СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Проблема некорректных заимствований в научной и образовательной среде имеет многоуровневый характер и включает не только явный плагиат, но и более скрытые формы нарушения академической этики: ошибочные ссылки, неправильно приписанные источники, искаженные цитаты, отсутствие достоверной библиографии. Масштаб проблемы растет с увеличением объема научных работ и доступности технологий, позволяющих легко манипулировать текстами.

Проблема обнаружения нечетких дубликатов является одной из наиболее важных и трудных задач анализа веб-данных и поиска информации. Основным препятствием для успешного решения данной задачи является гигантский объем данных, что делает практически невозможным попарное сравнение текстов документов в разумное время [1].

Современные подходы к выявлению некорректных заимствований выходят далеко за рамки простого поиска текстовых совпадений. Требуется применение комплекса методов обработки естественного языка (NLP), машинного обучения, информационного поиска (IR) и автоматизированных систем верификации источников. Каждый класс алгоритмов решает специфическую задачу: от синтаксического обнаружения копируемых фрагментов до семантического анализа парафраза и проверки корректности ссылок на литературу [2].

Цель данной работы — классифицировать и описать основные методы и алгоритмы, применяемые для выявления некорректных заимствований.

Для достижения поставленной цели необходимо решить следующие задачи:

1) провести анализ литературы по методам детекции заимствований;
2) разработать классификацию методов и алгоритмов;
3) описать основные подходы каждого класса;
4) провести сравнительное экспериментальное исследование эффективности методов.

1 Классификация и основные подходы к обнаружению заимствований

1.1 Классификация алгоритмов и методов детекции

Все многообразие подходов к обнаружению некорректных заимствований можно систематизировать в четыре основных класса:

1) методы синтаксического сравнения и «отпечатков» текста;
2) лексико-статистические методы и метрики схожести;
3) методы семантического/смыслового сопоставления;
4) модели авторского стиля и стилометрия.

Такая классификация позволяет проанализировать каждый подход с точки зрения его особенностей, преимуществ и ограничений. Дополнительно рассматриваются алгоритмы анализа и верификации цитирования как специализированное направление детекции некорректных заимствований [1].

2 Синтаксические и лексико-статистические методы

2.1 Методы синтаксического сравнения и отпечатков текста

Для решения задачи обнаружения нечетких дубликатов текстов применяются методы синтаксического сравнения. Идея этих методов заключается в том, чтобы получить компактное представление текста, сохраняющее его уникальные черты, и сравнивать эти представления вместо полных текстов. Такое представление называется «отпечатком» или сигнатурой документа. Это позволяет значительно ускорить сравнение больших объемов текстовых данных. Основной принцип состоит в следующем: текст разбивается на перекрывающиеся или неперекрывающиеся фрагменты фиксированной длины, каждому фрагменту вычисляется хеш-значение, и затем сравниваются хеши [1].

2.1.1 Шинглы и метод Бродера

Одним из первых исследований в области нахождения нечетких дубликатов является работа А. Бродера, в которой был предложен синтаксический метод оценки сходства между документами, основанный на представлении документа в виде множества всевозможных последовательностей фиксированной длины k, состоящих из соседних слов. Такие последовательности были названы шинглами (k-грамм слов). Два документа считались похожими, если их множества шинглов существенно пересекались.

Мера схожести двух документов вычисляется с помощью коэффициента Жаккара:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|},$$

где A и B — множества шинглов двух документов. Значения коэффициента близкие к 1 указывают на высокую схожесть документов, а близкие к 0 — на отсутствие схожести.

Поскольку число шинглов примерно равно длине документа в словах, были предложены два метода сэмплирования для получения репрезентативных подмножеств. Первый метод оставлял только те шинглы, чьи дактилограммы (численные отпечатки, вычисляемые по алгоритму Рабина-Карпа) делились без

остатка на некоторое число m. Второй метод отбирал фиксированное число s шинглов с наименьшими значениями дактилограмм [1].

