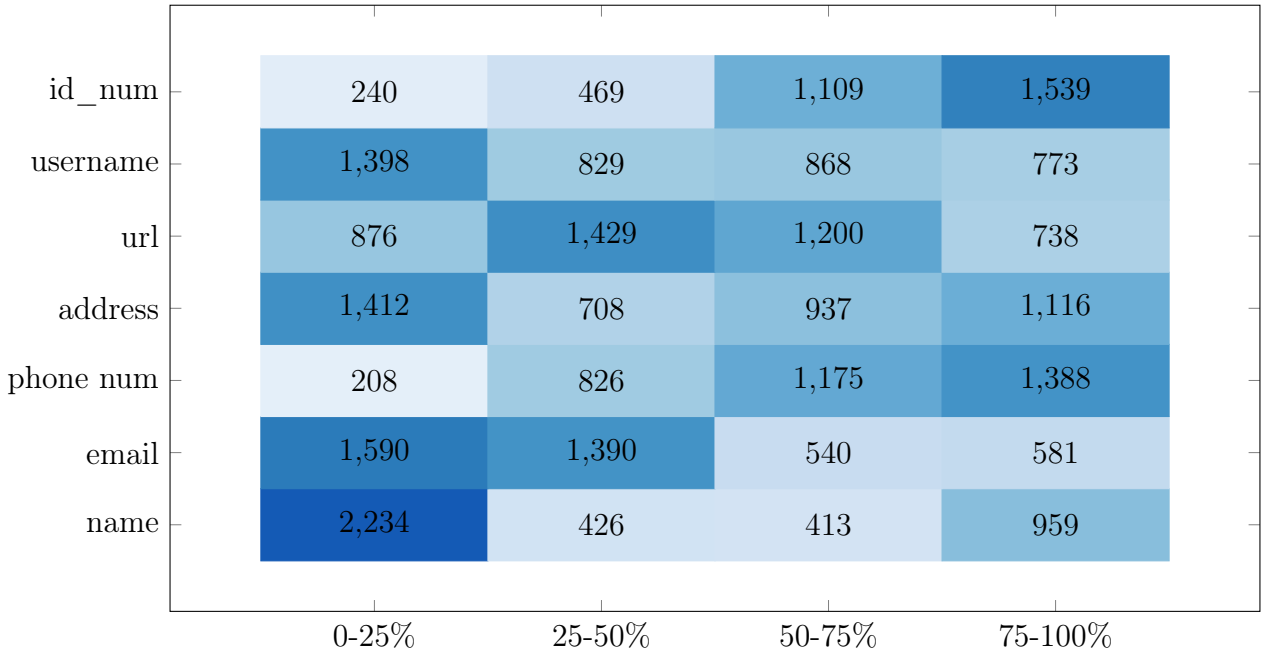


Рис. 3: Распределение позиций различных типов PII-сущностей в текстов юридического домена датасета SPY. По оси $x$ представлены четыре интервала с позициями токенов в тексте (от 0% до 100% длины текста). По оси $y$ отображены категории сущностей. Цвет и числовые значения в ячейках матрицы отражают количество встречаемых сущностей данного типа в соответствующем интервале позиции.

все семь типов представлены в объёме от $3,500$ до $5,100$ экземпляров. Это обеспечивает хороший баланс классов, минимизирует смещение модели к отдельным категориям и позволяет использовать традиционные метрики (precision, recall, F1) для корректной оценки качества модели.

| **Тип сущности** | **Домен** | |
|---|---|---|
| | Legal questions | Medical Consultations |
| url | 4,243 | 4,322 |
| email | 4,101 | 4,493 |
| username | 3,868 | 4,273 |
| address | 4,173 | 5,122 |
| name | 4,032 | 4,707 |
| phone number | 3,597 | 4,222 |
| id_num | 3,357 | 4,284 |

Таблица 3: Распределение количества PII-сущностей в датасете SPY.

| Entity | Legal questions | | | Medical Consultations | | |
|---|---|---|---|---|---|---|
| | PII update | | Final | PII update | | Final |
| | #1 | #2 | | #1 | #2 | |
| Name | 0.6 | 1.1 (+0.5) | 0.9 (+0.3) | 0.7 | 1.1 (+0.4) | 1.0 (+0.3) |
| Email | 1.0 | 1.2 (+0.1) | 0.9 (-0.2) | 1.0 | 1.1 (+0.1) | 0.9 (-0.1) |
| Username | 0.9 | 1.1 (+0.2) | 1.3 (+0.4) | 0.8 | 1.2 (+0.4) | 1.3 (+0.5) |
| Phone | 0.9 | 1.1 (+0.2) | 0.8 (-0.1) | 0.9 | 1.1 (+0.2) | 0.9 (+0.0) |
| URL | 1.1 | 1.3 (+0.3) | 0.9 (-0.2) | 1.0 | 1.3 (+0.3) | 0.9 (-0.2) |
| Address | 0.7 | 1.2 (+0.5) | 0.9 (+0.2) | 0.7 | 1.3 (+0.6) | 1.1 (+0.3) |
| ID | 0.4 | 1.0 (+0.6) | 0.7 (+0.3) | 0.5 | 1.1 (+0.5) | 0.9 (+0.4) |
| Average | 0.8 | 1.1 (+0.4) | 0.9 (+0.1) | 0.8 | 1.2 (+0.4) | 1.0 (+0.2) |

Таблица 4: Среднее количество PII-сущностей в тексте, сгенерированном согласно методологии SPY на рисунке 2. Обозначение *PII update #k* соответствует количеству маркеров после $k$ итераций шага два; *Final* — соответствует среднему количество PII-сущностей в итоговых текстах.

**Контроль плотности персональных данных** Ещё одно важное преимущество предлагаемой методологии — возможность контролировать среднее количество PII-сущностей в тексте. Этот контроль реализуется через итеративный механизм добавления маркеров: после первой итерации (PII update #1) в текст добавляется начальный набор маркеров, затем во второй итерации (PII update #2) производится их расширение, а на финальном этапе (Final) добавляются неперсональные данные.

В Таблице 4, показано среднее количество маркеров и PII-сущностей в сгенерированном датасете на разных шагах генерации. Наблюдается стабильное увеличение среднего количества сущностей заданного типа во время итеративного добавления маркеров. В текущей версии датасета было решено остановиться на двух обновлениях, чтобы текст не был перегружен чрезмерным количеством персональной информации.

**Контроль общедоступных сущностей** Кроме персональных данных, важным элементом методологии является возможность избирательно добавлять общедоступные сущности, такие как организации, локации, известные личности и другие. В корпусах, основанных на реальных текстах и автоматической разметке, невозможно точно проконтролировать, какие сущности будут включены в выборку без под-

ключение дополнительных инструментов определения того является ли информация общедоступной. В нашей методологии этого ограничения нет, за счет того что вся персональная информация относится к автору текста.

Генерация общедоступных сущностей может приводить к "вытеснению"ранее сгенерированных PII-маркеров, особенно если модель пытается оптимизировать связность или реализм текста. Это видно в колонке *Final* таблицы 4, где после генерации общедоступных NER-сущностей количество некоторых PII-типов снижается по сравнению с предыдущими итерациями.

### 3.1.3 Постановка экспериментов

В большинстве существующих датасетов для задачи обнаружения PII, аннотация производится по типам сущностей (например, имена, адреса, номера телефонов и др.) без учёта принадлежности этих данных конкретному человеку или их статуса как общедоступной информации (например, публичные имена организаций, географические объекты). Вследствие этого обучение моделей на таких датасетах сводится к идентификации всех сущностей заданных категорий без разделения на персональные и общедоступные. Поскольку в рамках настоящей работы требуется такое разделение, все подобные датасеты не будут использованы в дальнейших экспериментах.

