МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ М. В. ЛОМОНОСОВА
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ
КАФЕДРА МАТЕМАТИЧЕСКИХ МЕТОДОВ ПРОГНОЗИРОВАНИЯ

Цыбанов Илья Антонович

Обучение и оценивание качества моделей обобщенного контент-анализа по размеченным текстовым корпусам

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:
профессор РАН, д.ф.-м.н.
Воронцов Константин Вячеславович

Москва, 2026

Оглавление

Введение . . . 3
1 Постановка задачи . . . 5
1.1 Рубрикатор . . . 5
1.2 Универсальный формат данных . . . 5
1.3 Разметка с перекрытием . . . 6
1.4 Критерии качества . . . 6
1.5 Постановка задачи в общем виде . . . 8
1.6 Бенчмарк . . . 9
2 Обзор существующих решений . . . 14
2.1 Контент-анализ и проблема его автоматизации . . . 14
2.2 Согласование альтернативных экспертных разметок . . . 14
2.3 Универсальные схемы и наборы данных для разметки текста . . . 15
2.4 Подходы к автоматизации контент-анализа . . . 15
2.5 Критерии качества в задачах с неоднозначной разметкой . . . 17
2.6 Ограничения существующих подходов . . . 17
3 Исследование и построение решения задачи . . . 19
3.1 Алгоритм оптимального сопоставления разметок . . . 19
3.2 Подходы к реализации моделей . . . 22
3.3 Подход №1: модель на основе архитектуры Mamba . . . 22
3.4 Подход №2: модель на основе архитектуры Transformer . . . 25
3.5 Подход №3: большие языковые модели . . . 25
4 Описание практической части . . . 28
4.1 Методология проведения экспериментов . . . 28
4.2 Результаты экспериментов . . . 28
4.3 Анализ результатов . . . 31
4.4 Вычислительная сложность . . . 32
5 Заключение . . . 33
5.1 Результаты, выносимые на защиту . . . 33
5.2 Основные выводы . . . 33
5.3 Степень решения поставленной задачи . . . 34

Список литературы **36**

Введение

Контент-анализ представляет собой метод выделения качественных и количественных характеристик текста по заданному рубрикатору и их последующего анализа. Данный метод широко применяется в лингвистике средств массовой информации, социологии, истории, политологии и других социо-гуманитарных науках. Традиционная процедура контент-анализа включает четыре основных этапа: разработку классификатора и методики разметки текстов, составление инструкций и рубрикаторов для разметчиков, проведение экспертной разметки, а также подсчёт и анализ количественных характеристик текстов.

В последние годы задача автоматизации контент-анализа приобретает всё большую актуальность. Рост объёмов текстовых данных в цифровых источниках делает ручную обработку экспертами практически невозможной, при этом труд разметчиков-экспертов крайне дорог. Современные языковые модели достигли значительного прогресса, что делает задачу автоматизации контент-анализа практически реализуемой.

Особенностью задач контент-анализа является неоднозначность разметок, получаемых от разных экспертов. Для повышения надёжности результатов применяется разметка с перекрытием: один и тот же текст независимо размечается несколькими экспертами, после чего их оценки агрегируются. Это порождает специфические проблемы при оценивании качества моделей машинного обучения, которые должны учитывать отсутствие единой эталонной разметки, а точнее наличие нескольких корректных разметок одного документа.

Основные понятия

В рамках настоящей работы используются следующие ключевые понятия:

1) **Фрагмент** — непрерывная часть текста, задаваемая начальной и конечной позицией.

2) **Элемент** — набор взаимосвязанных фрагментов, объединённых некоторым отношением.

3) **Затекст** — комментарий, объяснение или дополнительная информация, относящиеся к фрагменту или элементу.

4) **Тег** — метка класса, присваиваемая фрагменту, элементу или затексту. Каждый из этих объектов имеет (возможно, пустое) множество тегов.

Для иллюстрации использования этих понятий приведём конкретные примеры задач контент-анализа:

- Выделение манипуляций в материалах средств массовой информации требует обнаружения фрагментов, содержащих приёмы скрытого воздействия на читателя. Кроме этого, для каждого фрагмента выделяются типы манипуляций, примененных в нем.

- Выделение культурных ценностей в текстах предполагает идентификацию фрагментов, отражающих определённые ценностные установки, например витальные, моральные

или политические ценности, т.е. выдялется и типы ценностей, выраженных в фрагменте. Кроме этого, выделяются группы фрагментов, выражающие схожую мысль или ценность, - элементы.

- Выделение ошибок в сочинениях ЕГЭ включает в себя выделение фрагментов текста, содержащих ошибки, выделение типов ошибок, а также описание затекстов, подробно комментирующих причину, по которой эксперт увидел в том или ином фрагменте ошибку.

Обобщённый контент-анализ

Задачи контент-анализа характеризуются значительным разнообразием. Они различаются по сигнатуре — набору операций, которые необходимо выполнить над текстом. К таким операциям относятся: выделение фрагментов текста, классификация фрагментов по тегам, выделение отношений (связей) между фрагментами, классификация самих отношений, а также составление затекстов. Произвольная задача контент-анализа представляет собой комбинацию этих операций в различных сочетаниях.

Традиционный подход предполагает разработку отдельной модели машинного обучения для каждой конкретной задачи. При переходе к новой задаче требуется заново проектировать архитектуру модели, определять формат входных и выходных данных, а зачастую и вовсе пересматривать подход к решению.

В настоящей работе предлагается принципиально иной подход, который можно назвать **обобщённым контент-анализом**. Его суть состоит в универсализации на двух уровнях.

Универсальность по сигнатуре задачи. Вместо разработки отдельной модели для каждой комбинации операций предлагается использовать единый универсальный формат представления данных, охватывающий все возможные типы задач контент-анализа. Этот формат описывает произвольную задачу как последовательность трёх вложенных уровней: выделение фрагментов текста и их тегов; выделение элементов (связанных групп фрагментов) и их тегов; составление затекстов. Любая практическая задача контент-анализа сводится к работе с одной и той же трёхуровневой структурой, что позволяет использовать единую модель машинного обучения без изменения архитектуры.

Универсальность по домену задачи. Единый формат данных и модель не привязаны к конкретной предметной области. Одна и та же модель способна работать с задачами контент-анализа из произвольного домена социо-гуманитарных наук — будь то анализ материалов СМИ, экзаменационных работ, исторических документов или юридических текстов — при условии наличия соответствующего рубрикатора тегов. Более того, тот же подход применим и за пределами социо-гуманитарной области, в любой предметной области, где требуется структурированная разметка текста.

Именно эта двойная универсальность — по сигнатуре и по домену — отличает предлагаемый обобщённый контент-анализ от существующих подходов и составляет новизну настоящей работы.

1. Постановка задачи

1.1. Рубрикатор

Каждая задача контент-анализа определяется рубрикатором — набором тегов, используемых для разметки. Рубрикатор формально задаётся тройкой множеств

$$\mathcal{T} = (T^{\text{frag}}, T^{\text{elem}}, T^{\text{ctx}}),$$

где T^{frag} — множество тегов фрагментов, T^{elem} — множество тегов элементов, T^{ctx} — множество тегов затекстов. Эти множества могут пересекаться, в частности возможно совпадение $T^{\text{frag}} = T^{\text{elem}} = T^{\text{ctx}}$.

Каждый тег $t \in T^{\text{frag}} \cup T^{\text{elem}} \cup T^{\text{ctx}}$ снабжается текстовым описанием, раскрывающим его семантику для экспертов-разметчиков. Описания тегов являются частью инструкции по разметке и обеспечивают качество разметки.

1.2. Универсальный формат данных

Используемый универсальный формат данных для задач контент-анализа описан в работе [1] и включает три вложенных уровня.

Уровень 1 (Уровень фрагментов). Фрагмент f представляет собой непрерывную последовательность символов текста, определяемую начальной и конечной позицией. Формально:

$$f = (b, e, F_t), \quad b, e \in \mathbb{N},\ b \leqslant e, \quad F_t \subseteq T^{\text{frag}},$$

где b — начальная позиция, e — конечная позиция, F_t — множество тегов фрагмента (может быть пустым).

Уровень 2 (Уровень элементов). Элемент e представляет собой набор взаимосвязанных фрагментов, объединённых некоторым отношением. Формально:

$$e = (F_e, E_t), \quad F_e \subseteq \{f\}, \quad E_t \subseteq T^{\text{elem}},$$

где F_e — множество фрагментов, входящих в элемент, E_t — множество тегов элемента.

Уровень 3 (Уровень затекстов). Затекст c представляет собой комментарий, пояснение или дополнительную информацию, присоединяемую к фрагменту или элементу. Формально:

$$c = (r, \text{text}, C_t), \quad r \in \{f\} \cup \{e\}, \quad C_t \subseteq T^{\text{ctx}},$$

где r — ссылка на комментируемый объект (фрагмент или элемент), text — текст затекста, C_t — множество тегов затекста.

Данная трёхуровневая структура является достаточной для представления произвольной задачи контент-анализа, встречающейся на практике. В зависимости от задачи может применяться разметка разного уровня. Также важно отметить, что каждый более высокий уровень включает в себя все сущности разметки из более низких уровней (так, например, разметка уровня 2 включает в себя и элементы, и фрагменты).

1.3. Разметка с перекрытием

При ручном решении задачи контент-анализа эксперты-разметчики работают независимо, вследствие чего их разметки одного и того же текста, как правило, не совпадают. Это проявляется в следующем:

- один и тот же фрагмент получает разные теги у разных экспертов;
- границы фрагментов у разных экспертов не совпадают;
- один эксперт выделяет фрагмент, а другой — нет;
- фрагменты, выделенные разными экспертами, частично перекрываются.

Важно пояснить, что несогласованность экспертных разметок не является проблемой или ошибкой. Она отражает объективную сложность задачи: многие текстовые фрагменты допускают неоднозначную интерпретацию, и различные эксперты фиксируют различные аспекты смысла. Данная вариативность не может быть устранена увеличением числа экспертов или более детальными инструкциями — она имманентна самой природе задачи и выражается в «шумности» разметки.

В связи с этим необходимо правильно выстроить методику обучения моделей машинного обучения так, чтобы несогласованность в разметках выступала не недостатком, а преимуществом. А именно: модель должна обучаться не на одной усреднённой разметке, а на всём множестве альтернативных экспертных оценок, учитывая распределение тегов и границ фрагментов. Это позволяет модели оценивать неопределённость своих предсказаний и лучше обобщаться на новые данные за счёт учёта полного спектра возможных интерпретаций.

1.4. Критерии качества

Для оценивания качества модели контент-анализа необходим механизм сравнения двух разметок: предсказанной моделью $\hat{r}$ и экспертной r. Однако альтернативные экспертные разметки одного текста, как правило, содержат фрагменты с несовпадающими границами и различными наборами тегов, что делает прямое покомпонентное сравнение невозможным. Более того, оно бессмысленно, так как, как уже было сказано, единой эталонной разметки попросту не существует.

