МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

Разработка системы автоматического сбора и выявления дубликатов текстового контента

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студентки 4 курса, 441 группы
направления (специальности) 02.03.03 «Математическое обеспечение и администрирование информационных систем»
факультета Компьютерных наук и информационных технологий
Тимофеевой Владиславы Михайловны

Научный руководитель
Зав. кафедрой ИиП, к.ф.-м.н., доцент _____ Огнева М. В.

Зав. кафедрой
к.ф.-м.н., доцент _____ Огнева М. В.

Саратов 2026

ВВЕДЕНИЕ

В современном цифровом пространстве количество текстовых публикаций растёт экспоненциально. Новостные агрегаторы, блоги и социальные сети сталкиваются с проблемой дублированного контента: одна и та же публикация появляется в виде точных копий, репостов или переформулированных материалов. Дубликаты снижают качество пользовательской ленты, увеличивают затраты на хранение и замедляют работу аналитических систем. В машинном обучении избыточность данных ведёт к переобучению моделей. Традиционные методы (точное совпадение хешей, сравнение длины текста) не справляются с почти дубликатами – текстами, отличающимися заголовком, структурой или редакционными правками. Современные подходы, такие как вероятностное хеширование (MinHash, SimHash) и нейросетевые эмбеддинги, требуют экспериментального обоснования для русскоязычного контента мессенджеров и новостных сайтов. Таким образом, разработка эффективной системы дедупликации с обоснованным выбором методов и порогов классификации является актуальной задачей.

Цель выпускной квалификационной работы: разработать систему сбора и дедупликации текстового контента для приложения-агрегатора и экспериментально оценить её качество на русскоязычных корпусах.

Задачи:

1. Провести анализ методов предобработки, вероятностного хеширования, лексического и семантического сравнения текстов, систематизировать их характеристики применительно к задаче дедупликации русскоязычного контента.
2. Описать математические основы и формальные постановки методов вероятностного хеширования (MinHash, SimHash), лексико-статистических методов (TF-IDF, BM25) и нейросетевых методов построения эмбеддингов (RuBERT, LaBSE, SBERT, BGE-M3), а также

определить метрики качества (точность, полнота, F1-мера, ROC-AUC, macro-F1) для трёхклассовой задачи дедупликации текстового контента

3. Выбрать корпус ParaPhraser.ru в качестве основного контрольного корпуса и описать процедуру подготовки данных для трёхклассовой задачи классификации пар.
4. Реализовать и протестировать 11 методов дедупликации, охватывающих базовые (Exact Match, Jaccard, MinHash, SimHash, Levenshtein), статистические (TF-IDF, BM25) и нейросетевые (RuBERT, LaBSE, SBERT, BGE-M3) подходы и провести сравнительный анализ методов на полном корпусе из 9 151 пар с вычислением macro-F1, ROC-AUC и матриц ошибок.
5. Исследовать чувствительность выбранных методов к порогам классификации и определить оптимальные рабочие точки.
6. Выполнить эксперимент по отключению компонент производственного конвейера для оценки вклада каждой стадии.
7. Провести доменную валидацию на выборке постов Telegram-каналов для проверки переноса результатов с корпуса заголовков на полнотекстовый контент мессенджера.
8. Спроектировать архитектуру системы дедупликации с шестистадийным конвейером обработки и реализовать систему на Python с использованием PostgreSQL, расширения pgvector и индекса HNSW для приближённого поиска ближайших соседей.
9. Обосновать инженерные решения (выбор модели BGE-M3, двухпороговая классификация, граф связей с тремя типами рёбер) на основе количественных результатов экспериментальной главы.
10. Провести экспериментальное исследование разработанной системы и проинтерпретировать результаты.

Основной экспериментальный корпус – ParaPhraser.ru, содержащий ~9000 пару русскоязычных заголовков новостей с экспертной разметкой трёх классов: NONE (не связаны), RELATED (частичные дубликаты/тематически связаны),

DUPLICATE (полные парафразы). Для доменной валидации использована выборка из ~150000 постов 48 русскоязычных Telegram-каналов.

Выпускная квалификационная работа состоит из введения, пяти глав, заключения, списка литературы и четырёх приложений с исходными кодами. В первой главе выполнен анализ существующих методов дедупликации и способов сбора контента. Во второй главе формально описаны базовые методы сравнения текстов. Третья глава содержит реализацию и сравнительный анализ 11 методов на корпусе ParaPhraser.ru, исследование чувствительности к порогам, отключение компонент конвейера и доменную валидацию. Четвёртая глава посвящена архитектуре и реализации подсистемы дедупликации. Пятая глава описывает систему сбора контента из трёх микросервисов.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

В разделе 1 рассмотрены основные подходы к дедупликации текстов. Описана предобработка (лемматизация, стемминг, удаление стоп-слов, нормализация Unicode), методы вероятностного хеширования MinHash и SimHash, позволяющие оценивать лексическое перекрытие за субквадратичное время с помощью LSH. Проанализированы лексико-статистические методы: TF-IDF и BM25, учитывающие важность терминов и нормализацию по длине документа. Нейросетевые эмбеддинги на основе BERT (RuBERT, LaBSE, Sentence-BERT, BGE-M3) обеспечивают семантическое сравнение, но требуют векторного поиска. Рассмотрены архитектурные подходы к производственным системам (гибридные конвейеры, кластеризация через пороговые графы) и способы сбора контента (API, скрапинг, TDLib, RSS).