В качестве базовых решений задачи распознавания PII были выбраны следующие подходы:

- **Presidio** [13] — инструмент от Microsoft, использующий комбинацию методов NER, регулярных выражений и дополнительную логику на основе правил.
- **LLaMA-3-70B** [25] — большая языковая модель работающая в zero-shot режиме. Эта модель обрабатывает и идентифицирует широкий спектр персональной информации напрямую из текста, демонстрируя высокую адаптивность и обобщаемость на различные типы PII-сущностей. Промпт для извлечения персональных сущностей, показан на рисунке 14.

**Валидация и перенос домена** В качестве обучаемого решения была выбрана энкодерная модель `DeBERTaV3-base` [20], показавших высокую эффективность в задачах извлечения именованных сущностей [24]. Для объективной оценки качества была применена схема валидации в условиях переноса домена, при которой обучение

| **Гиперпараметр** | **DeBERTaV3-base** | **DistilBERT-base-uncased** |
|---|---|---|
| Optimizer | AdamW | AdamW |
| Adam $\beta_1$, $\beta_2$ | 0.9, 0.999 | 0.9, 0.999 |
| Adam $\epsilon$ | 1e-6 | 1e-6 |
| Learning rate (LR) | 5e-6 | 2e-5 |
| Learning rate decay | Linear | Cosine |
| Weight decay | 0.01 | 0.01 |
| Gradient clipping | 1.0 | 1.0 |
| Epochs | 2 | 3.5 |
| Batch size | 2 | 8 |

Таблица 5: Гиперпараметры обучения моделей `DebertaV3-base` и `DistilBERT-base-uncased` на датасетах SPY и Popet соответственно.

модели осуществляется на текстах одного домена, а оценка — на текстах другого. Такой подход снижает риск переобучения на особенностях одного конкретного домена и позволяет получить менее завышенные метрики. Различия в стилистике, лексике и распределении персональных сущностей между доменами приближают экспериментальные условия к реальным сценариям применения, где модель сталкивается с ранее не наблюдавшимися данными. Гиперпараметры, использованные при обучении модели, приведены в Таблице 5.

### 3.1.4 Экспериментальные результаты

В первую очередь была проведена проверка, демонстрирующая, что датасет SPY содержит значительное количество сущностей, не относящихся к персональной информации. Для этого была использована языкова модель из семейства UniNER [34], основанная на архитектуре LLaMA-7B. В частности, модель UniNER-7B-type была предобучена на комбинации двух типов данных: синтетических примерах сгенерированных ChatGPT (Pile-NER-type) и выборки из 40 аннотированных датасетов. Это позволило обеспечить высокую точность детекции и покрытие более 13000 типов сущностей из разнообразных доменов. При этом модель обрабатывает по одному типу сущности за запрос и возвращает структурированный JSON.

Для извлечения сущностей с использованием UniNER-7B-type модели используется следующий шаблон:

```
USER: Text: {входной текст}
ASSISTANT: I've read this text.
USER: What describes {entity-type} in the text?
ASSISTANT: {json-список сущностей}
```

Результаты аннотации приведены моделью UniNER-7B-type в таблице 6. Как видно из таблицы, метрика полноты значительно превышает метрику точности, что свидетельствует о склонности модели ошибочно классифицировать общедоступные имена как PII-сущности. Это наблюдение подтверждается также примером на рисунке 4.

| **Legal Questions** | | | **Medical Consultations** | | |
|---|---|---|---|---|---|
| P | R | F1 | P | R | F1 |
| 21.5 | 89.5 | 34.7 | 21.7 | 80.4 | 34.1 |

Таблица 6: Результаты оценки UniNER на датасете SPY. Метрики рассчитаны для сущностей типа `name`, с использованием промпта из оригинальной статьи UniNER [34]

Hi all, I'm Nuria Batista, reaching out because I'm in a bit of a tricky situation and I'm hoping someone with legal expertise can offer some guidance. I'm a marketing coordinator at an advertising agency, and one of our clients is accusing us of breach of contract. My team and I have reviewed the contract thoroughly, and we're confident that we've met all of the requirements. However, the client is still pushing for a refund and is threatening to take legal action against me, specifically at the office of attorney Emily Brown, located at 123 Main St, San Francisco, CA 94105.

Рис. 4: Имя Nuria Batista было правильно классифицировано как персональные данные, тогда как Emily Brown было ошибочно отнесено к PII-сущностям, поскольку модель UniNER не различает персональные и общедоступные именованные сущности.

Следуя предложенной методологии на рисунке 2, было сгенерировано два набора данных из юридического и медицинского доменов. Таблица 7 показывает, как различные модели справляются с задачей обнаружения персональных данных на наборе данных SPY. Модель Presidio имеет значительно более низкую точность по сравнению с полнотой для всех категорий, что означает, что она ошибочно классифициро-

| Entity | | Legal Questions | | | Medical Consultations | | |
|---|---|---|---|---|---|---|---|
| | | Llama-3 | Presidio | DeBERTa | Llama-3 | Presidio | DeBERTa |
| Name | P | 64.7 | 17.9 | **87.4** | 73.0 | 17.1 | **86.9** |
| | R | 68.9 | 79.4 | **93.2** | 62.9 | 80.4 | **88.7** |
| | F1 | 66.7 | 29.2 | **90.2** | 67.6 | 28.2 | **87.8** |
| Email | P | 91.8 | 33.7 | **92.1** | 92.7 | 37.6 | **97.6** |
| | R | 88.5 | 91.8 | **99.1** | 90.9 | 92.2 | **99.5** |
| | F1 | 90.1 | 49.3 | **95.5** | 91.8 | 53.4 | **98.5** |
| Username | P | 66.1 | - | **90.3** | 68.8 | - | **92.1** |
| | R | 59.7 | - | **98.0** | 70.4 | - | **95.4** |
| | F1 | 62.7 | - | **94.0** | 69.6 | - | **93.8** |
| URL | P | 84.5 | 7.9 | **94.4** | 83.6 | 6.9 | **97.5** |
| | R | 92.5 | 21.3 | **99.0** | 91.9 | 19.4 | **98.9** |
| | F1 | 88.3 | 11.5 | **96.7** | 87.5 | 10.2 | **98.2** |
| ID | P | 91.9 | 20.6 | **93.0** | 91.7 | 26.1 | **96.7** |
| | R | 62.2 | 34.4 | **96.6** | 75.1 | 38.9 | **98.3** |
| | F1 | 74.2 | 25.8 | **94.8** | 82.6 | 31.2 | **97.5** |
| Phone | P | 85.7 | 34.1 | **87.5** | 89.8 | 37.4 | **93.3** |
| | R | 92.8 | 68.1 | **98.7** | 90.0 | 65.5 | **96.9** |
| | F1 | 89.1 | 45.4 | **92.8** | 89.9 | 47.6 | **95.0** |
| Address | P | **93.7** | - | 88.3 | **96.2** | - | 89.3 |
| | R | 81.3 | - | **94.5** | 90.4 | - | **95.1** |
| | F1 | 87.1 | - | **91.3** | **93.2** | - | 92.1 |

Таблица 7: Метрики качества моделей на различных доменах и типах сущностей датасета SPY, где P — точность (Precision), R — полнота (Recall), F1 — F-мера. **Presidio** — инструмент от Microsoft для быстрого выявления PII с использованием NER и регулярных выражений. **LLaMA-3** — модель LLaMA-3-70B, применяемая в zero-shot режиме для задачи выявления PII, с промптом на рисунке 14. **DeBERTa** — модель, обученная в режиме переноса домена. Пропуски в таблице означают, что соответствующий класс сущностей не поддерживается моделью.

вала значительную часть общедоступных данных как PII. Еще одним наблюдением является то, что модель LLaMA-3-70B стабильно превосходит Presidio, что можно

объяснить ее способностью понимать контекст в котором упоминаются именованные сущности.