Вводятся критерии качества C_1–C_5, позволяющие численно оценить степень согласованности двух разметок. Все величины лежат в отрезке $[0, 1]$, чем выше — тем лучше. Пусть A и B — две разметки одного текста, содержащие фрагменты $F(A)$, $F(B)$ и элементы $E(A)$, $E(B)$. Пусть M — результат сопоставления фрагментов, M_E — результат сопоставления элементов. Тогда:

C_1 — доля фрагментов из A, для которых в B найдено сопоставление:

$$C_1 = \frac{|M|}{|F(A)|};$$

C_2 — точность совпадения текстовых границ сопоставленных фрагментов:

$$C_2 = \frac{1}{|M|} \sum_{(f_A, f_B) \in M} \text{Jaccard}(f_A, f_B), \quad \text{Jaccard}(f_A, f_B) = \frac{|[b_A, e_A] \cap [b_B, e_B]|}{|[b_A, e_A] \cup [b_B, e_B]|};$$

C_3 — точность совпадения множеств тегов сопоставленных фрагментов:

$$C_3 = \frac{1}{|M|} \sum_{(f_A, f_B) \in M} \frac{|F_t(f_A) \cap F_t(f_B)|}{|F_t(f_A) \cup F_t(f_B)|};$$

C_4 — доля элементов из A, для которых в B найдено сопоставление:

$$C_4 = \frac{|M_E|}{|E(A)|};$$

C_5 — точность совпадения множеств тегов сопоставленных элементов:

$$C_5 = \frac{1}{|M_E|} \sum_{(e_A, e_B) \in M_E} \frac{|E_t(e_A) \cap E_t(e_B)|}{|E_t(e_A) \cup E_t(e_B)|}.$$

Совокупный критерий:

$$C = \alpha_1 C_1 + \alpha_2 C_2 + \alpha_3 C_3 + \alpha_4 C_4 + \alpha_5 C_5, \quad \alpha_i \geqslant 0, \ \sum_i \alpha_i = 1.$$

Выбор коэффициентов α_i определяется сигнатурой задачи: для задач выделения фрагментов наиболее значимы C_1 и C_2, для задач классификации — C_3, для задач выделения и классификации отношений — C_4 и C_5 соответственно.

Для разметки с перекрытием, когда каждый документ имеет несколько альтернативных экспертных разметок $R(d) = \{r_1, \ldots, r_k\}$, качество на данном документе определяется как среднее значение совокупного критерия по всем экспертным разметкам:

$$C(d) = \frac{1}{k} \sum_{i=1}^{k} C(\hat{r}, r_i).$$

Для вычисления критериев C_1–C_5 необходимо предварительно сопоставить фрагменты и элементы двух альтернативных разметок. Без сопоставления фрагментов и элементов вычисление критериев невозможно, так как неизвестно, между какими парами фрагментов и элементов вычислять критерии. Для решения этой задачи был разработан алгоритм оптимального сопоставления разметок, подробное описание которого приводится в разделе 3.1. Пример сопоставления двух разметок представлен на Рис. 1.

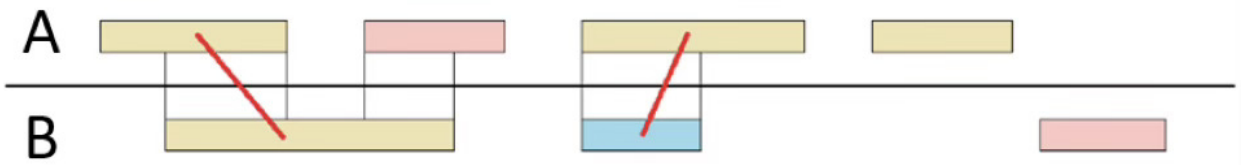

Рис. 1. Пример сопоставления двух альтернативных разметок одного текста. Верхняя строка — фрагменты разметки A, нижняя — фрагменты разметки B, рёбра обозначают установленное соответствие. Цвет обозначает набор тегов фрагмента.

1.5. Постановка задачи в общем виде

Формальная постановка задачи обобщённого контент-анализа формулируется следующим образом.

Дано:

- рубрикатор $\mathcal{T} = (T^{\text{frag}}, T^{\text{elem}}, T^{\text{ctx}})$;
- набор данных $D = \{(d_i, R_i)\}_{i=1}^{n}$, где d_i — документ, $R_i = \{r_{i,1}, \ldots, r_{i,k_i}\}$ — множество альтернативных экспертных разметок этого документа, каждая из которых представлена в универсальном трёхуровневом формате на уровне $L \in \{1, 2, 3\}$;
- разбиение набора данных $D = D_{\text{train}} \cup D_{\text{test}}$.

Требуется:

1) На этапе обучения по обучающей выборке D_{train} подобрать параметры модели θ:

$$\theta^* = \arg\max_{\theta} \; \mathbb{E}_{d \in D_{\text{train}}} \Big[\sum_{i=1}^{k(d)} C(r_i(d), \hat{r}_\theta(d)) \Big],$$

где C — совокупный критерий качества, $k(d)$ — количество экспертных разметок документа d, $r_i(d)$ — i-я экспертная разметка документа d, $\hat{r}_\theta(d)$ — предсказанная моделью разметка документа d.

2) На этапе инференса построить модель M_{θ^*}, которая для произвольного документа d выдаёт предсказанную разметку $\hat{r} = M_{\theta^*}(d)$ на уровне L, соответствующем данному набору данных.

Ограничения:

1) Модель M_θ является единой для любых наборов данных, представленных в универсальном формате разметки любого уровня.
2) Для наборов данных с разметкой с перекрытием модель обучается с учётом всех альтернативных экспертных разметок, используя вариативность как источник дополнительной информации.

3) Модель генерирует разметку на том уровне универсального формата, который представлен в наборе данных.

4) Модель не требует модификации архитектуры при переходе к новой задаче — достаточно предоставить рубрикатор $\mathcal{T}$ и обучающую выборку.

1.6. Бенчмарк

Для экспериментального сравнения подходов использовался бенчмарк, содержащий наборы данных в универсальном формате разметки, предложенный в работе [1]. Бенчмарк содержит задачи разметки текстов и обеспечивает широкое покрытие различных доменов и сигнатур задач. Бенчмарк включает 21 набор данных и 17 типов задач. Каждый набор данных характеризуется списком задач, которые можно решать, используя этот набор данных.

При этом важно отметить, что уровень универсального формата, в котором представляются разметки, зависит не от самого набора данных, а от сигнатуры задачи. На одном и том же наборе данных можно решать совершенно разные задачи, при этом использование слишком высокого уровня модели данных повлечет за собой бессмысленное использование усложненной модели данных, что может повлечь за собой ухудшение итогового результата. Для каждого набора данных зафиксирован максимальный уровень модели данных, используемый в задачах набора, он характеризует необходимый уровень модели данных, который должен использоваться для хранения разметок набора данных для того, чтобы их можно было использовать для решения любой задачи набора.

В таблице 1 приведено описание наборов данных бенчмарка. Первый столбец содержит список наборов данных бенчмарка. Второй столбец содержит названия задач каждого набора данных. Третий столбец содержит описание сигнатур задач и уровень модели данных, соответствующий данной сигнатуре (он обозначается как L). Четвертый столбец содержит домен набора данных.

Набор данных	**Задача**	**Сигнатура**	**Домен**
NEREL [2]	Распознавание именованных сущностей	Выделение фрагментов ($L = 1$)	Новостные статьи
	Извлечение отношений	Выделение элементов ($L = 2$)	
	Извлечение мультиспанов с именованными сущностями	Выделение элементов ($L = 2$)	

Набор данных	**Задача**	**Сигнатура**	**Домен**
RuSentNE [3]	Распознавание именованных сущностей	Выделение фрагментов ($L = 1$)	Новостные статьи
	Семантический анализ	Выделение элементов ($L = 2$)	
	Извлечение кортежей мнений	Выделение элементов ($L = 2$)	
UpGreat READ//ABLE [4]	Выделение и классификация фрагментов с ошибками	Выделение фрагментов ($L = 1$)	Экзаменационные работы
	Выделение мультиспанов с ошибками	Выделение элементов ($L = 2$)	
	Аннотирование фрагментов текста комментариями	Уровень затекстов ($L = 3$)	
CoNLL 2012 Ontonotes [5]	Распознавание именованных сущностей	Выделение фрагментов ($L = 1$)	Новостные статьи
	Выделение семантической роли	Выделение элементов ($L = 2$)	
	Разрешение кореференций	Выделение элементов ($L = 2$)	
SWDA [6]	Распознавание паттернов в диалоге	Выделение элементов ($L = 2$)	Телефонные диалоги
	Классификация текста	Выделение фрагментов ($L = 1$)	
Kaggle NER [7]	Распознавание именованных сущностей	Выделение фрагментов ($L = 1$)	Социальные сети
	POS-разметка	Выделение фрагментов ($L = 1$)	
MultiCoNER [8]	Распознавание именованных сущностей	Выделение фрагментов ($L = 1$)	Статьи с Wikipedia [9]

Набор данных	**Задача**	**Сигнатура**	**Домен**
RuTermEval Dialogue[10]	Распознавание именованных сущностей	Выделение фрагментов ($L = 1$)	Научные тексты
ADE [11]	Классификация текста	Выделение фрагментов ($L = 1$)	Медицинские тексты
	Распознавание именованных сущностей	Выделение фрагментов ($L = 1$)	
	Извлечение отношений	Выделение элементов ($L = 2$)	
	Разрешение кореференций	Выделение элементов ($L = 2$)	
DDI corpus [12]	Распознавание именованных сущностей	Выделение фрагментов ($L = 1$)	Медицинские тексты
	Извлечение отношений	Выделение элементов ($L = 2$)	
PcMSP [13]	Классификация текста	Выделение фрагментов ($L = 1$)	Научные тексты
	Распознавание именованных сущностей	Выделение фрагментов ($L = 1$)	
	Извлечение отношений	Выделение элементов ($L = 2$)	
ChemProt [14]	Классификация текста	Выделение фрагментов ($L = 1$)	Научные тексты
	Распознавание именованных сущностей	Выделение фрагментов ($L = 1$)	
	Извлечение отношений	Выделение элементов ($L = 2$)	
NERRE [15]	Распознавание именованных сущностей	Выделение фрагментов ($L = 1$)	Научные тексты
	Извлечение отношений	Выделение элементов ($L = 2$)	