В разделе 2 представлены формальные постановки базовых методов: точное совпадение строк (Exact Match), коэффициент Жаккара на основе n-грамм, алгоритмы MinHash и SimHash, расстояние Левенштейна. Для каждого метода приведены принципы работы, достоинства и ограничения.

В разделе 3 проведён экспериментальный сравнительный анализ 11 методов дедупликации. Использован полный корпус ParaPhraser.ru (9 151 пара) с тремя классами. Основная метрика – macro-F1 (устойчива к дисбалансу). В таблице 1 приведены результаты лучших методов в режиме оптимальных порогов.

Таблица 1 – Сравнение методов на полном корпусе ParaPhraser.ru, оптимальные пороги

Метод	macro-F1	F1DUP	F1REL	F1NONE	θ_dup (cos)	θ_rel (cos)
BGE-M3	0.694	0.617	0.661	0.804	0.915	0.746
LaBSE	0.651	0.612	0.605	0.736	0.893	0.733
RuBERT-tiny2	0.643	0.581	0.609	0.739	0.922	0.805

Метод	macro-F1	F1DUP	F1REL	F1NONE	θ_dup (cos)	θ_rel (cos)
mpnet-paraphrase	0.634	0.581	0.595	0.726	0.915	0.716
TF-IDF char-3	0.590	0.517	0.554	0.700	0.385	-0.128
Jaccard char-3	0.578	0.503	0.569	0.663	0.011	-0.519
MinHash-128	0.570	0.519	0.545	0.644	-0.071	-0.560
Jaccard char-5	0.559	0.464	0.560	0.651	-0.147	-0.668
TF-IDF word 1-2	0.551	0.496	0.509	0.648	-0.128	-0.692
Левенштейн	0.531	0.455	0.534	0.605	0.288	-0.380
SimHash-64	0.526	0.450	0.515	0.614	0.510	0.264

Модель BGE-M3 превосходит ближайшего конкурента LaBSE на 4,3 п.п. и лучший лексический метод (TF-IDF) на 10,4 п.п. Производственные пороги θ_{dup}=0,85, θ_{rel}=0,70 обеспечивают macro-F1=0,632, потеря относительно оптимума составляет 0,062. Эксперимент по отключению компонент конвейера показал, что стадия SHA-256 на корпусе ParaPhraser.ru не даёт вклада (нет точных копий), но в реальных Telegram-данных обрабатывает повторные публикации без вызова GPU. Доменная валидация на 99 парах Telegram-корпуса подтвердила, что BGE-M3 сохраняет разумное поведение: распределение предсказаний соответствует стратифицированным диапазонам косинусного сходства. Анализ ошибок выявил три типа ошибок: мягкие на границе DUPLICATE/RELATED (41,7% истинных DUPLICATE отнесены к RELATED), средние на границе NONE/RELATED и грубые (1% пар). Основной

источник потерь – перекрытие распределений score классов DUPLICATE и RELATED в зоне cos∈[0,60; 0,80], что отражает принципиальную семантическую сложность задачи.

В разделе 4 описана реализация подсистемы дедупликации на Python с применением чистой архитектуры. Подсистема работает как фоновый сервис, опрашивающий таблицу raw_content через SELECT … FOR UPDATE SKIP LOCKED. Шестистадийный конвейер включает:

1. нормализацию текста (NFC, нижний регистр, схлопывание пробелов);
2. вычисление SHA-256 и поиск точных совпадений;
3. кодирование эмбеддингов BGE-M3 в режиме FP16 (размерность 1024);
4. поиск k=20 ближайших соседей через индекс HNSW pgvector с временным окном 72 часа;
5. двухпороговую классификацию (DUPLICATE при score≥0,85, RELATED при score≥0,70);
6. запись рёбер графа в таблицу similarities с UPSERT-условием GREATEST.

Инженерные решения обоснованы экспериментально: выбор BGE-M3 (наивысший macro-F1), FP16 (сохранение качества при двукратном сокращении памяти), пороги 0,85/0,70 (минимальная потеря F1 на классе DUPLICATE), pgvector вместо отдельного векторного хранилища (транзакционная согласованность), Clean Architecture (тестируемость и замена адаптеров).