2.1.2 Алгоритм Winnowing

Winnowing — это модификация простого k-граммного анализа, предложенная для повышения эффективности. Алгоритм работает следующим образом: текст разбивается на k-граммы (подстроки из k символов); для каждой k-граммы вычисляется хеш-значение; из всех хешей выбираются только минимальные значения в пределах скользящего окна размером w. Эти избранные хеши образуют отпечаток документа, после чего отпечатки различных документов сравниваются для определения схожести. Преимущество алгоритма Winnowing состоит в устойчивости к перестановкам фрагментов и синтаксическим изменениям, при этом сохраняя низкую вычислительную сложность. Алгоритм используется в системах MOSS и JPlag для обнаружения плагиата в исходном коде и текстах [4].

2.1.3 Дактилограммы и алгоритм Рабина-Карпа

Одними из первых исследований в области нахождения нечетких дубликатов являются работы U. Manber и N. Heintze. Дактилограмма (также называемая «отпечатком» или хеш-сигнатурой) файла или документа включает все текстовые подстроки фиксированной длины. Численное значение дактилограмм вычисляется с помощью алгоритма случайных полиномов Рабина-Карпа. Дактилограмма отличается от простого отпечатка тем, что представляет собой набор множественных хеш-значений для разных подстрок, а не единственное значение. В качестве меры сходства двух документов используется отношение числа общих подстрок к размеру файла или документа. Алгоритм Рабина-Карпа основан на быстром вычислении хешей для перекрывающихся подстрок с использованием скользящего окна. Полиномиальное хеширование позволяет за $O(1)$ пересчитать хеш для следующей подстроки на основе хеша предыдущей. Метод особенно эффективен при поиске точных совпадений в больших массивах текста [1].

2.1.4 Мегашинглы и супершинглы

Дальнейшим развитием концепций Бродера являются исследования D. Fetterly. Для каждого документа вычисляются 84 дактилограммы по алгоритму Рабина-Карпа с помощью взаимно-однозначных и независимых функций. В результате каждый документ представлялся 84 шинглами, минимизирующими значение соответствующей функции. Затем 84 шингла разбиваются на 6 групп по 14 шинглов в каждой. Эти группы называются супершинглами. Документ представляется всевозможными попарными сочетаниями из 6 супершинглов, которые называются мегашинглами. Число таких мегашинглов равно 15. Два документа сходны по содержанию, если у них совпадает хотя бы один мегашингл. Ключевое преимущество данного алгоритма состоит в том, что любой документ (в том числе и очень маленький) всегда представляется вектором фиксированной длины, и сходство определяется простым сравнением координат вектора [1].

2.1.5 N-граммный анализ и SimHash

N-граммный анализ — одна из самых базовых, но действенных методик. Документ представляется как набор n-грамм (последовательности из n символов или слов). Сравнение документов производится по пересечению их n-грамм. Алгоритм не учитывает порядок n-грамм в документе и работает с ними как с множеством. SimHash расширяет эту идею: для каждого документа строится битовый отпечаток фиксированной длины путем комбинирования хешей его n-грамм. Два документа считаются схожими, если расстояние Хемминга между их отпечатками мало [5].

2.1.6 Locality Sensitive Hashing (LSH)

Locality Sensitive Hashing (LSH) — это техника быстрого поиска похожих объектов (документов, предложений, наборов слов или векторов признаков) в больших коллекциях. Алгоритм строит такие хеш-функции, что близкие объекты с высокой вероятностью попадают в одну и ту же «корзину» (bucket), а далекие — в разные корзины.

Типичный сценарий при поиске похожих текстов по словам выглядит сле-

дующим образом:

1) текст каждого документа преобразуется в компактное представление: множество шинглов, вектор численных признаков (например, TF · IDF) или семантический вектор документа.
2) для этих представлений выбирается семейство локально-чувствительных хеш-функций, максимизирующих вероятность совпадения хешей для близких объектов (по косинусной мере близости или по коэффициенту Жаккара).
3) хеш-значения группируются в несколько «полос» и заносятся в несколько хеш-таблиц. Объекты, имеющие одинаковые хеши в одной и той же полосе, попадают в одну корзину.
4) при поиске похожего документа или фрагмента текста его представление хешируется тем же набором функций. По полученным хешам находятся все объекты, попавшие в те же корзины во всех таблицах — это кандидаты на сходство по набору слов или признаков.
5) только для небольшой подвыборки кандидатов выполняется более точное сравнение (например, по косинусной мере близости или по пересечению шинглов).