Набор данных	**Задача**	**Сигнатура**	**Домен**
RURED [16]	Распознавание именованных сущностей	Выделение фрагментов ($L = 1$)	Новостные статьи
	Извлечение отношений	Выделение элементов ($L = 2$)	
	Связывание сущностей	Выделение элементов ($L = 2$)	
SciERC [17]	Распознавание именованных сущностей	Выделение фрагментов ($L = 1$)	Научные тексты
	Извлечение отношений	Выделение элементов ($L = 2$)	
	Разрешение кореференций	Выделение элементов ($L = 2$)	
RuSuperGLUE RWSD [18]	Классификация отношений	Выделение элементов ($L = 2$)	Нет определенного домена
MERA RWSD [19]	Классификация отношений	Выделение элементов ($L = 2$)	Нет определенного домена
MERA Ruethics [19]	Классификация отношений	Выделение элементов ($L = 2$)	Культурные ценности
SemEval-2010 Task 8 [20]	Классификация отношений	Выделение элементов ($L = 2$)	Нет определенного домена
SemEval-2018 Task 7 [21]	Распознавание именованных сущностей	Выделение фрагментов ($L = 1$)	Научные тексты
	Извлечение отношений	Выделение элементов ($L = 2$)	
	Классификация отношений	Выделение элементов ($L = 2$)	
	Построение графа знаний	Выделение элементов ($L = 2$)	

Набор данных	**Задача**	**Сигнатура**	**Домен**
Human Values Dataset[22]	Классификация текста	Выделение фрагментов ($L = 2$)	Культурные ценности
	Извлечение и классификация фрагментов	Выделение фрагментов ($L = 2$)	
	Извлечение и классификация элементов	Выделение элементов ($L = 2$)	
	Семантический анализ элементов	Выделение элементов ($L = 2$)	
	Аннотирование разметки комментариями	Выделение затекстов ($L = 3$)	

Таблица 1: Наборы данных бенчмарка

2. Обзор существующих решений

2.1. Контент-анализ и проблема его автоматизации

Контент-анализ представляет собой систематическую процедуру выделения качественных и количественных характеристик текста по заданному рубрикатору. Метод широко применяется в социологии, политологии, лингвистике средств массовой информации и других социо-гуманитарных науках. Традиционная процедура включает разработку классификатора и методики разметки, составление инструкций для экспертов, проведение экспертной разметки и подсчёт количественных характеристик.

Масштабный переход к цифровым текстовым данным сделал ручную обработку экспертами практически невозможной: объёмы текстов в сети Интернет на несколько порядков превышают то, что могут обработать специалисты [1]. Это порождает потребность в автоматизации контент-анализа, которая, однако, сталкивается с рядом специфических трудностей.

Первая трудность — **неоднозначность экспертных разметок**. При ручной разметке один и тот же текст, как правило, получает различные интерпретации у разных экспертов: границы выделяемых фрагментов не совпадают, один эксперт приписывает фрагменту определённый тег, а другой — нет, допускаются частичные перекрытия [23]. Эта вариативность не является ошибкой — она отражает объективную многозначность текста и не устраняется ни увеличением числа экспертов, ни более детальными инструкциями.

Вторая трудность — **отсутствие единого формата данных**. На практике встречаются задачи контент-анализа с различной сигнатурой: выделение фрагментов текста, классификация фрагментов по тегам, установление отношений между фрагментами, описание затекстами и комментариями. Существующие наборы данных используют собственные форматы представления разметки, что препятствует построению универсальных моделей и проведению сравнительных экспериментов.

Третья трудность — **ограниченность обучающих данных**. Разметка экспертами стоит дорого, поэтому для большинства задач доступны лишь малые обучающие выборки, что критично для моделей, требующих большого числа примеров.

2.2. Согласование альтернативных экспертных разметок

Классические меры согласованности экспертов — каппа Коэна, каппа Фляйсса — предполагают, что каждый объект аннотируется всеми экспертами, и оценивают согласованность на уровне меток. Однако они неприменимы к задачам, где эксперты независимо выделяют произвольные фрагменты текста с произвольными границами, поскольку в таких условиях невозможно установить, какой фрагмент одного эксперта соответствует какому фрагменту другого. При этом существуют подходы, учитывающие несогласованность разметок экспретов, такие как [24].

2.3. Универсальные схемы и наборы данных для разметки текста

Попытки унификации форматов разметки предпринимались в различных предметных областях. В научном дискурсе схема SciIE [17] предусматривает совместное распознавание сущностей, извлечение отношений и разрешение кореференций. Работа [25] предлагает фреймворк OneIE для совместного решения всех этих задач в рамках единой модели.

На сессии ValueEval [26] в рамках конференции SemEval был создан систематизированный бенчмарк для задачи обнаружения человеческих ценностей в текстах. Теоретической основой служит теория ценностей Шварца [27], описывающая универсальную структуру из десяти типов ценностей. В ValueEval-2024 оценивались две подзадачи: определение ценностей в предложении и определение того, достигается ли ценность или ограничивается в контексте. Для обучения использовалась коллекция ValuesML, содержащая около 3000 аннотированных текстов на восьми языках.

Однако область применения существующих бенчмарков ограничена: ValueEval охватывает только две подзадачи, а большинство универсальных схем привязаны к конкретной предметной области. Задачи контент-анализа в реальных приложениях значительно разнообразнее и могут включать выделение произвольных фрагментов, иерархические классификации, установление произвольных отношений и составление затекстов.

2.4. Подходы к автоматизации контент-анализа

Современные подходы к автоматизации контент-анализа можно разделить на три класса.

Словарные методы и ранние статистические подходы.

Ранние работы в области автоматизации контент-анализа опирались на словарные методы: каждой категории ставился в соответствие набор лексем, и для подсчёта упоминаний достаточно было найти соответствующие слова. Например, классические системы вроде General Inquirer [28] опирались на вручную составленные словари психологических и моральных категорий, тем самым реализовав «машиночитаемый» контент-анализ. Аналогично, инструмент LIWC (Linguistic Inquiry and Word Count) [29] подсчитывает слова по заранее определённым психологическим категориям: его применение показало способность выявлять смысловые сигналы (эмоции, стили мышления, социальные отношения и др.) в самых разных экспериментальных задачах. Было показано, что такие методы демонстрируют приемлемое качество в пределах одного домена, однако их способность к обобщению при переносе на новые домены остаётся ограниченной.

В работе [30] использовались словарные подходы анализа текстов для изучения дискурса о ценностях в материалах Европейского парламента.

Главное ограничение словарных методов состоит в том, что категории многих задач контент-анализа невозможно свести к ограниченному набору лексем, что делает эти подходы малоприменимыми на практике.

Подходы на основе моделей BERT и рекуррентных архитектур.

Современные подходы к автоматическому контент-анализу активно используют модели на основе архитектуры Transformer [31], такие как модели семейства BERT [32].

Общая схема подхода для задач разметки текста включает выделение фрагментов (NER - Named Entity Recognition) с использованием схемы BIO или её модификаций с последующей классификацией выделенных фрагментов. Для этого используются дополнительные слои поверх эмбеддингов токенов.

В [1] была детально исследована модель на основе мультиязычной RoBERTa [33] для русскоязычных текстов социальных сетей. Показано, что модель эффективно обнаруживает родительские классы ценностей (F1 = 0.753), однако значительно хуже работает на дочерних уровнях иерархического классификатора (F1 = 0.274). Обнаружена практически линейная зависимость между значением F1 и количеством аннотированных примеров на класс, что указывает на критическую зависимость качества от объёма обучающей выборки.

Рекуррентные архитектуры на основе селективных пространств состояний (Mamba) [34] представляют альтернативу Transformers с линейной сложностью относительно длины последовательности, что делает их привлекательными для задач обработки длинных текстов. Однако предобученных моделей Mamba для русского языка в открытом доступе существенно меньше, чем моделей семейства BERT.

Подходы на основе больших языковых моделей.

Ряд работ демонстрирует эффективность использования больших языковых моделей (LLM) для анализа текстов методом промптинга. В [35] промптинг применялся для построения графа связей между предложениями в документе. Работа [36] использует ансамбли моделей с различными шаблонами промптов и несколько примеров для few-shot обучения.

В [1] в качестве базовой модели использовался Qwen 2.5 14B Instruct. Результаты показали, что промптовый подход значительно превосходит BERT-модель на дочерних уровнях классификатора (F1 = 0.537 против F1 = 0.274). Это объясняется тем, что LLM способна обобщать на основе few-shot примеров даже при малом количестве данных, что критически важно для задач с редкими классами.

Ряд недавних работ исследует использование LLM непосредственно в качестве разметчика. В [37] показано, что сильнейшие современные модели достигают межэкспертной согласованности, сопоставимой со средней согласованностью между людьми-разметчиками. В работе [38] авторы приходят к выводу о том, что LLM показывают высокую согласованность с людьми-программистами на комплексных задачах программирования.

Ансамблевые стратегии.

В ряде работ для преодоления ограничений отдельных моделей предлагались ансамблевые стратегии. В [1] была предложена стратегия Mixture of Experts (MoE), которая выбирает предсказание от модели, показавшей лучшее качество на этапе обучения для данного класса. Результаты демонстрируют, что MoE-модель превосходит отдельные модели во всех экспериментах, что указывает на перспективность ансамблевых подходов.

2.5. Критерии качества в задачах с неоднозначной разметкой

Стандартные критерии Accuracy, Precision, Recall и F1-мера предполагают наличие единственной эталонной разметки. Для задач с множественной экспертной разметкой такой подход неприменим, поскольку эталонной разметки не существует — есть лишь набор альтернативных экспертных оценок.

Один из подходов состоит в агрегировании множественных разметок в одну путём мажоритарного голосования или выбора наиболее частотной разметки. Однако такой подход игнорирует информацию о несогласованности экспертов и может искажать оценку качества, особенно когда эксперты систематически расходятся во мнениях.

Другой подход, применяемый в настоящей работе, состоит в оценивании модели на каждой альтернативной экспертной разметке с последующим усреднением результатов. При этом для сравнения двух разметок (предсказанной и экспертной) необходимо предварительно сопоставить их фрагменты и элементы, что требует алгоритма оптимального сопоставления, описанного в разделе 3.1. Подходы к реализации универсальных для различных задач обработки естественного языка критериев качества существуют: так, в работе [39] предлагается подход оценивания более узкого класса задач при помощи универсальных критериев.

2.6. Ограничения существующих подходов

Проведённый обзор позволяет выделить следующие ограничения существующих решений.

1) **Отсутствие универсальности.** Существующие подходы, как правило, разрабатываются для конкретной задачи или предметной области. BERT-подобные модели дообучаются под конкретный рубрикатор, а промптовые стратегии требуют индивидуальной настройки шаблонов. Перенос на новую задачу требует существенной переработки [1].

2) **Отсутствие единого формата данных.** Различные наборы данных для задач контент-анализа используют собственные форматы представления разметки, что затрудняет построение единой модели и проведение сравнительных экспериментов.