Модель DeBERTav3-base, провалидированная на наборе данных SPY в условиях доменного трансфера, способна точнее обнаруживать PII-сущности по сравнению с zero-shot методами, показывая значительно более высокую метрику F1-меры.

## 3.2 Датасет PoPet для задачи распознавания именованных сущностей

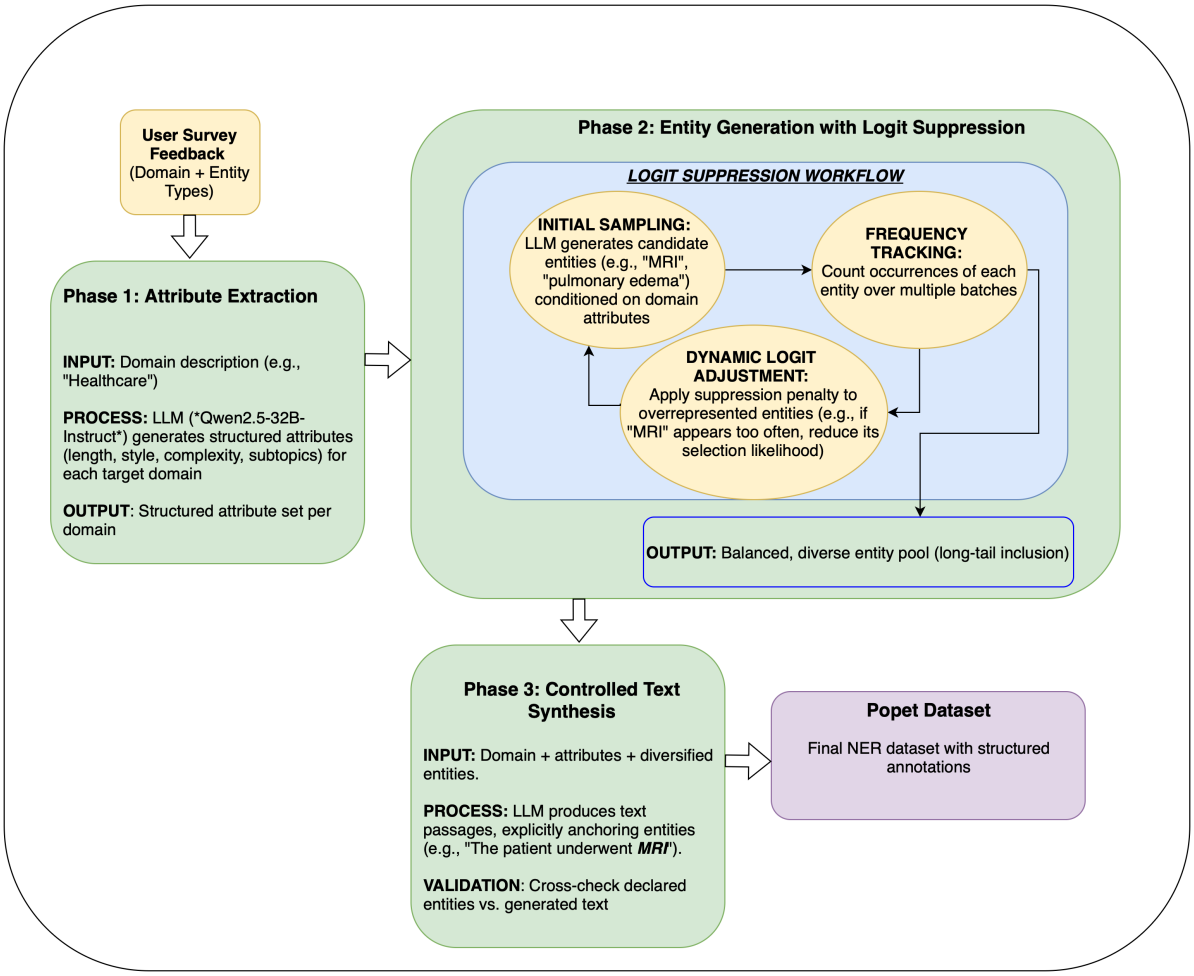


Рис. 5: Методология генерации датасета Popet.

Существующие синтетические датасеты, как правило, создаются без учёта реальных потребностей конечных пользователей — в частности, игнорируется, предметные области и типы сущностей, которые представляют наибольший практический интерес. В отличие от них, разработанный в рамках данной работы набор данных

Popet ориентирован на прикладное применение и нацелен на максимальное разнообразие. Это разнообразие обеспечивается как на уровне охвата предметных областей и типов сущностей, так и на уровне конкретных примеров, используемой лексики и стилистики текстов. Такой подход позволяет получить более универсальный и реалистичный корпус.

### 3.2.1 Методология генерации

Сводная схема предложенной методологии приведена на рисунке 5. В ней отражены все ключевые этапы: выбор предметной области, определение атрибутов текстов, генерация сущностей с использованием техники подавления логитов и финальный этап синтеза текста с выбранными сущностями. В качестве основной модели используется Qwen2.5-32B-Instruct [36], параметры генерации описаны в разделе экспериментов. Все используемые при генерации датасета промпты приведены в приложении 5.1.

**Алгоритм 3.1** Формирование словаря `logit_bias` для подавления частотных токенов. На вход подаётся: список текстов $T$, число наиболее частых токенов $top_k$, множитель штрафа $penalty_multiplier$, и смещение штрафа $penalty_bias$.

**Вход:** $texts, top_k, penalty_multiplier, penalty_bias$

**Выход:** $logit_bias$

$tokenized_texts := \texttt{Tokenizer}(texts)$

$all_tokens := \bigcup tokenized_texts$

$token_counts := \texttt{Counter}(all_tokens)$

$total_tokens := \sum_{t \in token_counts} token_counts[t]$

$logit_bias := \{\}$

**для** каждой пары $(token_id,\ count)$ в $token_counts.\texttt{most_common}(top_k)$

  $p_t := \frac{count}{total_tokens} \cdot 100$

  $s_t := \max(penalty_multiplier \cdot p_t,\ penalty_bias)$

  $logit_bias[token_id] := s_t$

**return** $logit_bias$

**Определение доменных признаков и формирование атрибутов.** На первом этапе модель получает на вход название предметной области и генерирует список

релевантных атрибутов, описывающих формальные и стилистические особенности текстов из заданного домена. Среди таких атрибутов — длина документа, структура (например, наличие заголовков и подзаголовков), тональность, уровень формальности, наличие профессиональной лексики, и другие.

Для каждого типа атрибутов модель синтезирует примеры возможных значений. Причем происходит формировании целой комбинации примеров значений атрибутов на основе их совместимости друг с другом. Такой подход позволяет получить разнообразный и достоверный набор параметров, который впоследствии используется для генерации синтетических сущностей и полноценных текстов.

Для каждого типа атрибутов модель синтезирует не отдельные значения, а целостные комбинации значений, совместимых между собой. Это достигается за счёт генерации всех атрибутов одновременно, в рамках одного запроса к языковой модели. Например, для домена потребительских отзывов модель может сгенерировать следующую комбинацию атрибутов: `Категория = Бытовая техника, Товар = Стиральная машина, Бренд = Samsung, Оценка = 4, Тональность = Позитивная`. Такое сочетание представляет собой логически согласованную конфигурацию, подходящую для генерации текста отзыва.

**Генерация разнообразных сущностей с учётом доменных признаков.** На втором этапе осуществляется генерация сущностей, соответствующих таксономии на рисунке 6. При генерации учитываются текстовые атрибуты, сформированные на предыдущем этапе, что позволяет адаптировать сущности согласно стилистическим и тематическим особенностям каждой предметной области.