За счет такого двухэтапного поиска (быстрый отбор кандидатов на совпадение + точная проверка) LSH позволяет за логарифмическое или даже сублогарифмическое время находить похожие тексты без полного попарного сравнения со всеми документами в базе. Метод особенно ценен для масштабируемых систем с миллионами или миллиардами документов [5].

2.2 Лексико-статистические методы и метрики схожести

Лексико-статистические методы и метрики схожести представляют собой класс алгоритмов, которые решают задачу выявления заимствований путем анализа статистических характеристик текста на уровне отдельных слов и терминов. В отличие от синтаксических методов, которые работают с символьными последовательностями и локальными структурами, лексико-статистические подходы рассматривают документ как набор терминов (слов) и их частотных характеристик.

Основная идея этого класса методов заключается в том, что два заим-

ствованных или скопированных текста, даже если они переформулированы или переставлены, будут содержать сходный набор ключевых слов и терминов. Метрики схожести позволяют количественно оценить степень совпадения между наборами слов двух документов и определить, насколько они семантически близки.

Для реализации лексико-статистических методов документы преобразуются в компактные представления: либо в виде векторов численных признаков (таких как TF · IDF), либо в виде бинарных сигнатур на основе отобранного подмножества слов. Затем для сравнения этих представлений применяются различные метрики расстояния и схожести: косинусная мера близости, коэффициент Жаккара, расстояние Евклида, расстояние Левенштейна и другие.

2.2.1 Численный признак TF · IDF и векторное пространство

Для решения задачи сравнения документов на основе терминов используются лексико-статистические методы. Одним из наиболее важных методов является представление документа в виде вектора численных признаков. TF · IDF (Term Frequency-Inverse Document Frequency) — это численный признак, характеризующий важность термина в документе относительно всей коллекции. Величина TF · IDF рассчитывается как произведение двух компонент: частоты термина в документе (TF) и его редкости в коллекции (IDF).

Такое представление документа по словам и их весам часто описывают моделью «мешка слов». В этой модели порядок слов в документе полностью игнорируется: важен только набор терминов и их частоты. Это делает метод устойчивым к перестановкам фраз, но менее чувствительным к изменениям порядка слов, критичным для смысла.

Построение вектора TF · IDF для документа происходит следующим образом:

1) строится частотный словарь документа;
2) для каждого слова вычисляется произведение TF · IDF;
3) вектор упорядочивается по убыванию этого произведения;
4) выбираются топ-N слов с наибольшими весами и сцепляются в алфавитном порядке;
5) в качестве сигнатуры документа вычисляется контрольная сумма (на-

пример, CRC32 или MD5) полученной строки.

Два текста сравниваются как векторы в многомерном пространстве, где каждое слово представляет одну координату. Схожесть между двумя такими векторами оценивается с помощью косинусной меры близости (косинус угла между векторами). Косинусная мера близости принимает значения от -1 до $+1$, где значение 1 означает полное совпадение, 0 означает ортогональность (отсутствие схожести), а -1 означает полную противоположность. Документы с высокой косинусной мерой близости (обычно выше порога 0.8) считаются потенциально плагиатными [6].

2.2.2 Численный признак TF · RIDF

Близкой по идее к признаку TF · IDF является модификация TF · RIDF (Term Frequency–Residual Inverse Document Frequency), основанная на понятии остаточной обратной документной частоты RIDF. Как показывают Зеленков и Сегалович, RIDF вводится как разность между “наблюдаемой” величиной IDF и её теоретическим ожиданием в модели Пуассона распределения слов по документам коллекции [1].