В разделе 5 рассмотрена система сбора контента, состоящая из трёх микросервисов на Kotlin/Spring Boot. Сервис управления конфигурацией (config-service) хранит источники (Habr, VC.ru, Telegram) в PostgreSQL и публикует события в Kafka. Сервис парсинга (content-parser-service) по расписанию Quartz опрашивает источники через специализированные обработчики (REST API для Habr/VC.ru, TDLight для Telegram) с использованием Resilience4j (RateLimiter, CircuitBreaker, Retry). Публикации сохраняются в raw_content с флагом is_processed_by_dedup=false. Сервис выдачи (content-aggregator-service) предоставляет REST API только на чтение для клиентского приложения, работая с таблицей published_content, которая

формируется после дедупликации. Взаимодействие сервисов построено на общей схеме PostgreSQL (шина данных) и Kafka (только для событий конфигурации), что исключает дополнительные точки отказа.

ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы разработана подсистема дедупликации для приложения-агрегатора текстового контента. Подсистема реализована на Python, интегрирована с PostgreSQL и pgvector, обеспечивает обнаружение точных копий, почти дубликатов и тематически связанных публикаций в потоковом режиме.

Проведён систематический обзор методов предобработки, вероятностного хеширования, лексико-статистического и семантического сравнения. На полном корпусе ParaPhraser.ru (~9000 пар) реализован и протестирован конвейер из 11 методов. Наилучший результат показала модель BGE-M3 (macro-F1=0,694), что на 4,3 п.п. выше LaBSE и на 10,4 п.п. выше лучшего лексического метода. Производственные пороги 0,85 и 0,70 обеспечивают macro-F1=0,632 с потерей всего 0,062 относительно оптимума. Эксперимент по отключению компонент подтвердил, что стадия SHA-256 разгружает GPU-инференс на точных копиях (в реальных Telegram-данных такие копии существуют). Доменная валидация на корпусе из ~150000 постов Telegram-каналов показала переносимость результатов: BGE-M3 корректно обрабатывает кросс-канальные парафразы и точные самоповторы.

Разработанная подсистема дедупликации внедрена в общую архитектуру системы агрегации контента, обеспечивая высокое качество фильтрации дубликатов и масштабируемость за счёт асинхронной обработки через общую базу данных.

Отдельные части бакалаврской работы были опубликованы/представлены на конференции: Тимофеева В.М., Устимов М.Д. РАЗРАБОТКА СИСТЕМЫ ДЛЯ СБОРА И ДЕДУПЛИКАЦИИ КОНТЕНТА ДЛЯ СИСТЕМЫ РЕКОМЕНДАЦИЙ В ПРИЛОЖЕНИИ ДЛЯ АГРЕГАЦИИ КОНТЕНТА // «СНК СГУ-2026»

Основные источники информации:

1. Протасова А. А. Эффективное использование данных: технология дедупликации // Информационные технологии в науке, управлении,

социальной сфере и медицине : сборник научных трудов IV Международной научной конференции, 5-8 декабря 2017 г., Томск. Ч. 2. Томск : Изд-во ТПУ, 2017. С. 74-77.

2. Янников И. М., Ершова М. В., Исенбаев А. Н. Методы и алгоритмы для поиска сходства между текстами // Интеллектуальные системы в производстве. 2024. Т. 22, № 2. С. 103-113. DOI: 10.22213/2410-9304-2024-2-103-113.
3. Куницын В. И. и др. Дедупликация данных обучения: SimHash vs MinHash vs векторные методы и их влияние на переобучение и разнообразие корпуса // Студенческие научные исследования. 2025. Вып. 3. С. 43.
4. Буянов И. О., Ядринцев В. В., Соченков И. В. Нейросетевые методы сжатия векторов для задачи приближённого поиска ближайших соседей // Труды Института системного программирования РАН. 2024. Т. 36, № 1. С. 7-22.
5. Зори С. А., Рудак Л. В. Обработка текста методами естественного языка // Информатика и кибернетика. 2024. № 3 (37). С. 39-44.
6. Christen P. Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. Berlin Heidelberg : Springer-Verlag, 2012. 270 p. ISBN 978-3-642-31163-5.
7. Жбанкова Е. А. Алгоритмы распознавания схожести текста в вопросно-ответных системах // Актуальные исследования. 2020. № 6. С. 11-15.
8. Broder A. Z. On the Resemblance and Containment of Documents // Proceedings of the Compression and Complexity of Sequences 1997. Salerno, 1997. P. 21-29.
9. Charikar M. S. Similarity Estimation Techniques from Rounding Algorithms // Proceedings of the 34th Annual ACM Symposium on Theory of Computing. Montreal, 2002. P. 380-388. DOI: 10.1145/509907.509965.

10. Robertson S., Zaragoza H. The Probabilistic Relevance Framework: BM25 and Beyond // Foundations and Trends in Information Retrieval. 2009. Vol. 3, № 4. P. 333-389. DOI: 10.1561/1500000019.