Для повышения разнообразия генерируемых сущностей реализован механизм динамического подавления логитов часто встречающихся токенов, описанный в алгоритме 3.1. Данный алгоритм основывается на идее штрафования часто используемых токенов, стимулируя модель выбирать менее тривиальные, но логически корректные альтернативы. Суть разработанного алгоритм заключается в следующем:

- На каждой итерации генерации отбираются $top_k$ наиболее частотных токенов
- Для каждого такого токена $t$ с долей $p_t$ (в процентах от общего числа токенов) логит корректируется по следующей формуле:

$$s_t = \max(penalty_multiplier \cdot p_t,\ penalty_bias),$$

- Обновленное значение логитов используется для вычисления нового вероятностного распределения токенов.

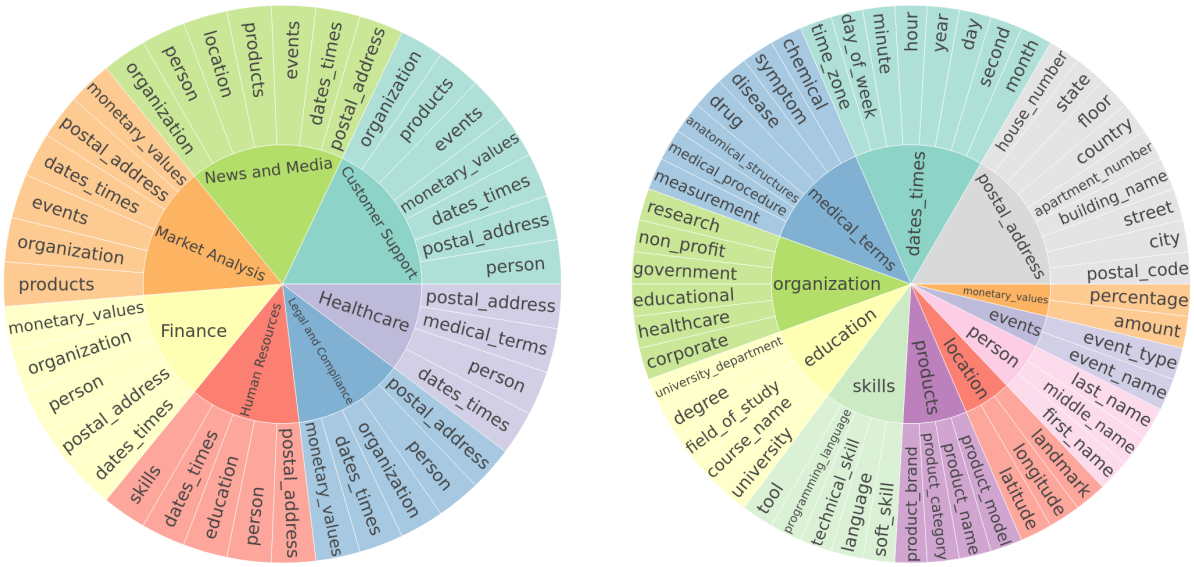


Рис. 6: Иерархическая структура доменов и типов сущностей в датасете Popet разработанная согласно опросам пользователей фреймворка DeepPavlov.

**Сравнение с подходом ProgGen** Методология ProgGen [30] предполагает генерацию множества сущностей путём многократных независимых вызовов языковой модели для одного и того же набора текстовых атрибутов. Все запросы выполняются в идентичных условиях, что ограничивает разнообразие получаемых сущностей: модель имеет тенденцию воспроизводить одни и те же примеры при повторных семплированиях. Кроме того, генерация более чем нескольких сотен уникальных сущностей за один проход затруднена из-за ограничений длины входа и свойств вероятностного семплирования современных LLM.

В отличие от ProgGen, предложенный в данной работе метод использует итеративную генерацию сущностей с динамическим подавлением логитов высокочастотных токенов. Процесс продолжается до тех пор, пока не будет исчерпано множество релевантных сущностей, допустимых в контексте заданной конфигурации текстовых атрибутов. Таким образом, обеспечивается более полное покрытие возможных значений и существенно повышается разнообразие синтетических сущностей.

**Формирование текстовых фрагментов согласно заданным атрибутам и сущностям.** На заключительном этапе осуществляется генерация текстов с ранее сгенерированными атрибутами и соответствующми им сущностями (см. рисунок 18). После генерации модель дополнительно анализирует текст, выявляя, какие сущности были включены и как они были переформулированы (см. рисунок 20). Эта информация используется для валидации корректности разметки.

В отличие от методологии генерации датасета SPY, в которой используются шаблоны с маркерами, позднее заменяемыми на сущности, предложенный подход обеспечивает более естественную интеграцию сущностей уже на этапе генерации. Это позволяет сохранить грамматическую согласованность слов и стилистическую целостность текста, что особенно важно при создании высококачественных синтетических данных.

### 3.2.2 Анализ данных

Итоговый синтетический датасет включает более 40 тысяч сущностей, распределённых по 52 типам. Наиболее частые классы — это организации, почтовые адреса и даты/время (см. рисунки 7). При этом иерархическая структура охватывает широкий спектр сущностей, включая редко представленные категории, такие как язык или географическая долгота. Данные охватывают 7 тематических доменов, среди которых наибольшее количество примеров относится к областям "Legal and Compliance“Healthcare” и “Finance” (см. рисунок 8).

### 3.2.3 Постановка экспериментов

Для проведения экспериментов исходный датасет был предварительно разбит в пропорции 70%/10%/20% на обучающую, валидационную и тестовую выборки.

В качестве первой базовой модели для задачи NER использовалась специализированная языковая модель UniNER-7B-type [34], дообученная на задачу распознавания именованных сущностей. Модель применялась в zero-shot режиме: каждая сущность обрабатывалась отдельно с помощью диалогового промпта, показанного на рисунке 19.

Второй базовой моделью является Qwen2.5-32B [36]. Для неё был составлен 5-shot промпт, где каждому примеру сопоставлялось краткое описание сущности. Как показано на рисунке 21, такой формат инструкции позволял извлекать все целевые

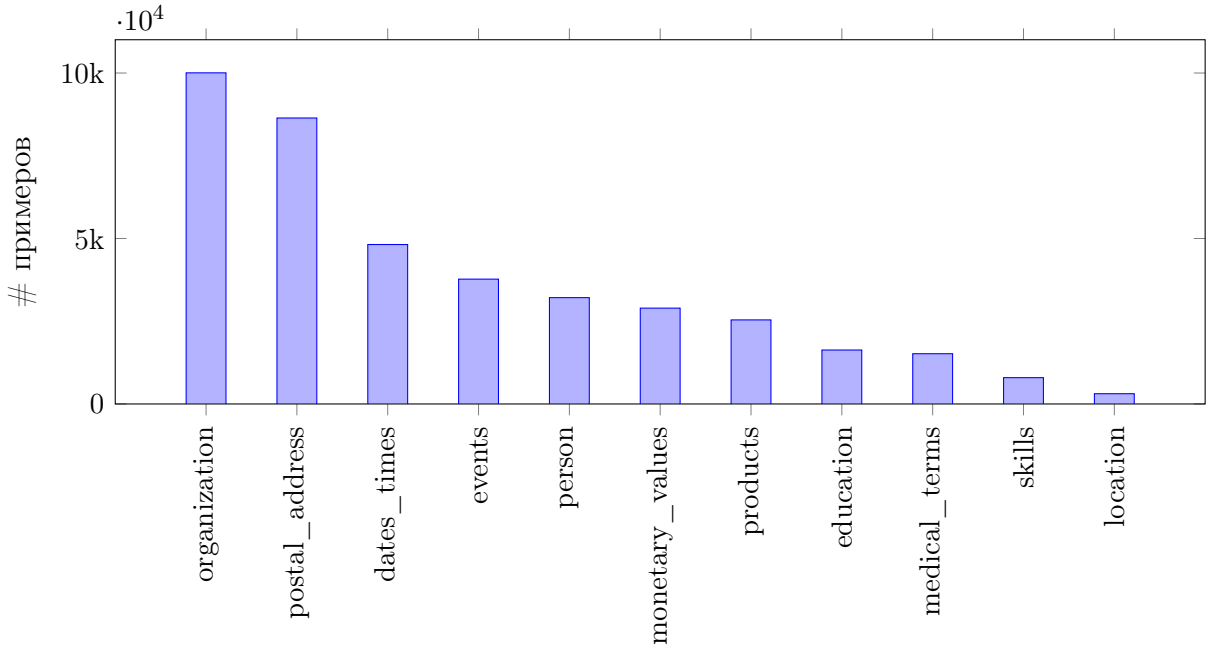


Рис. 7: Распределение крупно-гранулярных типов сущностей в датасете Popet.