Пусть N — число документов в коллекции, df — число документов, в которых слово встречается хотя бы один раз, а cf — суммарная частота данного слова по всей коллекции. Тогда “обычная” обратная документная частота вычисляется как

$$\mathrm{IDF} = -\log \frac{df}{N},$$

а теоретически ожидаемое количество информации о факте появления слова в документе при пуассоновской модели задаётся выражением

$$P_{\mathrm{IDF}} = -\log\bigl(1 - \exp(-cf/N)\bigr).$$

Остаточная IDF (Residual IDF) определяется как

$$\mathrm{RIDF} = \mathrm{IDF} - P_{\mathrm{IDF}} = -\log \frac{df}{N} + \log\bigl(1 - \exp(-cf/N)\bigr).$$

Интерпретация величины RIDF состоит в том, что она измеряет прирост информации, содержащейся в реальном распределении слова по докумен-

там, по сравнению с равномерным случайным (пуассоновским) распределением. “Хорошие”, содержательные слова, как правило, распределены неравномерно и встречаются в относительно небольшом числе документов, что даёт большие значения RIDF; напротив, общеязыковые и малоинформативные слова рассеиваются практически равномерно по всей коллекции и имеют малые значения RIDF [1].

Практическая схема вычисления признака TF · RIDF у Зеленкова и Сегаловича выглядит следующим образом. По всей коллекции строится словарь, ставящий каждому слову в соответствие число документов df, в которых оно встречается хотя бы один раз, и суммарную частоту cf по коллекции. Затем для конкретного документа строится его частотный словарь, и для каждого слова рассчитывается вес

$$w_t = TF \cdot \mathrm{RIDF}, \qquad TF = 0.5 + 0.5 \cdot \frac{tf}{tf_{\max}},$$

где tf — частота слова в данном документе, а $tf_{\max}$ — максимальная частота какого-либо слова в этом же документе [1].

Как и в случае TF · IDF, далее выбираются несколько слов (в оригинальной работе — шесть) с наибольшими значениями w_t, приводятся к канонической форме, упорядочиваются в алфавитном порядке и сцепляются в одну строку; над полученной строкой вычисляется контрольная сумма (например, CRC32), которая и используется как лексическая сигнатура документа [1].

2.2.3 Метод I-Match

Для решения задачи быстрого выявления дубликатов применяется сигнатурный подход, основанный на лексических принципах, предложенный A. Chowdhury. Основная идея метода I-Match состоит в вычислении дактилограммы для представления содержания документов на основе отобранного подмножества слов. Сначала для исходной коллекции документов строится словарь L, который включает слова со средними значениями IDF (исключаются очень частые служебные слова и очень редкие слова). Для каждого документа формируется множество U различных слов, входящих в него, и определяется пересечение U и словаря L. Список слов, входящих в пересечение, упорядочивается, и

для него вычисляется I-Match сигнатура (хеш-функция SHA1). Два документа считаются похожими, если у них совпадают I-Match сигнатуры [1].

2.2.4 Метод опорных слов

Метод опорных слов, предложенный С. Ильинским, применяется для выявления дубликатов на основе двоичного представления документа. Сначала из индекса по определенному правилу выбирается множество из N слов, называемых опорными. Затем каждый документ представляется N-мерным двоичным вектором, где i-я координата равна 1, если i-е опорное слово имеет в документе относительную частоту выше определенного порога, и равна 0 в противном случае. Этот двоичный вектор называется сигнатурой документа. Два документа похожи, если у них совпадают сигнатуры или совпадает большинство бит. Для каждого слова строится распределение документов по внутридокументной частоте. Проводится несколько итераций оптимизации, в которых максимизируется покрытие документов при фиксированной точности, а затем максимизируется точность при фиксированном покрытии [1].

2.2.5 Расстояние Левенштейна

Для решения задачи поиска близких вариантов фраз и обнаружения парафраза применяется редакционное расстояние. Расстояние Левенштейна — это минимальное количество однозначных операций редактирования (вставка, удаление, замена символа), необходимых для преобразования одной строки в другую. Метрика полезна для поиска близких вариаций текста, включая опечатки и синтаксические ошибки. Однако расстояние Левенштейна не учитывает порядок элементов — два текста с переставленными предложениями будут считаться различными. На больших текстах требует квадратичного времени вычисления $O(n \cdot m)$, что ограничивает его применимость в масштабных системах [7].

2.2.6 Сигнатуры на основе отдельных предложений документа

Исследование Зеленкова и Сегаловича включает алгоритмы, основанные на выборе характерных предложений документа. Эти методы решают задачу быстрого выявления потенциально похожих документов за счет анализа наибо-

лее информативных фрагментов текста.