3) **Несовместимость с разметкой с перекрытием.** Существующие критерии качества (Accuracy, Precision, Recall, F1) не учитывают специфику разметки с перекрытием, при которой одни и те же фрагменты могут иметь различные границы и теги у разных экспертов.

4) **Критическая зависимость от объёма данных.** BERT-модели демонстрируют значительную деградацию качества при малом числе обучающих примеров, что ограничивает их применимость в задачах контент-анализа, где экспертная разметка дорога и малочисленна.

Настоящая работа направлена на преодоление указанных ограничений путём разработки критериев качества, адаптированных для разметки с перекрытием, и подходов к постро-

ению моделей машинного обучения, применимых к произвольным задачам без изменения архитектуры.

3. Исследование и построение решения задачи

3.1. Алгоритм оптимального сопоставления разметок

Как было показано в разделе 1.4, для вычисления критериев качества C_1–C_5 необходимо сопоставить фрагменты и элементы двух альтернативных разметок A и B. Прямое покомпонентное сравнение невозможно из-за того, что неизвестно, для каких пар фрагментов и элементов 2 разметок нужно вычислять критерии качества. Ниже описан алгоритм оптимального сопоставления, предложенный в настоящей работе.

Алгоритм решает следующую задачу: по двум альтернативным разметкам A и B найти оптимальное множество пар фрагментов D и множество пар элементов E_D, для которых затем вычисляются критерии качества C_1–C_5. Оптимальность понимается в смысле максимизации совокупного критерия качества, т.е. на самом деле алгоритм максимизирует следующий функционал:

$$\alpha_1 C_1(D) + \alpha_2 C_2(D) + \alpha_3 C_3(D) + \alpha_4 C_4(D, E_D) + \alpha_5 C_5(D, E_D) \to max_{D,E_D}$$

где оптимизация ведется по всем возможным наборам пар фрагментов и элементов разметок A и B. Очевидно, полный перебор слишком затратен вычислительно, именно поэтому вводится этот алгоритм оптимального сопоставления разметок. Ниже описаны шаги самого алгоритма.

1. Матрица расстояний между фрагментами

Для каждой пары фрагментов (f_A, f_B), $f_A \in A$, $f_B \in B$, вычисляется расстояние $d(f_A, f_B)$. Каждый фрагмент представляется тройкой (b, e, T), где b, e — номера начального и конечного символа фрагмента, T — множество тегов фрагмента.

Мера сходства Жаккара для перекрытия спанов:

$$\mathrm{J}(f_A, f_B) = 1 - \frac{|[b_A, e_A] \cap [b_B, e_B]|}{|[b_A, e_A] \cup [b_B, e_B]|}.$$

Легко видеть, что для пары неперекрывающихся фрагментов $\mathrm{J} = 1$. Чем сильнее перекрываются фрагменты, тем меньше штраф за их несовпадение. Фрагменты обязательно имеют натуральную длину, поэтому меру сходства Жаккара можно вычислить для любых 2 фрагментов.

Штраф за несовпадение тегов:

$$\Delta(f_A, f_B) = \begin{cases} 0, & T_A = T_B = \varnothing, \\ 1 - \dfrac{|T_A \cap T_B|}{|T_A \cup T_B|}, & \text{иначе.} \end{cases}$$

Множество тегов фрагмента может быть пустым, поэтому обязательна отдельная обработка ситуации, при которой оба фрагмента имеют пустые множества фрагментов. В остальных случаях механизм вычисления штрафа сильно напоминает меру сходства Жаккара.

Итоговое расстояние:

$$d(f_A, f_B) = \mathrm{J}(f_A, f_B) + \Delta(f_A, f_B).$$

Штраф за различие тегов и мера сходства Жаккара принимают значения из диапазона $[0, 1]$. Таким образом, $d \in [0, 2]$.

Построенная матрица расстояний $D = \|d_{ij}\|$ имеет размер $N_A \times N_B$, где N_A, N_B — числа фрагментов в разметках A и B. Далее будем рассматривать двудольный граф, который задает эта матрица (каждая доля графа — фрагменты соответствующей разметки, а каждая вершина — фрагмент некоторой разметки).

2. Удаление заведомо нежелательных рёбер

Из двудольного графа, образованного фрагментами A и B с рёбрами веса d, удаляются все рёбра с весом $\geqslant 1$. Это значение выбрано эмпирически в ходе ручного тестирования алгоритма: при $d \geqslant 1$ совокупный штраф слишком велик, и сопоставление таких пар нецелесообразно.

После удаления рёбер граф может распасться на несколько компонент связности. Каждая компонента обрабатывается независимо, что значительно ускоряет работу алгоритма относительно обработки всего графа целиком. Следует отметить, что этап удаления рёбер напрямую определяет множество сопоставленных фрагментов и тем самым влияет на значение критерия C_1: если фрагмент из A не имеет ни одного ребра к фрагментам из B с весом < 1, он останется несопоставленным и снизит критерий C_1. Аналогично для фрагментов разметки B.

3. Сопоставление внутри компонент связности

Внутри каждой компоненты задача сводится к поиску оптимального паросочетания в двудольном графе. Для этого используется венгерский алгоритм (алгоритм Куна) [40], решающий задачу о назначениях за полиномиальное время $O(n^3)$, где n — размер компоненты.

Венгерский алгоритм находит паросочетание, минимизирующее суммарное расстояние между сопоставленными парами. Сопоставленные таким образом пары составляют множество D. Далее для каждого фрагмента из A, вошедшего в D, запоминается соответствующий ему фрагмент из B. Эта информация используется на этапе сопоставления элементов и при вычислении критериев качества.

4. Сопоставление элементов

После сопоставления фрагментов аналогичная процедура применяется к элементам. Каждый элемент представляется парой (S, T), где S — множество входящих в него фрагментов, T — множество тегов.

Для элементов $e_A = (S_A, T_A)$, $e_B = (S_B, T_B)$ вычисляется расстояние $d_{\text{elem}}(e_A, e_B)$, включающее:

- штраф за различие тегов элементов:

$$\delta_T(e_A, e_B) = \begin{cases} 0, & T_A = T_B = \varnothing, \\ 1 - \dfrac{|T_A \cap T_B|}{|T_A \cup T_B|}, & \text{иначе,} \end{cases}$$

- штраф за неперекрытие элементов:

$$\delta_S(e_A, e_B) = 1 - \frac{|S_A \cap S_B|}{|S_A \cup S_B|},$$

где $|S_A \cap S_B|$ — число фрагментов, сопоставленных на предыдущем этапе.

Итоговое расстояние между элементами:

$$d_{\text{elem}}(e_A, e_B) = \delta_S(e_A, e_B) + \delta_T(e_A, e_B).$$

После построения матрицы расстояний для элементов применяется та же процедура: удаление рёбер с весом $\geqslant 1$, разбиение на компоненты связности, применение венгерского алгоритма внутри каждой компоненты. Результат — множество пар элементов E_D.

Результатом работы всего алгоритма являются множества D (пары фрагментов) и E_D (пары элементов), на основе которых вычисляются критерии C_1–C_5.

Вычисление критериев

По результатам сопоставления (D — пары фрагментов, E_D — пары элементов) критерии вычисляются следующим образом.

C_1 — доля сопоставленных фрагментов:

$$C_1 = \frac{2|D|}{|A| + |B|}.$$

C_2 — среднее по парам расстояние по Жаккару:

$$C_2 = \frac{1}{|D|} \sum_{(f_A, f_B) \in D} \big(1 - \mathrm{J}(f_A, f_B)\big).$$

C_3 — средняя точность совпадения тегов фрагментов:

$$C_3 = \frac{1}{|D|} \sum_{(f_A, f_B) \in D} \frac{|T_A \cap T_B|}{|T_A \cup T_B|},$$

где при $T_A = T_B = \varnothing$ слагаемое равно 1.

C_4 — доля сопоставленных элементов:

$$C_4 = \frac{2|E_D|}{|E_A| + |E_B|}.$$

C_5 — средняя точность совпадения тегов элементов:

$$C_5 = \frac{1}{|E_D|} \sum_{(e_A, e_B) \in E_D} \frac{|T_A \cap T_B|}{|T_A \cup T_B|},$$

где при $T_A = T_B = \varnothing$ слагаемое равно 1.

3.2. Подходы к реализации моделей

Для решения задачи универсализации и автоматизации контент-анализа было выбрано три перспективных направления, основанных на различных современных архитектурах машинного обучения:

1) модель на основе рекуррентной архитектуры с селективными пространствами состояний (Mamba), будем называть этот подход более общим классом моделей RNN (Recurrent Neural Networks — Рекуррентные Нейронные Сети), в который входит указанная архитектура;

2) модель на основе архитектуры Transformer (BERT — Bidirectional Encoder Representations from Transformers);

3) большие языковые модели (БЯМ или LLM — Large Language Models).

3.3. Подход №1: модель на основе архитектуры Mamba

В качестве базовой модели используется mamba-1.4b-ru — рекуррентная нейронная сеть на основе селективных пространств состояний, архитектура который была предложена в работе Gu and Dao (2023) [34]. Выбранная модель является одной из немногих предобученных для работы с русским языком. Довольно сложно найти предобученные на русскоязычных текстах модели такой архитектуры в открытом доступе, что делает данную модель особенно ценной для исследования.

Общая схема

Модель работает в два этапа, соответствующие двум уровням универсального формата данных: на первом этапе решается задача выделения фрагментов текста, на втором — задача выделения элементов (групп взаимосвязанных фрагментов) и их тегов. При этом обучение также проводится в два этапа: на каждом из них решается независимая задача классификации со своей функцией потерь. Базовая модель остаётся замороженной на протяжении всего обучения; обучаются только дополнительные слои, добавляемые к базовой модели. Такое проектное решение обусловлено высокой вычислительной стоимостью обучения базовой модели: её предобученные параметры уже позволяют получить богатые представления текста в латентном пространстве, достаточные для решения поставленной задачи, и дообучение базовой модели не оправдано с точки зрения затрат ресурсов. Общая схема подхода представлена на Рис. 2. Фактически предложен достаточно общий подход, ведь в качестве базовой модели может быть использована любая модель, генерирующая обогащенные латентные предствления (экмбеддинги) поданного на входа текста. Этот факт будет использоваться далее: подход на основе RNN предполагает использование модели архитектуры Mamba [34], а подход на основе BERT - модели на основе архитектуры Transformer [31].