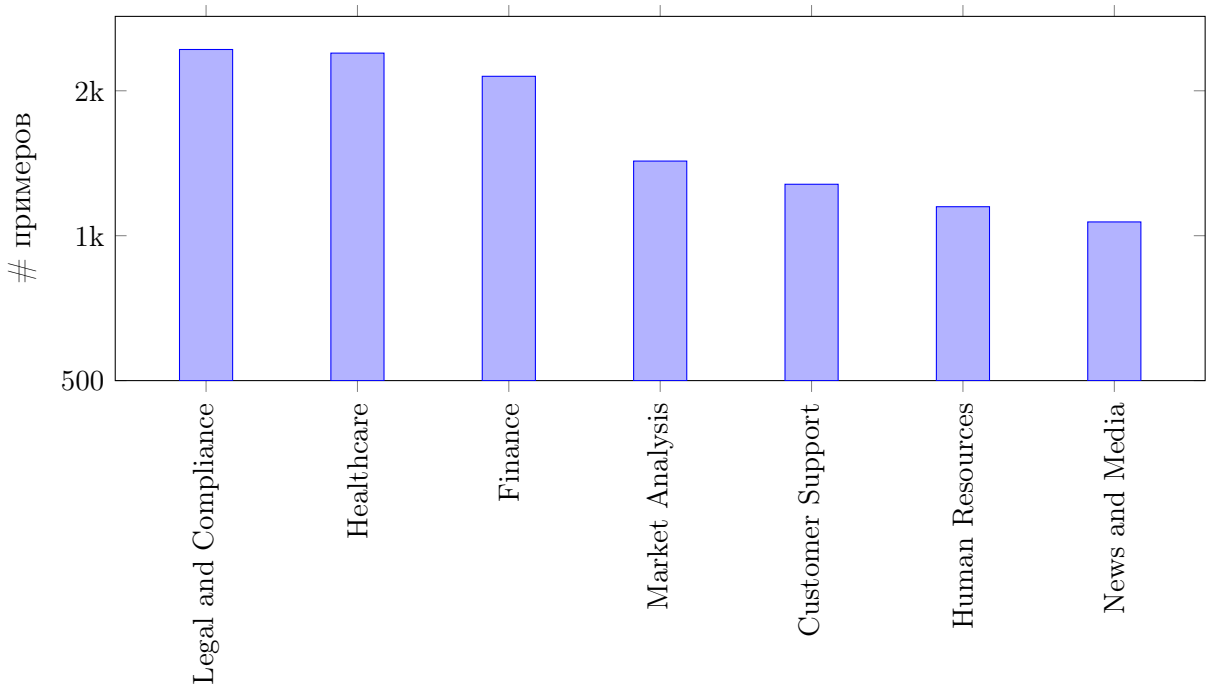


Рис. 8: Распределение доменов в датасете Popet.

классы сущностей за один вызов модели.

Для оценки качества разметки энкодерная модель DistilBERT были обучена в supervised режиме на тренировочной выборке. Подбор гиперпараметров обучения производился на валидационной выборке; значения гиперпараметров приведены в таблице 5.

### 3.2.4 Экспериментальные результаты

Результаты валидации различных моделей на датасете Popet представлены в таблице 8. Наиболее высокие значения метрики F1-score в большинстве доменов демонстрирует модель distilbert-base-uncased, дообученная в supervised режиме на тренировочной части датасета. Эта модель показала наилучшие результаты в шести из семи доменов, исключением стал домен **News**, где наивысшее значение (64.49) продемонстрировала модель Qwen2.5-32B. Модель Uniner-7B-type, применённая в zero-shot режиме, систематически продемонстрировала наихудшие значения F1-score во всех доменах. Во многом такая просадка может быть обусловлена спецификой датасета Popet: наличие большого количества мелкогранулярных классов, под которые Uniner изначально не адаптирована, существенно затрудняет обобщение в условиях отсутствия дообучения. В то же время модель Qwen2.5-32B за счёт значительно большего размера и развитых reasoning-способностей способна частично компенсировать отсутствие дообучения.

Таким образом, полученные результаты подтверждают выводы предыдущих исследований [28, 32], согласно которым дообучение энкодеров на целевой задаче с использованием размеченных данных обеспечивает более высокую эффективность по сравнению с применением больших языковых моделей в zero-shot или few-shot режимах. Это подчёркивает, что качество и релевантность обучающей выборки остаются ключевыми факторами при построении эффективных NER-моделей.

| **Модель** | **Режим** | News | Legal | Support | Health | Market | Finance | HR |
|---|---|---|---|---|---|---|---|---|
| distilbert-base-uncased [8] | SFT | 55.09 | **91.59** | **82.41** | **73.5** | **74.80** | **63.35** | **53.36** |
| Uniner-7B-type [34] | 0-shot | 29.01 | 38.06 | 19.56 | 22.65 | 28.35 | 25.6 | 21.44 |
| Qwen2.5-32B [31] | 5-shot | **64.49** | 75.00 | 75.00 | 58.56 | 68.43 | 53.11 | 50.59 |

Таблица 8: Значение метрики micro F1-score на доменах датасета Popet.

# 4 Заключение

В данной секции подводятся итоги проведённого исследования, обобщаются достигнутые результаты и формулируются направления для будущих экспериментов.

**Основные результаты работы:**

- Предложен новый подход к генерации синтетических датасетов для задачи распознавания именованных сущностей, включая подзадачу детекции персональных данных. Доказано, что синтетические данные, полученные по предложенной методологии, позволяют эффективно обучать NER-модели в условиях ограниченного доступа к реальным размеченным корпусам.

- Разработан и реализован специализированный датасет **SPY** для задач детекции персональных данных. Он обеспечивает чёткое разграничение между персональными и общедоступными сущностями, контролируемую плотность разметки и сбалансированное распределение по классам. Такой датасет особенно актуален для сценариев, где конфиденциальность играет ключевую роль.

- Создан обобщённый NER-корпус **PoPet** с расширенной иерархией классов: более чем 50 классов сущностей и семь предметных областей. Корпус ориентирован на обеспечение лексического и стилистического разнообразия и позволяет моделям адаптироваться к широкому спектру доменов и форматов текстов. Достигаемый уровень контроля за характеристиками синтетических текстов значительно превосходит возможности ручной или автоматической разметки реальных данных.

- Проведены вычислительные эксперименты с обучением энкодерных моделей на сгенерированных данных. Установлено, что модели, обученные исключительно на синтетических корпусах, демонстрируют высокие значения метрик полноты и точности, превосходя few-shot подходы. Также показано, что модели, обученные на одном домене, сохраняют обобщающую способность при переносе на тексты из другого домена, что делает их применимыми в реальных сценариях с высокой вариативностью входных текстов.

Таким образом, результаты работы подтверждают применимость синтетических датасетов как полноценной альтернативы ручной разметке в задачах NER. Разработанные подходы могут быть масштабированы на другие языки, домены и типы

сущностей, а также интегрированы в более широкие системы обработки текстовых данных.