Long Sent: выбираются 2 самых длинных предложения документа, сцепляются в алфавитном порядке, и вычисляется контрольная сумма CRC32. Алгоритм предполагает, что длинные предложения содержат больше информации и менее вероятно копируются с модификациями. Этот алгоритм показал наивысшую F-меру (0.82) среди всех исследованных методов.

Heavy Sent: вычисляется вес каждого предложения как сумма произведений TF $\cdot$ IDF для всех слов предложения. Выбираются 2 самых тяжелых (информативных) предложения, и для них вычисляется сигнатура. Метод не учитывает порядок предложений в документе [1].

3 Семантические методы и стилометрический анализ

Методы, рассматриваемые в данном разделе, решают задачу семантического сопоставления текстов: необходимо определить, являются ли два фрагмента текста смысловыми эквивалентами, содержат ли они парафраз, тематически близкую информацию или скрытое заимствование. В отличие от синтаксических и лексико-статистических подходов, семантические методы стремятся учитывать не только совпадение слов, но и их значение в контексте, что позволяет выявлять заимствования при глубокой переформулировке исходного текста.

3.1 Методы семантического сопоставления

3.1.1 Встраивания слов: Word2Vec, fastText, GloVe

Для решения задачи выявления парафраза и семантически эквивалентных текстов применяются методы семантического сопоставления на основе встраиваний слов (word embeddings). Встраивание слова — это отображение каждого слова в плотный вещественный вектор в пространстве низкой размерности, где геометрическая близость векторов соответствует семантической близости слов.

Основная идея метода состоит в следующем: вместо того чтобы сравнивать тексты по совпадению слов, каждый текст переводится в векторное представление, а затем сравниваются уже эти векторы. Тексты с близкими векторами (по косинусной мере близости) считаются семантически похожими, даже если в них используются различные, но синонимичные или близкие по смыслу слова [8].

Классические модели встраиваний слов включают:

— Word2Vec — обучает векторы слов с помощью задач Skip-gram или CBOW, предсказывая слово по контексту или контекст по слову и тем самым кодируя распределительную семантику слов;

— fastText — расширяет Word2Vec за счёт учёта символьных n-грамм, что позволяет лучше обрабатывать редкие и морфологически сложные слова;

— GloVe — строит векторы слов на основе матричной факторизации глобальной матрицы совместных встречаемостей слов в корпусе.

Для перехода от векторов отдельных слов к вектору всего документа используются различные способы построения представления текста:

— простое усреднение векторов всех слов документа;

— взвешенное усреднение с весами TF · IDF, подчеркивающее содержательные слова;

— специализированные модели представления предложений и документов (Doc2Vec, Sentence-BERT и др.), возвращающие вектор всего фрагмента текста [9, 11].

Далее два текста сравниваются как векторы в многомерном пространстве с помощью косинусной меры близости. Высокие значения косинусной меры близости свидетельствуют о семантическом сходстве документов и потенциальном наличии парафраза или смыслового плагиата.

Таким образом, методы на основе встраиваний слов решают задачу семантического сопоставления текстов: они переводят тексты в векторное пространство и позволяют находить скрытые заимствования за счет сравнения смысловых, а не только лексических характеристик [8, 9].

3.1.2 BERT и трансформерные модели

BERT (Bidirectional Encoder Representations from Transformers) — модель глубокого обучения, предварительно обученная на большом количестве текста, которая генерирует контекстные представления слов и предложений. В отличие от статических встраиваний, BERT учитывает контекст слова в предложении, что позволяет более точно захватывать смысл. Sentence-BERT (SBERT) расширяет BERT для создания встраиваний всех предложений, которые прямо оптимизированы для семантической схожести. Русскоязычные варианты BERT (например, RuBERT) обеспечивают качественное представление текстов на русском языке [9].

3.1.3 Siamese и Triplet Loss архитектуры

Данный класс методов решает задачу обучения метрического пространства, в котором расстояние между представлениями текстов отражает степень их семантического сходства. Цель состоит в том, чтобы вектора парафраз и корректных переформулировок располагались близко друг к другу, а несвязанные тексты — далеко.