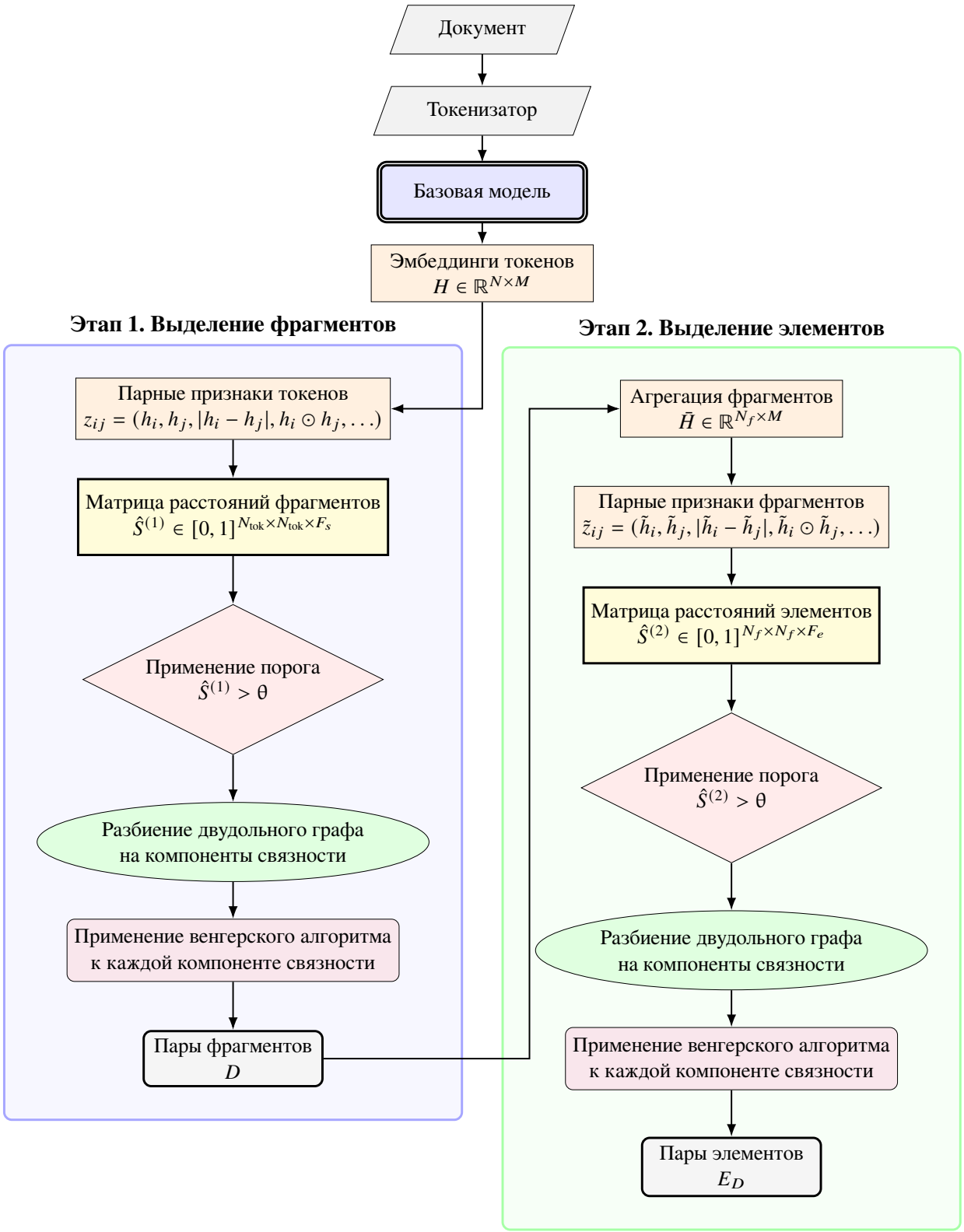

Рис. 2. Схема подходов на основе RNN и BERT (в зависимости от подхода в качестве базовой модели используются соответствующие модели)

Этап 1: выделение фрагментов.

На первом этапе решается задача определения для каждой пары токенов (w_i, w_j), $i < j$, принадлежит ли эта пара одному фрагменту и, если да, то каким типам фрагмента. Базовая модель используется для получения контекстных эмбеддингов всех токенов текста.

Размерность эмбеддинга равна $M = 4096$.

Для каждой пары токенов (i, j) с ограничением $j - i \leqslant 64$ (ограничение на максимальное расстояние вводится для управления вычислительной сложностью, которая иначе составляла бы $O(N^2)$, где N - количество токенов текста) формируется вектор признаков $z_{ij} \in \mathbb{R}^{4M+3}$:

$$z_{ij} = \big(h_i,\ h_j,\ |h_i - h_j|,\ h_i \odot h_j,\ |b_i - b_j|,\ |e_i - e_j|,\ |l_i - l_j|\big),$$

где h_i, h_j — эмбеддинги токенов, $|h_i - h_j|$, $h_i \odot h_j$ — их разность и покомпонентное произведение, $|b_i - b_j|$, $|e_i - e_j|$, $|l_i - l_j|$ — абсолютные разности начала, конца и длины спанов соответственно. Последние три признака нормализуются в значения из $[0, 1]$.

Вектор z_{ij} подаётся на вход классифицирующего перцептрона $g_\varphi^{(1)}$, параметризованного вектором φ:

$$g_\varphi^{(1)}(z_{ij}) = W_2^{(1)}\,\sigma(W_1^{(1)} z_{ij} + b_1^{(1)}) + b_2^{(1)}, \quad W_1^{(1)} \in \mathbb{R}^{M\times(4M+3)},\ W_2^{(1)} \in \mathbb{R}^{F_s\times M},$$

где σ — функция активации GELU, F_s — число возможных тегов фрагментов. Результат $g_\varphi^{(1)}(z_{ij}) \in \mathbb{R}^{F_s}$ интерпретируется как логиты; применение к нему сигмоиды позволяет получить вектор, который можно интерпретировать как вероятности наличия соответствующего тега у фрагмента: $\hat{p}_{ij} = \sigma(g_\varphi^{(1)}(z_{ij})) \in [0, 1]^{F_s}$.

Для каждого типа тегов фрагментов f определяется бинарное отношение на токенах: токены i и j связываются ребром, если $\hat{p}_{ij}[f] > \theta_f$. Порог θ_f подбирается на валидационной выборке: после каждой эпохи перебираются 20 значений, равномерно распределенных по диапазону $[0.05, 0.8]$, и выбирается порог, максимизирующий среднее критериев качества C_1, C_2, C_3. Если несколько одинаковых пар токенов связаны разными отношениями, то такая связь интерпретируется как фрагмент с несколькими тегами.

Функция потерь на этапе 1. Задача предсказания наличия фрагмента с определёнными тегами для пары токенов (i, j) формулируется как задача классификации с F_s бинарными выходами. Для каждой пары (i, j) и каждого тега фрагмента f модель предсказывает вероятность $\hat{p}_{ij}[f]$. Функция потерь — бинарная кросс-энтропия:

$$\mathcal{L}_1(\varphi) = -\frac{1}{N_{\text{pairs}} F_s} \sum_{i<j} \sum_{f=1}^{F_s} \Big[y_{ij}(f)\ \log \hat{p}_{ij}[f] + (1 - y_{ij}(f))\ \log(1 - \hat{p}_{ij}[f]) \Big],$$

где $y_{ij}(f) \in \{0, 1\}$ — индикатор того, что пара (i, j) входит во фрагмент типа f. При обучении применяется отрицательное сэмплирование: для каждой положительной пары в батче (подвыборка, по которой производится шаг обучения) отбирается ограниченное число отрицательных пар с целью поддержания заданного соотношения положительных и отрицательных примеров.

Этап 2: выделение элементов.

На втором этапе решается задача группировки уже выделенных фрагментов в элементы и определения их тегов. Входом служат агрегированные эмбеддинги N фрагментов, полученных на первом этапе, представленные в виде матрицы $H \in \mathbb{R}^{N\times M}$. Агрегация эмбеддингов фрагментов выполняется при помощи усреднения.

Для каждой пары фрагментов (f_i, f_j) строится трёхканальная матрица признаков $[P_{ij}]$, где $P_{ij} \in \mathbb{R}^3$:

$$P_{ij} = \big(|b_i - b_j|,\ |e_i - e_j|,\ |l_i - l_j|\big),$$

где $|b_i - b_j|$, $|e_i - e_j|$ — расстояния между началами и концами спанов, $|l_i - l_j|$ — разность длин. Все компоненты нормализуются в диапазон $[0, 1]$.

Для каждой пары (i, j) формируется расширенный вектор признаков:

$$\tilde{z}_{ij} = \big(h_i,\ h_j,\ |h_i - h_j|,\ h_i \odot h_j,\ P_{ij}\big) \in \mathbb{R}^{4M+3},$$

где h_i, h_j — агрегированные эмбеддинги фрагментов i и j, P_{ij} — вектор признаков расстояний. Вектор $\tilde{z}_{ij}$ подаётся на вход классифицирующего перцептрона $g_\varphi^{(2)}$, имеющего ту же архитектуру, что и $g_\varphi^{(1)}$, но с выходным слоем размерности F_e, где F_e — число возможных тегов элементов. Результат — вектор вероятностей $\hat{q}_{ij} = \sigma(g_\varphi^{(2)}(\tilde{z}_{ij})) \in [0, 1]^{F_e}$. Для каждого тега элемента f выбирается порог и исходя из него определяются теги элементов аналогично этапу 1.

Набор типов фрагментов (F_s) и набор типов элементов (F_e) определяются рубрикатором задачи и могут полностью различаться. Поэтому пороги для этапа 2 подбираются независимо от этапа 1, причём набор подбираемых порогов относится к множеству типов элементов, а не фрагментов.

Функция потерь на этапе 2. Задача предсказания наличия элемента с определенными тегами для пары фрагментов (i, j) формулируется как задача классификации с F_e бинарными выходами. Функция потерь — бинарная кросс-энтропия:

$$\mathcal{L}_2(\varphi) = -\frac{1}{N_{pairs}^2 F_e} \sum_{i<j} \sum_{f=1}^{F_e} \Big[z_{ij}(f)\ \log \hat{q}_{ij}[f] + (1 - z_{ij}(f))\ \log(1 - \hat{q}_{ij}[f]) \Big],$$

где $z_{ij}(f) \in \{0, 1\}$ — индикатор того, что фрагменты i и j входят в элемент типа f.

3.4. Подход №2: модель на основе архитектуры Transformer

Подход на основе архитектуры Transformer [31] реализуется полностью аналогично подходу на основе RNN: используется та же двухэтапная схема с выделением фрагментов и элементов, те же классифицирующие перцептроны $g_\varphi^{(1)}$ и $g_\varphi^{(2)}$, то же разбиение на этапы обучения. Единственное отличие состоит в базовой модели: вместо Mamba используется mDeBERTa-v3-base, на основе которой строятся эмбеддинги токенов и фрагментов.

3.5. Подход №3: большие языковые модели

Принципиальным отличием данного подхода является замена этапа обучения на формирование контекста. В парадигме few-shot обучения [36] этап градиентного обучения модели заменяется на сбор информации о задаче из немногих размеченных примеров. Общая схема работы подхода представлена на Рис. 3.