**Будущие направления исследований** В настоящей работе остается ряд направлений для будущего изучения и развития. В частности, необходима комплексная оценка обобщающей способности моделей, обученных на синтетических данных, при переносе на сторонние датасеты с ручной разметкой. Это позволит установить границы применимости разработанной методологии.

Несмотря на полученное разнообразие сгенерированных данных за счёт варьирования доменов, стилей и типовых структур, синтетические тексты существенно уступают реальным по естественности. В большинстве случаев они сохраняют формально правильную структуру, которая слабо отражает характерные особенности настоящих пользовательских и технических текстов. Реальные данные часто включают фрагменты с нарушенной грамматикой, нестандартным форматированием, вставками кода, HTML-разметкой, комментариями, а также различными полуструктурированными конструкциями, что делает задачу извлечения сущностей в них существенно более сложной. Отсутствие таких элементов в текущей версии синтетических датасетов ограничивает способность моделей, обученных на них, адаптироваться к сложным случаям. В дальнейшей работе представляется целесообразным сосредоточиться на включении в процесс генерации намеренно «зашумлённых» и нерегулярных текстов, имитирующих реальные сценарии использования.

Дополнительный интерес представляет исследование масштабируемости предложенного подхода — в частности, анализ зависимости качества итоговых моделей от объёма синтетического корпуса и выявление возможной точки насыщения, за пределами которой прирост метрик становится незначительным.

# Список литературы


1. *Sang E. F. T. K.*, *Meulder F. D.* Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition // CoRR. — 2003. — T. cs.CL/0306050. — URL: http://arxiv.org/abs/cs/0306050.
2. *Sang E. F. T. K.*, *Meulder F. D.* Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition // Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003 / под ред. W. Daelemans, M. Osborne. — ACL, 2003. — С. 142—147. — URL: https://aclanthology.org/W03-0419/.
3. Named Entity Recognition in Wikipedia / D. Balasuriya, N. Ringland, J. Nothman, T. Murphy, J. R. Curran // Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources (People's Web) / под ред. I. Gurevych, T. Zesch. — Suntec, Singapore : Association for Computational Linguistics, 08.2009. — С. 10—18. — URL: https://aclanthology.org/W09-3302/.
4. Towards Robust Linguistic Analysis using OntoNotes / S. Pradhan, A. Moschitti, N. Xue, H. T. Ng, A. Björkelund, O. Uryupina, Y. Zhang, Z. Zhong // Proceedings of the Seventeenth Conference on Computational Natural Language Learning, CoNLL 2013, Sofia, Bulgaria, August 8-9, 2013 / под ред. J. Hockenmaier, S. Riedel. — ACL, 2013. — С. 143—152. — URL: https://aclanthology.org/W13-3516/.
5. *Faraglia D.* Faker. — 2014. — URL: https://github.com/joke2k/faker.
6. Results of the WNUT2017 Shared Task on Novel and Emerging Entity Recognition / L. Derczynski, E. Nichols, M. van Erp, N. Limsopatham // Proceedings of the 3rd Workshop on Noisy User-generated Text / под ред. L. Derczynski, W. Xu, A. Ritter, T. Baldwin. — Copenhagen, Denmark : Association for Computational Linguistics, 09.2017. — С. 140—147. — DOI: 10.18653/v1/W17-4418. — URL: https://aclanthology.org/W17-4418/.
7. *Nakayama H.* seqeval: A Python framework for sequence labeling evaluation. — 2018. — URL: https://github.com/chakki-works/seqeval ; Software available from https://github.com/chakki-works/seqeval.

8. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter / V. Sanh, L. Debut, J. Chaumond, T. Wolf. — 2020. — arXiv: 1910.01108 [cs.CL]. — URL: https://arxiv.org/abs/1910.01108.

9. *Johnson A. E. W.*, *Bulgarelli L.*, *Pollard T. J.* Deidentification of free-text medical records using pre-trained bidirectional transformers // ACM CHIL '20: ACM Conference on Health, Inference, and Learning, Toronto, Ontario, Canada, April 2-4, 2020 [delayed] / под ред. M. Ghassemi. — ACM, 2020. — С. 214—221. — DOI: 10.1145/3368555.3384455. — URL: https://doi.org/10.1145/3368555.3384455.

10. Language Models are Few-Shot Learners / T. B. Brown [и др.] // Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual / под ред. H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin. — 2020. — URL: https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

11. Few-NERD: A Few-shot Named Entity Recognition Dataset / N. Ding, G. Xu, Y. Chen, X. Wang, X. Han, P. Xie, H. Zheng, Z. Liu // Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) / под ред. C. Zong, F. Xia, W. Li, R. Navigli. — Online : Association for Computational Linguistics, 08.2021. — С. 3198—3213. — DOI: 10.18653/v1/2021.acl-long.248. — URL: https://aclanthology.org/2021.acl-long.248/.

12. MasakhaNER: Named Entity Recognition for African Languages / D. I. Adelani [и др.] // Trans. Assoc. Comput. Linguistics. — 2021. — Т. 9. — С. 1116—1131. — DOI: 10.1162/TACL_A_00416. — URL: https://doi.org/10.1162/tacl%5C_a%5C_00416.

13. *Microsoft*. Presidio. — 2021. — URL: https://microsoft.github.io/presidio/.

14. The Pile: An 800GB Dataset of Diverse Text for Language Modeling / L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, S. Presser, C. Leahy // CoRR. — 2021. — Т. abs/2101.00027. — arXiv: 2101.00027. — URL: https://arxiv.org/abs/2101.00027.

15. WikiNEuRal: Combined Neural and Knowledge-based Silver Data Creation for Multilingual NER / S. Tedeschi, V. Maiorca, N. Campolungo, F. Cecconi, R. Navigli // Findings of the Association for Computational Linguistics: EMNLP 2021 / под ред. M.-F. Moens, X. Huang, L. Specia, S. W.-t. Yih. — Punta Cana, Dominican Republic : Association for Computational Linguistics, 11.2021. — С. 2521—2533. — DOI: 10.18653/v1/2021.findings-emnlp.215. — URL: https://aclanthology.org/2021.findings-emnlp.215/.

16. Extended Overview of HIPE-2022: Named Entity Recognition and Linking in Multilingual Historical Documents / M. Ehrmann, M. Romanello, S. Najem-Meyer, A. Doucet, S. Clematide // Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022. Т. 3180 / под ред. G. Faggioli, N. Ferro, A. Hanbury, M. Potthast. — CEUR-WS.org, 2022. — С. 1038—1063. — (CEUR Workshop Proceedings). — URL: https://ceur-ws.org/Vol-3180/paper-83.pdf.

17. *Tedeschi S.*, *Navigli R.* MultiNERD: A Multilingual, Multi-Genre and Fine-Grained Dataset for Named Entity Recognition (and Disambiguation) // Findings of the Association for Computational Linguistics: NAACL 2022 / под ред. M. Carpuat, M.-C. de Marneffe, I. V. Meza Ruiz. — Seattle, United States : Association for Computational Linguistics, 07.2022. — С. 801—812. — DOI: 10.18653/v1/2022.findings-naacl.60. — URL: https://aclanthology.org/2022.findings-naacl.60/.

18. The Text Anonymization Benchmark (TAB): A Dedicated Corpus and Evaluation Framework for Text Anonymization / I. Pilán, P. Lison, L. Øvrelid, A. Papadopoulou, D. Sánchez, M. Batet // Comput. Linguistics. — 2022. — Т. 48, № 4. — С. 1053—1101. — DOI: 10.1162/COLI_A_00458. — URL: https://doi.org/10.1162/coli%5C_a%5C_00458.

19. From Sparse to Dense: GPT-4 Summarization with Chain of Density Prompting / G. Adams, A. R. Fabbri, F. Ladhak, E. Lehman, N. Elhadad // CoRR. — 2023. — Т. abs/2309.04269. — DOI: 10.48550/ARXIV.2309.04269. — arXiv: 2309.04269. — URL: https://doi.org/10.48550/arXiv.2309.04269.

20. *He P.*, *Gao J.*, *Chen W.* DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing // The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda,

May 1-5, 2023. — OpenReview.net, 2023. — URL: https://openreview.net/forum?id=sE7-XhLxHA.

21. *Project B.* BigCode PII Dataset. — 2023. — Accessed: 2025-05-02. https://huggingface.co/datasets/bigcode/bigcode-pii-dataset.

22. StarCoder: may the source be with you! / R. Li [и др.] // Trans. Mach. Learn. Res. — 2023. — Т. 2023. — URL: https://openreview.net/forum?id=KoFOg41haE.

23. The Stack: 3 TB of permissively licensed source code / D. Kocetkov, R. Li, L. B. Allal, J. Li, C. Mou, Y. Jernite, M. Mitchell, C. M. Ferrandis, S. Hughes, T. Wolf, D. Bahdanau, L. von Werra, H. de Vries // Trans. Mach. Learn. Res. — 2023. — Т. 2023. — URL: https://openreview.net/forum?id=pxpbTdUEpD.

24. *Tirskikh D., Konovalov V.* Zero-Shot NER via Extractive Question Answering // Advances in Neural Computation, Machine Learning, and Cognitive Research VII / под ред. B. Kryzhanovsky, W. Dunin-Barkowski, V. Redko, Y. Tiumentsev, V. Klimov. — Cham : Springer Nature Switzerland, 2023. — С. 22—31. — ISBN 978-3-031-44865-2. — DOI: 10.1007/978-3-031-44865-2_3.

25. *AI@Meta.* Llama 3 Model Card. — 2024. — URL: https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

26. Better Synthetic Data by Retrieving and Transforming Existing Datasets / S. Gandhi, R. Gala, V. Viswanathan, T. Wu, G. Neubig // Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024 / под ред. L. Ku, A. Martins, V. Srikumar. — Association for Computational Linguistics, 2024. — С. 6453—6466. — DOI: 10.18653/V1/2024.FINDINGS-ACL.385. — URL: https://doi.org/10.18653/v1/2024.findings-acl.385.

27. GLiNER: Generalist Model for Named Entity Recognition using Bidirectional Transformer / U. Zaratiana, N. Tomeh, P. Holat, T. Charnois // Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024 / под ред. K. Duh, H. Gómez-Adorno, S. Bethard. — Association for Computational Linguistics, 2024. — С. 5364—5376. — DOI: 10.18653/V1/2024.NAACL-LONG.300. — URL: https://doi.org/10.18653/v1/2024.naacl-long.300.

28. NuNER: Entity Recognition Encoder Pre-training via LLM-Annotated Data / S. Bogdanov, A. Constantin, T. Bernard, B. Crabbé, E. Bernard // Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024 / под ред. Y. Al-Onaizan, M. Bansal, Y. Chen. — Association for Computational Linguistics, 2024. — С. 11829—11841. — URL: https://aclanthology.org/2024.emnlp-main.660.

29. NuNER: Entity Recognition Encoder Pre-training via LLM-Annotated Data / S. Bogdanov, A. Constantin, T. Bernard, B. Crabbé, E. Bernard // Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024 / под ред. Y. Al-Onaizan, M. Bansal, Y. Chen. — Association for Computational Linguistics, 2024. — С. 11829—11841. — URL: https://aclanthology.org/2024.emnlp-main.660.

30. ProgGen: Generating Named Entity Recognition Datasets Step-by-step with Self-Reflexive Large Language Models / Y. Heng, C. Deng, Y. Li, Y. Yu, Y. Li, R. Zhang, C. Zhang // Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024 / под ред. L. Ku, A. Martins, V. Srikumar. — Association for Computational Linguistics, 2024. — С. 15992—16030. — DOI: 10.18653/V1/2024.FINDINGS-ACL.947. — URL: https://doi.org/10.18653/v1/2024.findings-acl.947.

31. Qwen2 Technical Report / A. Yang [и др.] // arXiv preprint arXiv:2407.10671. — 2024.

32. *Tang Y.*, *Hasan R.*, *Runkler T. A.* FsPONER: Few-Shot Prompt Optimization for Named Entity Recognition in Domain-Specific Scenarios // ECAI 2024 - 27th European Conference on Artificial Intelligence, 19-24 October 2024, Santiago de Compostela, Spain - Including 13th Conference on Prestigious Applications of Intelligent Systems (PAIS 2024). Т. 392 / под ред. U. Endriss, F. S. Melo, K. Bach, A. J. B. Diz, J. M. Alonso-Moral, S. Barro, F. Heintz. — IOS Press, 2024. — С. 3757—3764. — (Frontiers in Artificial Intelligence and Applications). — DOI: 10.3233/FAIA240936. — URL: https://doi.org/10.3233/FAIA240936.

33. The Learning Agency Lab - PII Data Detection / H. Langdon, C. Scott, B. Perpetual, K. Jules, B. Lauryn, D. Maggie, H. Ryan, R. Walter, H. Addison. — 2024. — URL: https://kaggle.com/competitions/pii-detection-removal-from-educational-data.

34. UniversalNER: Targeted Distillation from Large Language Models for Open Named Entity Recognition / W. Zhou, S. Zhang, Y. Gu, M. Chen, H. Poon // The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024. — OpenReview.net, 2024. — URL: https://openreview.net/forum?id=r65xfUb76p.

35. *Ai4Privacy*. Open PII Masking 500k Dataset. — 2025. — DOI: 10.57967/hf/4852. — URL: https://huggingface.co/datasets/ai4privacy/open-pii-masking-500k-ai4privacy.

36. Qwen2.5 Technical Report / Qwen [и др.]. — 2025. — arXiv: 2412.15115 [cs.CL]. — URL: https://arxiv.org/abs/2412.15115.

37. *Savkin M.*, *Ionov T.*, *Konovalov V.* SPY: Enhancing Privacy with Synthetic PII Detection Dataset // Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 4: Student Research Workshop, Albuquerque, NM, USA, April 30 - May 1, 2025 / под ред. A. Ebrahimi, S. Haider, E. Liu, S. Haider, M. L. Pacheco, S. Wein. — Association for Computational Linguistics, 2025. — С. 236—246. — URL: https://aclanthology.org/2025.naacl-srw.23/.

# 5 Приложения

## 5.1 Промпты

В данном разделе описаны инструкции использорвавшиеся для запуска каждого шага генерации датасета и для запуска базовых LLM решений.

Generate a biography of a fictional man named <***generated-name-goes-here***>.

Occupation: any average job you can come up with
Personality: describe in 5 sentences

Present results in json format with fields “occupation”: str, “personality”: str

Рис. 9: Подсказка для генерации биографии в датасете **SPY**. Маркер “<generated-name-goes-here>” заменяется на случайное имя.

Step 1) Look through the personality of the text author and pretend to be that person.

occupation: <***generated-occupation***>
personality: <***generated-personality***>

Step 2) Use the following instructions to generate a text:

<***domain-specific-instructions***>

Requirements:
- At any circumstance do not include any personal information in generated text.

Respond only with generated text with no commentary. Here goes your text:

Рис. 10: Промпт для генерации текстов, не содержащих персональную информацию. Маркеры “<generated-*>” и “<domain-specific-instructions>” заменяются соответствующими описаниями. Датасет **SPY**.

**Text:** {}

**Task:** You are given a Text, which contains author's personal information. Your task is to add new entities, which are not related to the text author. Generate entities using the following classes: name, email, username, phone number, url, address, identifier.

**Requirements:**
- At any circumstance DO NOT change author's personal information in the above text
- Newly generated entities should not disclose the personal information of the author of the text

Respond only with updated text with no commentary. Here goes an updated text:

Рис. 11: Промпт для добавления в тексты датасета **SPY** именованных сущностей, не относящихся к автору текста.