Методология работы включает два этапа.

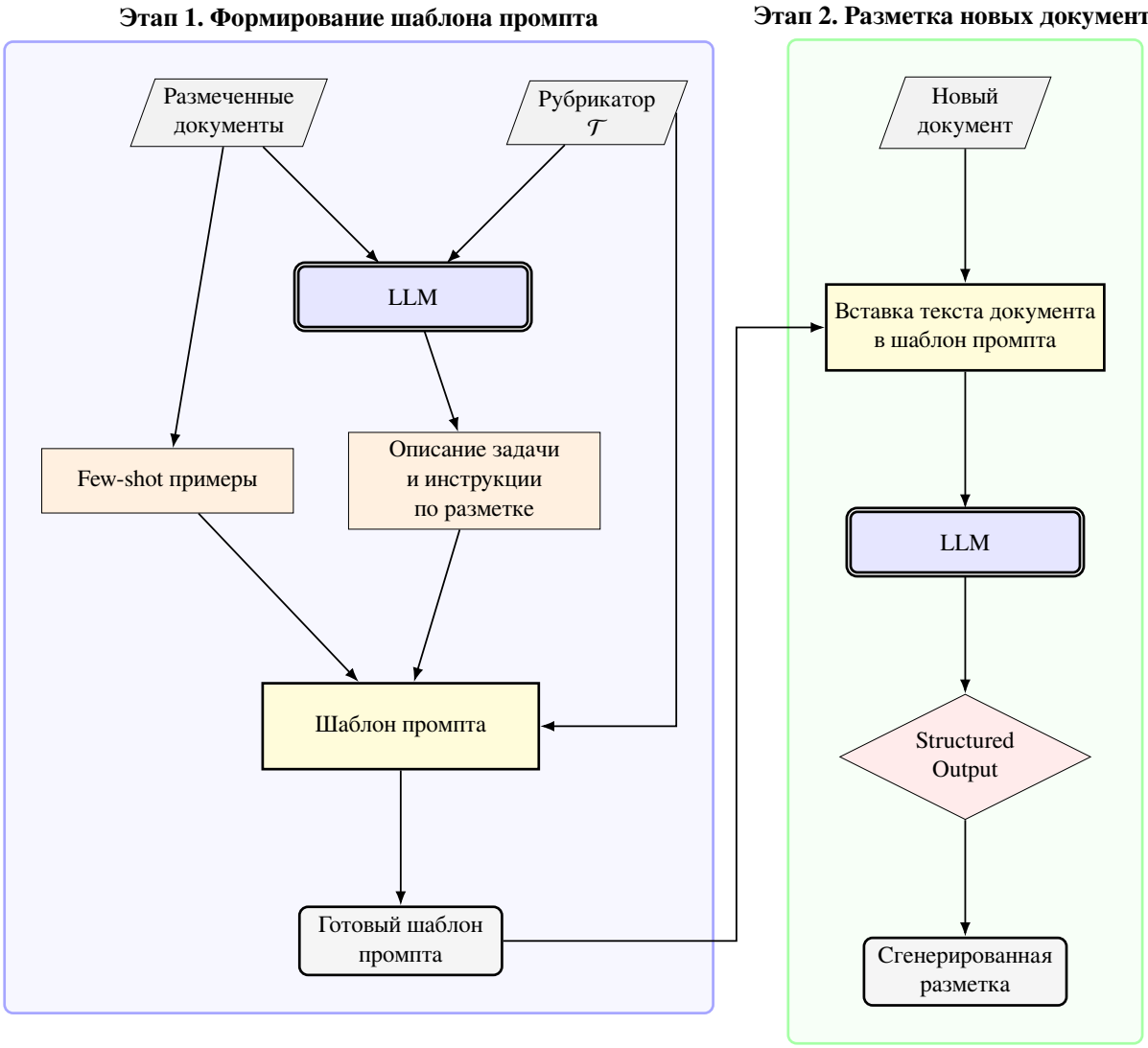

Рис. 3. Схема подхода на основе больших языковых моделей

Этап 1 (сбор контекста). На первом этапе большая языковая модель обрабатывает несколько размеченных документов из обучающей выборки и формирует промпт, включающий подробное описание заадачи и инструкции по составлению разметки.

В сгенерированный промпт автоматически добавляется следующая информация:

- рубрикатор $\mathcal{T}$ с описаниями тегов;
- примеры правильной разметки, иллюстрирующие типичные случаи (few-shot примеры).

Вся эта информация подается в промпт для того, чтобы модель могла обобщить структуру задачи и формат разметки без изменения своих параметров.

Этап 2 (инференс). Основная модель итеративно обрабатывает документы набора данных. Для каждого документа в контекст подаётся подготовленный на 1 этапе промпт и текст документа. Модель генерирует разметку документа на основе полученных инструкций.

Для обеспечения строгого оценивания ответа модели используется структурированный вывод: каждый из уровней универсального формата описан в виде схемы, определяющей формат ожидаемого ответа. Это гарантирует, что модель сгенерирует разметку в строго

определённом формате, после чего ее можно будет оценивать при помощи разработанных критериев качества, как и в предыдущих подходах.

4. Описание практической части

4.1. Методология проведения экспериментов

Эксперименты проводились с целью сравнительной оценки трёх предложенных подходов на бенчмарке универсальных моделей разметки. Для каждого набора данных бенчмарка выполнялось разбиение на обучающую, валидационную и тестовую выборки в соотношении 8:1:1. Все три подхода тестировались на каждом наборе данных с использованием единых критериев качества C_1–C_5.

Подходы на основе RNN и BERT. Дообучение выполнялось в два этапа, соответствующих двум уровням универсального формата данных. На каждом этапе обучались классифицирующие перцептроны $g_\varphi^{(1)}$ и $g_\varphi^{(2)}$ с использованием бинарной кросс-энтропии в качестве функции потерь. Применялось отрицательное семплирование для балансировки положительных и отрицательных примеров в батчах. Оптимизация выполнялась алгоритмом AdamW [41] со скоростью обучения $3 \cdot 10^{-5}$ и коэффициентом L_2-регуляризации 0.01. График скорости обучения: линейный разогрев на первых 10% шагов, затем линейный спад до нуля. Пороги θ_f для каждого типа фрагментов и элементов подбирались на валидационной выборке: после каждой эпохи перебирались 20 значений, равномерно распределенных по диапазону $[0.05, 0.8]$, и выбирался порог, максимизирующий сумму соответствующих уровню модели данных критериев (для 1 уровня — критерии C_1, C_2, C_3, для второго — критерии C_4, C_5). Базовые модели оставались замороженными на протяжении всего обучения, они используются исключительно для получения обогащенных эмбеддингов токенов и фрагментов.

Подход на основе LLM. Для данного подхода этап градиентного обучения заменялся на сбор контекста о задаче, включаего в себя few-shot примеры. В качестве базовой модели использовался Claude Sonnet 4.6. Для каждого набора данных отбиралось 3 примера корректной разметки. Промпт формировался полуавтоматически и включал рубрикатор $\mathcal{T}$ с описаниями тегов, описание задачи (полученное при помощи LLM) и отобранные примеры. При инференсе применялся структурированный вывод (structured output), гарантирующий генерацию разметки в строгом соответствии с универсальным форматом.

Оценивание. Для каждого набора данных и каждого подхода вычислялись критерии C_1–C_5 путём сопоставления предсказанных разметок с экспертными. Для наборов данных с разметкой с перекрытием результат определялся как среднее по всем парам экспертных разметок.

4.2. Результаты экспериментов

В таблице 2 приведены значения критериев качества C_1–C_5 и их агрегированного значения C для каждого из трёх подходов на каждом наборе данных бенчмарка.

Набор данных	**Подход**	C_1	C_2	C_3	C_4	C_5	C
NEREL [2]	RNN	0.14	0.04	0.58	0.21	0.63	0.32
	BERT	0.09	0.12	0.66	0.07	0.57	0.30
	LLM	0.71	0.62	0.79	0.54	0.74	0.68
RuSentNE [3]	RNN	0.18	0.09	0.81	0.13	0.69	0.38
	BERT	0.07	0.15	0.62	0.19	0.64	0.33
	LLM	0.78	0.69	0.73	0.77	0.71	0.74
UpGreat READ//ABLE [4]	RNN	0.24	0.11	0.69	0.05	0.76	0.37
	BERT	0.10	0.08	0.74	0.16	0.58	0.33
	LLM	0.67	0.81	0.75	0.64	0.82	0.74
CoNLL 2012 Ontonotes [5]	RNN	0.06	0.17	0.71	0.10	0.64	0.34
	BERT	0.13	0.05	0.59	0.18	0.60	0.31
	LLM	0.63	0.74	0.77	0.61	0.76	0.70
SWDA [6]	RNN	0.27	0.14	0.74	0.22	0.68	0.41
	BERT	0.11	0.18	0.76	0.09	0.66	0.36
	LLM	0.81	0.70	0.73	0.69	0.82	0.75
Kaggle NER [7]	RNN	0.19	0.23	0.72	0.11	0.85	0.42
	BERT	0.12	0.09	0.79	0.15	0.65	0.36
	LLM	0.74	0.83	0.69	0.80	0.73	0.76
MultiCoNER [8]	RNN	0.13	0.03	0.75	0.08	0.66	0.33
	BERT	0.04	0.11	0.61	0.13	0.56	0.29
	LLM	0.76	0.65	0.72	0.63	0.79	0.71
RuTermEval Dialogue [10]	RNN	0.21	0.10	0.70	0.19	0.76	0.39
	BERT	0.08	0.16	0.73	0.07	0.71	0.35
	LLM	0.69	0.78	0.81	0.66	0.75	0.74
ADE [11]	RNN	0.25	0.09	0.83	0.18	0.75	0.42
	BERT	0.17	0.14	0.69	0.08	0.77	0.37
	LLM	0.82	0.72	0.77	0.79	0.80	0.78
DDI corpus [12]	RNN	0.10	0.15	0.73	0.07	0.67	0.34
	BERT	0.13	0.04	0.58	0.12	0.63	0.30
	LLM	0.75	0.66	0.70	0.73	0.71	0.71
PcMSP [13]	RNN	0.04	0.13	0.59	0.12	0.62	0.30
	BERT	0.08	0.02	0.63	0.03	0.54	0.26
	LLM	0.72	0.61	0.76	0.65	0.71	0.69