**Text:** {}

**Task:** You are an author of the above Text. Your task is to add new placeholders in the Text from the list below. You will be penalized for mentioning any placeholders other than what is listed below!

**Here is the list of placeholders representing your personal information:**

<author_personal_name> - A full or partial name of the text author

<author_personal_email> - An author's email address

<author_personal_username> - An author's username on any website, social media etc.

<author_personal_phone_number> - A phone number associated with the author or his relatives

<author_personal_url> - A link to author's social media page or personal website

<author_personal_address> - A full or partial street address that is associated with the author, such as home address

<author_personal_identifier> - A number or sequence of characters that could be used to identify an author, such as a social security number or medical policy number

**Requirements:**

- Do NOT change existing placeholders
- Distribute placeholders evenly throughout your text, do not stack them all in one place
- New text must be more entity-dense than the previous one

Respond only with updated text with no commentary. Here goes an updated text:

Рис. 12: Промпт для добавления маркеров заменяющих персональные данные в текст. Датасет **SPY**.

**Extract the following personal information entities from the provided text, ensuring that only personally identifiable information (PII) related to the author of the text is captured:**

- **Person:** Names of the author. Do not include names of other people, famous authors, celebrities, or historical figures.
- **Email:** Personal email addresses of the author.
- **Phone:** Personal phone numbers of the author.
- **ID:** Personal identification numbers of the author (e.g., Social Security Number, passport number).
- **URL:** URLs that are personal to the author and lead to pages containing personal data (e.g., the author's personal blogs, social media profiles).
- **Username:** Personal usernames of the author for online platforms.
- **Address:** Personal home addresses of the author.

**Text:** {text}

**Format your response in JSON as follows:**
{ "person": ["list of personal names"], "email": ["list of personal emails"], "phone": ["list of personal phone numbers"], "id": ["list of personal IDs"], "url": ["list of personal URLs"], "username": ["list of personal usernames"], "address": ["list of personal addresses"] }

If there is no information for a particular category, return an empty list for that category.

Рис. 13: Промпт для извлечения заданного списка персональных данных из текста. Датасет **SPY**.

Extract the following personal information entities from the provided text, ensuring that only personally identifiable information (PII) related to the author of the text is captured:

- **Person:** Names of the author. Do not include names of other people, famous authors, celebrities, or historical figures.
- **Email:** Personal email addresses of the author.
- **Phone:** Personal phone numbers of the author.
- **ID:** Personal identification numbers of the author (e.g., Social Security Number, passport number).
- **URL:** URLs that are personal to the author and lead to pages containing personal data (e.g., the author's personal blogs, social media profiles).
- **Username:** Personal usernames of the author for online platforms.
- **Address:** Personal home addresses of the author.

Text: "text"

Format your response in JSON as follows:
{{ "person": ["list of the author's personal names"],
"email": ["list of the author's personal emails"],
"phone": ["list of the author's personal phone numbers"],
"id": ["list of the author's personal IDs"],
"url": ["list of the author's personal URLs"],
"username": ["list of the author's personal usernames"],
"address": ["list of the author's personal addresses"]
}}

If there is no information for a particular category, return an empty list for that category.

Рис. 14: Промпт для извлечения PII-сущностей из текста моделью LLaMA-3-70B. Датасет **SPY**.

Which attribute dimensions do you consider vital for an article in {domain_name} domain?

For example the appropriate attributes for news domain may be: ['Topic', 'Writing Style', 'Location', 'Length', ...]

Рис. 15: Промпт для генерации атрибутов текста. Датасет **PoPet**.

What value should I estimate for these attributes? Answer with json.

Example of the correctly formatted output:

{ 'Topic': ['Politics', 'Sports', ... ],

'Text Style': ['Article', 'Social media post', ... ] }

Рис. 16: Промпт для генерации примеров атрибутов. Датасет **PoPet**.

You are generating named entities for a synthetic text.

The following strings {fine_grained_entity_types} are all subtypes of the general category: {coarse_grained_entity_type}.

Your goal is to generate **one realistic example** for as many of these types as makes sense in the given context.

The text has the following attributes (domain-specific context and generation constraints): {domain_attributes}

Your task:

- Generate a single realistic value for each entity type.

- Return the output as a JSON dictionary with entity types as keys and corresponding example entities as values.

- Do not leave the dictionary empty, and do not skip any entity types.

- Values must be plausible within the domain and attributes.

Example output format:

json:

{{

"person": "Dr. Emily Carter

"city": "Boston

"organization": "Massachusetts General Hospital

"disease": "hypertension

"drug": "Lisinopril"

}}

Рис. 17: Промпт для генерации примеров сущностей, согласованных с заданным списком примеров текстовых атрибутов. Датасет **PoPet**.

You are generating a synthetic text in the domain of: {domain_name}.

The following attributes describe the text to be generated: {domain_attributes}

Here is a list of possible named entities relevant to this domain: {generated_entities}

Your task is to write a realistic and coherent text based on the domain and attributes. Naturally incorporate a reasonable subset of the provided entities into the text. Do not include all of them—only use what makes sense. The entities should appear in a natural way without being explicitly labeled or listed.

Text:

Рис. 18: Промпт для генерации текстов по заданному списку сущностей и текстовых атрибутов. Датасет **PoPet**.

User: Text: {text}

Assistant: I've read this text.

User: What describes {entity_type} in the text?

Рис. 19: Zero-shot промпт для модели Uniner-7B-type для извлечения фиксированного типа именованных сущностей. Датасет **PoPet**.

You are given:

1. A dictionary of named entity types. Each type maps to a list of surface forms (aliases or variations) of a **single logical entity**.

2. A generated text where some of these entities may appear under one or more of their surface forms.

Your task:

- Identify all the surface forms from the dictionary that are actually present in the text.

- For each entity type, return a list of all surface forms **that appear in the text**, in the order they appear.

- Use **exact string matching** (case-sensitive).

- If no surface forms from a type are present, omit that type from the output.

—

Example:

* Generated text: John Smith, the founder of OpenAI, visited SF last week. Mr. Smith gave a talk about artificial general intelligence.

* Entity dictionary:

{ "person": ["John Smith"], "organization": ["OpenAI"], "location": ["SF"] }

* Expected output:

{ "person": ["John Smith "Mr. Smith"], "organization": ["OpenAI"], "location": ["SF"] }

Now process the following input:

* Generated text: {generated_text}

* Entity dictionary:

{}

Рис. 20: Промпт для извлечения заданного списка сущностей из текста. Датасет **PoPet**.

You are a named entity recognizer that must detect the next entities:

**person**: A person name, it can include first and last names, for example: John Kennedy and Bill Gates
**organization**: An organization name, it can be a company, a government agency, etc.
**location**: A location name, it can be a city, a country, etc.
**miscellaneous**: Words derived from location/organization/person, religions, political ideologies, nationalities, languages, programs, events, wars, sports related names, titles, slogans, eras in time

You must answer with the same input text, but with the named entities annotated with in-line tag annotations (<entity>text</entity>), where each tag corresponds to an entity name, for example: <name>John Doe</name> is the owner of <organization>ACME</organization>.

The only available tags are: ['person', 'organization', 'location', 'miscellaneous'], you cannot add more tags than the included in that list.

**IMPORTANT**: YOU SHOULD NOT CHANGE THE INPUT TEXT, ONLY ADD THE TAGS.

**Example 1:**
Input: Gabriel Boric is the president of Chile
Output: <person>Gabriel Boric</person> is the president of <location>Chile</location>

<...>

**Your task:**
Input: {text}

Рис. 21: Few-shot промпт для модели Qwen2.5-32B для извлечения фиксированных типов именованных сущностей. Датасет **PoPet**.