Набор данных	**Подход**	C_1	C_2	C_3	C_4	C_5	C
ChemProt [14]	RNN	0.29	0.12	0.79	0.24	0.76	0.44
	BERT	0.11	0.19	0.72	0.17	0.74	0.39
	LLM	0.83	0.76	0.85	0.71	0.80	0.79
NERRE [15]	RNN	0.15	0.05	0.63	0.14	0.68	0.33
	BERT	0.06	0.10	0.67	0.05	0.57	0.29
	LLM	0.74	0.63	0.71	0.75	0.68	0.70
RURED [16]	RNN	0.03	0.11	0.65	0.04	0.62	0.29
	BERT	0.07	0.01	0.53	0.09	0.55	0.25
	LLM	0.70	0.59	0.75	0.62	0.74	0.68
SciERC [17]	RNN	0.22	0.08	0.71	0.20	0.74	0.39
	BERT	0.09	0.15	0.76	0.10	0.64	0.35
	LLM	0.79	0.71	0.75	0.77	0.73	0.75
RuSuperGLUE RWSD [18]	RNN	0.09	0.02	0.56	0.11	0.57	0.27
	BERT	0.01	0.06	0.60	0.02	0.48	0.23
	LLM	0.61	0.73	0.68	0.58	0.75	0.67
MERA RWSD [19]	RNN	0.11	0.03	0.68	0.06	0.62	0.30
	BERT	0.02	0.08	0.55	0.07	0.58	0.26
	LLM	0.71	0.65	0.77	0.61	0.72	0.69
MERA Ruethics [19]	RNN	0.16	0.07	0.79	0.09	0.69	0.36
	BERT	0.05	0.12	0.63	0.14	0.61	0.31
	LLM	0.75	0.69	0.80	0.65	0.76	0.73
SemEval 2010 Task 8 [20]	RNN	0.18	0.21	0.75	0.14	0.82	0.42
	BERT	0.19	0.07	0.69	0.16	0.74	0.37
	LLM	0.80	0.74	0.77	0.81	0.78	0.78
SemEval-2018 Task 7 [21]	RNN	0.12	0.18	0.71	0.10	0.74	0.37
	BERT	0.15	0.05	0.66	0.13	0.64	0.33
	LLM	0.77	0.68	0.79	0.70	0.76	0.74
Human Values [22]	RNN	0.05	0.12	0.70	0.06	0.67	0.32
	BERT	0.10	0.03	0.57	0.09	0.61	0.28
	LLM	0.73	0.64	0.76	0.62	0.75	0.70

Таблица 2: Значения критериев качества C_1–C_5 для различных подходов на наборах данных бенчмарка

В таблице 3 приведены средние значения критериев C_1–C_5 и агрегированного критерия C по всем наборам данных бенчмарка. По ней можно оценить усредненное качество работы

каждого из подходов в разных аспектах задачи разметки. Жирным в каждом из столбцов выделен лучший результат для соответствующего критерия.

Подход	C_1	C_2	C_3	C_4	C_5	C
RNN	0.15	0.11	**0.71**	0.13	0.70	0.36
BERT	0.09	0.10	0.66	0.11	0.62	0.32
LLM	**0.74**	**0.70**	0.65	**0.68**	**0.75**	**0.73**

Таблица 3. Средние значения критериев C_1–C_5 и агрегированного критерия C по бенчмарку

4.3. Анализ результатов

На основании полученных результатов можно сделать следующие выводы.

Подход на основе LLM демонстрирует наилучшее качество по всем критериям с существенным отрывом ($C \approx 0.71$). Значения C_1–C_5 для данного подхода примерно одинаковы и лежат в диапазоне 0.69–0.74, что свидетельствует о сбалансированном качестве работы модели по всем аспектам задачи. В отсутствие этапа градиентного обучения модель позволяет решить задачу достаточно качественно, задействуя лишь свои встроенные языковые знания, что особенно важно для задач с малым объёмом обучающих данных. Это говорит о том, что для качественного решения задачи при помощи LLM достаточно всего нескольких примеров от разметчиков-экспертов, скорость и стоимость получения которых и являлось основной проблемой, которую должна была решить модель обобщенного контент-анализа. Таким образом, этот подход продемонстрировал успешное достижение поставленных целей.

Подход на основе RNN показывает промежуточные результаты ($C \approx 0.12$). При этом характерен следующий паттерн: критерии C_3 и C_5, отвечающие за точность совпадения тегов, заметно превышают критерии C_1, C_2, C_4, отвечающие за локализацию фрагментов и элементов. Более того, они находятся на сравнимых с соответствующими критериями для подхода на основе LLM уровнях. Это говорит о том, что модель испытывает значительные трудности с выделением границ фрагментов и элементов в тексте, однако если локализация выполнена, теги определяются с относительно высокой точностью.

Подход на основе BERT показывает наименьшее общее качество ($C \approx 0.08$). Для данного подхода характерен тот же паттерн: $C_3 > C_{1,2}$, $C_5 > C_4$, однако выражен он ещё более ярко, чем для RNN. Transformer-архитектура в данной конфигурации не обеспечивает преимущества над рекуррентной моделью, что может объясняться малым объёмом данных для дообучения и заморозкой базовой модели.

Общая закономерность: для подходов с дообучением (RNN, BERT) характерна значительная разница между критериями локализации (C_1, C_2, C_4) и критериями совпадения тегов (C_3, C_5). Это указывает на то, что основная проблема данных подходов — именно в локализации фрагментов и элементов в тексте, а не в присвоении тегов. Подход на основе LLM лишён этого недостатка благодаря способности обобщать на основе немногих примеров даже при отсутствии этапа дообучения.

4.4. Вычислительная сложность

С практической точки зрения важны также вычислительные характеристики подходов. Подходы на основе RNN и BERT требуют сопоставимых вычислительных ресурсов как на этапе обучения, так и на этапе инференса. При этом RNN демонстрирует несколько лучшую эффективность благодаря линейной сложности относительно длины последовательности, в то время как BERT характеризуется квадратичной сложностью attention-механизма. Оба подхода требуют значительных затрат на обучение, поскольку для каждого набора данных необходимо проводить градиентную оптимизацию классифицирующих перцептронов.

Подход на основе LLM не требует этапа обучения, что является его существенным преимуществом. Однако инференс на больших языковых моделях вычислительно затратен и требует специализированного оборудования (GPU с большим объёмом памяти) или использования платных сервисов, предоставляющих доступ к проприеарным LLM по API.

При выборе подхода для практического применения необходимо учитывать баланс между качеством, вычислительными затратами и объёмом доступных данных. Для задач с малыми обучающими выборками и высокими требованиями к качеству целесообразно использовать подход на основе LLM. Для задач с большими обучающими выборками и ограниченными вычислительными ресурсами может быть оправдано применение RNN, несмотря на меньшее качество. Кроме того, следует упомянуть, что на практике целесообразно рассматривать комбинации упомянутых подходов: локализация фрагментов может выполняться при помощи LLM (так как качество у остальных подходов на этой задаче слишком мало), а выделение тегов может выполняться менее ресурсозатратным подходом на основе RNN или BERT.

5. Заключение

В рамках настоящей работы исследована задача универсализации и автоматизации контент-анализа текстовых данных. Ниже перечислены основные результаты, выносимые на защиту.

5.1. Результаты, выносимые на защиту

1) **Критерии качества.** Предложены критерии качества C_1–C_5, адаптированные для оценки моделей обобщённого контент-анализа и учитывающие долю сопоставленных фрагментов, точность совпадения их границ и тегов, а также аналогичные характеристики для элементов.

2) **Алгоритм оптимального сопоставления разметок.** Разработан алгоритм сопоставления альтернативных экспертных разметок, позволяющий вычислить критерии качества C_1–C_5 с учётом особенностей разметки с перекрытием. Алгоритм основан на построении матрицы расстояний между фрагментами с использованием меры Жаккара, удалении заведомо нежелательных рёбер, разбиении на компоненты связности и применении венгерского алгоритма внутри каждой компоненты. Алгоритм решает оптимизационную задачу, в которой достигается баланс между различными аспектами качества.

3) **Подходы к реализации моделей.** Предложены и реализованы три подхода к построению моделей машинного обучения для решения задачи универсализации и автоматизации контент-анализа:

   - подход на основе рекуррентной нейронной сети;
   - подход на основе Transformer-модели;
   - подход на основе больших языковых моделей без дополнительного обучения.

   Для подходов с дообучением (RNN, BERT) разработана двухэтапная схема, соответствующая трёхуровневой структуре универсального формата данных, и введены математические функции потерь. Для LLM-подхода разработана двухэтапная методология промптинга.

4) **Экспериментальное исследование.** Проведены вычислительные эксперименты на бенчмарке, включающем 21 набор данных и 17 типов задач. Эксперименты подтвердили работоспособность предложенных подходов и позволили провести их сравнительный анализ.

5.2. Основные выводы

На основании проведённых экспериментов можно сделать следующие выводы.

Подход на основе LLM значительно превосходит остальные подходы по всем критериям ($C \approx 0.71$), причём все пять критериев имеют близкие значения. Это объясняется тем, что LLM не требует дообучения и способна обобщать на основе встроенных языковых знаний и корректных примеров разметки. При этом LLM является самой ресурсоемкой моделью на этапе инференса, так как ее размеры могут превосходить модели других подходов на порядки.

Подход на основе RNN занимает промежуточное положение ($C \approx 0.12$), причём для него характерен выраженный паттерн: критерии C_3 и C_5 (точность совпадения тегов) заметно превышают критерии C_1, C_2, C_4 (локализация фрагментов и элементов). Это свидетельствует о том, что основная проблема подхода — в выделении границ фрагментов и элементов в тексте, тогда как присвоение тегов выполняется с относительно высокой точностью, сравнимой с результатами подхода на основе LLM. Подход на основе RNN требует довольно большое количество ресурсов на дообучение, но при этом затраты на этапе инференса намного меньше в сравнении с подходом на основе LLM.

Подход на основе BERT показывает наименьшее, хотя и не слишком сильно отличающееся от подхода на основе RNN, качество ($C \approx 0.08$), с ещё более выраженным аналогичным паттерном. Transformer-архитектура в данной конфигурации не обеспечивает преимущества над рекуррентной моделью, что может объясняться малым объёмом данных для дообучения и заморозкой базовой модели или простотой обучаемого слоя, который в силу недостатка параметов не смог раскрыть потенциал эмбеддингов, генерируемых базовой моделью. Расходование ресурсов для этого подхода аналогично подходу на основе RNN как для этапа дообучения, так и для этапа инференса. При этом важно отметить, что усложнение обучаемого слоя приведет к еще большим затратам на обучение, что может несколько улучшить результаты этого подхода, но сделает совершенно нецелесообразным его использование в реальных задачах.

Общая закономерность состоит в том, что для подходов с дообучением характерна значительная разница между критериями локализации и критериями совпадения тегов, в то время как LLM демонстрирует сбалансированное качество по всем аспектам задачи.

Комбинирование подходов демонстрирует потенциал для дальнейшего улучшения качества: объединение сильных сторон различных моделей может позволить превзойти результаты каждого отдельного подхода как по итоговому качеству, так и по затратам на вычислительные ресурсы. Важно отметить, что для отдельных задач разные подходы могут показывать себя лучше или хуже остальных. Приведенные выше выводы основаны на усредненных результатах, их можно считать репрезентативными, так как для проведения экспериментов использовался бенчмарк, покрывающий основной объем задач разметки текстов.

5.3. Степень решения поставленной задачи

Разработанные алгоритм сопоставления разметок, критерии качества и подходы к построению моделей в совокупности обеспечивают решение задачи универсализации и авто-

матизации контент-анализа в значительной степени. Предложенные инструменты позволяют обрабатывать текстовые данные из произвольной предметной области с использованием единого подхода, что подтверждено экспериментами на бенчмарке с широким покрытием задач различных доменов и сигнатур.

Вместе с тем результаты указывают на наличие резерва для улучшения качества подходов с дообучением, в частности за счёт развития стратегий объединения моделей, а также на необходимость дальнейшего расширения экспериментов с методикой дообучения моделей, так как на результат могли повлиять неудачно выбранные гиперпараметры обучения.

Список литературы

1. Automated Detection of Human Values in Texts: ML Challenges and Performance Benchmarks / O. Rink [и др.] // Interacción. — 2025. — URL: https://api.semanticscholar.org/CorpusID:279266154.
2. NEREL: a Russian information extraction dataset with rich annotation for nested entities, relations, and wikidata entity links / N. V. Loukachevitch [и др.] // Language Resources and Evaluation. — 2023. — С. 1—37. — URL: https://api.semanticscholar.org/CorpusID:262168528.
3. *Golubev A.*, *Rusnachenko N.*, *Loukachevitch N. V.* RuSentNE-2023: Evaluating Entity-Oriented Sentiment Analysis on Russian News Texts // ArXiv. — 2023. — Т. abs/2305.17679. — URL: https://api.semanticscholar.org/CorpusID:258960182.
4. The methodology of multi-criteria evaluation of text markup models based on inconsistent expert markup / A. Levikin [и др.] // Computational Linguistics and Intellectual Technologies. — 2025. — URL: https://api.semanticscholar.org/CorpusID:280953166.
5. Towards Robust Linguistic Analysis using OntoNotes / S. Pradhan [и др.] // Conference on Computational Natural Language Learning. — 2013. — URL: https://api.semanticscholar.org/CorpusID:14515377.
6. Dialogue act modeling for automatic tagging and recognition of conversational speech / A. Stolcke [и др.] // Computational Linguistics. — 2000. — Т. 26. — С. 339—373. — URL: https://api.semanticscholar.org/CorpusID:215825908.
7. *N. A.-Q.* Named Entity Recognition NER Corpus / Kaggle. — URL: https://www.kaggle.com/datasets/naseralqaydeh/named-entity-recognition-ner-corpus (дата обр. 14.05.2025).
8. MultiCoNER: A Large-scale Multilingual Dataset for Complex Named Entity Recognition / S. Malmasi [и др.] // International Conference on Computational Linguistics. — 2022. — URL: https://api.semanticscholar.org/CorpusID:251953674.
9. Wikipedia / Wikipedia. — URL: https://ru.wikipedia.org (дата обр. 18.05.2025).
10. *Mamontova A. N.*, *Ischenko R.* RuTermEval-2024: Cross-domain Automatic Term Extraction and Classification in Russian scientific texts // Computational Linguistics and Intellectual Technologies. — 2025. — URL: https://api.semanticscholar.org/CorpusID:280953044.

11. *Gurulingappa H.*, *Mateen-Rajput A.*, *Toldo L.* Extraction of potential adverse drug events from medical case reports // Journal of Biomedical Semantics. — 2012. — T. 3. — C. 15—15. — URL: https://api.semanticscholar.org/CorpusID:9831785.

12. The DDI corpus: An annotated corpus with pharmacological substances and drug-drug interactions / M. Herrero-Zazo [и др.] // Journal of biomedical informatics. — 2013. — T. 46 5. — C. 914—20. — URL: https://api.semanticscholar.org/CorpusID:23935739.

13. PcMSP: A Dataset for Scientific Action Graphs Extraction from Polycrystalline Materials Synthesis Procedure Text / X. Yang [и др.] // Conference on Empirical Methods in Natural Language Processing. — 2022. — URL: https://api.semanticscholar.org/CorpusID:253098009.

14. Overview of the BioCreative VI chemical-protein interaction Track / M. Krallinger [и др.] //. — 2017. — URL: https://api.semanticscholar.org/CorpusID:13690520.

15. Structured information extraction from scientific text with large language models / J. Dagdelen [и др.] // Nature Communications. — 2024. — T. 15. — URL: https://api.semanticscholar.org/CorpusID:267700596.

16. RELATION EXTRACTION DATASET FOR THE RUSSIAN / D. Gordeev [и др.] // Computational Linguistics and Intellectual Technologies. — 2020. — URL: https://api.semanticscholar.org/CorpusID:229204578.

17. Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction / Y. Luan [и др.] // ArXiv. — 2018. — T. abs/1808.09602. — URL: https://api.semanticscholar.org/CorpusID:52118895.

18. RussianSuperGLUE: A Russian Language Understanding Evaluation Benchmark / T. Shavrina [и др.] // ArXiv. — 2020. — T. abs/2010.15925. — URL: https://api.semanticscholar.org/CorpusID:226222281.

19. MERA: A Comprehensive LLM Evaluation in Russian / A. Fenogenova [и др.] // ArXiv. — 2024. — T. abs/2401.04531. — URL: https://api.semanticscholar.org/CorpusID:266899830.

20. SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals / I. Hendrickx [и др.] // International Workshop on Semantic Evaluation. — 2010. — URL: https://api.semanticscholar.org/CorpusID:260557571.

21. SemEval-2018 Task 7: Semantic Relation Extraction and Classification in Scientific Papers / K. Gábor [и др.] // International Workshop on Semantic Evaluation. — 2018. — URL: https://api.semanticscholar.org/CorpusID:44163645.

22. *Rink O.*, *Lobachev V.*, *Vorontsov K. V.* Detecting Human Values and Sentiments in Large Text Collections with a Context-Dependent Information Markup: A Methodology and Math // Interacción. — 2024. — URL: https://api.semanticscholar.org/CorpusID:270393808.

23. *Kim N.*, *Park C.* Inter-Annotator Agreement in the Wild: Uncovering Its Emerging Roles and Considerations in Real-World Scenarios // ArXiv. — 2023. — T. abs/2306.14373. — URL: https://api.semanticscholar.org/CorpusID:259252349.

24. *James J.* Counting on Consensus: Selecting the Right Inter-annotator Agreement Metric for NLP Annotation and Evaluation //. — 2026. — URL: https://api.semanticscholar.org/CorpusID:286372110.

25. A Joint Neural Model for Information Extraction with Global Features / Y. Lin [и др.] // Annual Meeting of the Association for Computational Linguistics. — 2020. — URL: https://api.semanticscholar.org/CorpusID:220048375.

26. SemEval-2023 Task 4: ValueEval: Identification of Human Values Behind Arguments / J. Kiesel [и др.] // International Workshop on Semantic Evaluation. — 2023. — URL: https://api.semanticscholar.org/CorpusID:259376541.

27. *Schwartz S. H.* Universals in the Content and Structure of Values: Theoretical Advances and Empirical Tests in 20 Countries // Advances in Experimental Social Psychology. — 1992. — T. 25. — С. 1—65. — URL: https://api.semanticscholar.org/CorpusID:14089770.

28. *Stone P. J.*, *Dunphy D. C.*, *Smith M. S.* The general inquirer: A computer approach to content analysis. // American Educational Research Journal. — 1967. — Т. 4. — С. 397. — URL: https://api.semanticscholar.org/CorpusID:60936250.

29. *Tausczik Y. R.*, *Pennebaker J. W.* The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods // Journal of Language and Social Psychology. — 2010. — Т. 29. — С. 24—54. — URL: https://api.semanticscholar.org/CorpusID:145665613.

30. *Lozano E. S.*, *Nakayama A.* Text-Mining Approach to Political Communication on Twitter: The Analysis of the Discourse of Spain's Principal Political Parties During the European Parliament Elections in 2019 // Strategic Communication in Context: Theoretical Debates and Applied Research. — 2021. — URL: https://api.semanticscholar.org/CorpusID:242974690.

31. Attention is All you Need / A. Vaswani [и др.] // Neural Information Processing Systems. — 2017. — URL: https://api.semanticscholar.org/CorpusID:13756489.

32. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding / J. Devlin [и др.] // North American Chapter of the Association for Computational Linguistics. — 2019. — URL: https://api.semanticscholar.org/CorpusID:52967399.

33. RoBERTa: A Robustly Optimized BERT Pretraining Approach / Y. Liu [и др.] // ArXiv. — 2019. — Т. abs/1907.11692. — URL: https://api.semanticscholar.org/CorpusID:198953378.

34. *Gu A.*, *Dao T.* Mamba: Linear-Time Sequence Modeling with Selective State Spaces // ArXiv. — 2023. — Т. abs/2312.00752. — URL: https://api.semanticscholar.org/CorpusID:265551773.

35. Few-Shot Multilingual Coreference Resolution Using Long-Context Large Language Models / M. Sajid [и др.] // Proceedings of the Eighth Workshop on Computational Models of Reference, Anaphora and Coreference. — 2025. — URL: https://api.semanticscholar.org/CorpusID:282904510.

36. Language Models are Few-Shot Learners / T. B. Brown [и др.] // ArXiv. — 2020. — T. abs/2005.14165. — URL: https://api.semanticscholar.org/CorpusID:218971783.

37. LLM-as-an-Annotator: Training Lightweight Models with LLM-Annotated Examples for Aspect Sentiment Tuple Prediction / N. C. Hellwig [и др.] //. — 2026. — URL: https://api.semanticscholar.org/CorpusID:286224240.

38. ChatGPT for Automated Qualitative Research: Content Analysis / R. Bijker [и др.] // Journal of Medical Internet Research. — 2024. — T. 26. — URL: https://api.semanticscholar.org/CorpusID:271457259.

39. A Unified View of Evaluation Metrics for Structured Prediction / Y. Chen [и др.] // ArXiv. — 2023. — T. abs/2310.13793. — URL: https://api.semanticscholar.org/CorpusID:264426523.

40. *Kuhn H. W.* The Hungarian method for the assignment problem // Naval Research Logistics (NRL). — 1955. — T. 52. — URL: https://api.semanticscholar.org/CorpusID:9426884.

41. *Loshchilov I.*, *Hutter F.* Decoupled Weight Decay Regularization // International Conference on Learning Representations. — 2017. — URL: https://api.semanticscholar.org/CorpusID:53592270.