Siamese архитектура состоит из двух или более копий одной и той же

нейронной сети, которые обрабатывают два входа и генерируют представления, сравниваемые для определения схожести. Triplet loss минимизирует расстояние между якорным примером и похожим примером, одновременно максимизируя расстояние между якорем и непохожим примером. Таким образом, модель явно обучается различать пары «заимствование / незаимствование».

В качестве базового энкодера в таких архитектурах могут использоваться как статические встраивания слов, так и контекстные модели (BiLSTM, трансформеры). В последнем случае порядок слов и структура предложения учитываются, что особенно важно для детекции сложного парафраза и стилевого заимствования [10].

3.1.4 LSTM и RNN с механизмом внимания

Recurrent Neural Networks (RNN), особенно в форме LSTM (Long Short-Term Memory) или GRU (Gated Recurrent Unit), решают задачу моделирования последовательности слов в тексте и захвата долгосрочных зависимостей между ними. Это позволяет выявлять случаи заимствования, когда сохраняется общая структура и логика изложения, но отдельные слова и фразы существенно изменены.

Такие сети могут обрабатывать текст пословно или по подсловам и по мере чтения «запоминать» важную информацию о ранее встреченных словах. Добавление attention-механизма позволяет модели сосредоточиться на наиболее релевантных частях входа при сравнении двух текстов, сопоставляя между собой фразы и предложения. BiLSTM (bidirectional LSTM) обрабатывает текст в обоих направлениях, улучшая представление.

Поскольку RNN-архитектуры явно работают с последовательностью, они учитывают порядок слов и предложений в документе, что критически важно для выявления сложных парафраз и стилистических заимствований [11].

3.2 Модели авторского стиля и стилометрия

3.2.1 Основы стилометрического анализа

Стилометрия — это область, изучающая характеристики письменного стиля, которые отличают одного автора от другого. Предполагается, что у каж-

дого автора есть уникальный стиль, который сохраняется даже при осознанной попытке его изменить. Признаки стилометрии включают:

— среднюю длину предложения, распределение длин предложений;

— часто используемые функциональные слова (предлоги, союзы, артикли);

— частотность части речи (POS tags);

— лексическое разнообразие (type-token ratio);

— использование пунктуации и заглавных букв;

— лексическую плотность;

— читаемость текста.

Стилометрический анализ применяется как для авторского сопоставления (определение авторства текста), так и для интринсивной детекции плагиата [12].

3.2.2 Интринсивная детекция плагиата

Интринсивная (intrinsic) детекция плагиата ищет признаки копирования внутри документа, без привлечения внешних источников. Основной метод решает задачу поиска стилистических разрывов: фрагменты заимствованного текста обычно имеют стиль отличающийся от основного стиля документа. Путем анализа последовательных блоков текста можно выявить участки, где стиль резко меняется. Применяются статистические тесты для определения значимости изменения стилистических параметров [12].

3.3 Анализ и верификация цитирования

Некорректное цитирование представляет собой особый вид неправильных заимствований, который включает не только прямое копирование текста без указания источника, но и более сложные формы академической недобросовестности: ошибки в оформлении библиографических ссылок, использование информации без должной атрибуции, искажение смысла первоисточника, отсылки к несуществующим или неверным источникам. Автоматизированная верификация цитирования позволяет выявить эти проблемы и повысить качество научных публикаций.

3.3.1 Проблемы и типы ошибок в цитировании

Ошибки в цитировании принимают различные формы и являются типичными видами некорректных заимствований:

— некорректный источник — ссылка приведена неправильно, источник не существует, или указывает на совершенно иное произведение;

— отсутствующая информация в источнике — утверждение, приписываемое источнику, в нем не содержится, или содержится в другом контексте;

— неверная страница — указан неправильный диапазон страниц, что затрудняет верификацию цитаты;

— отсутствует источник — информация используется без ссылки на источник;

— призрачные цитаты — источники цитируются по вторичным источникам без прямого обращения к первичному источнику.

Для решения задачи выявления этих ошибок используется автоматизированная верификация источников. Каждая проблема решается следующим образом:

— некорректные источники выявляются через сравнение извлеченных метаданных с библиографическими базами;

— отсутствующая информация проверяется семантическим сопоставлением контекста цитирования с содержанием источника;

— неверные страницы исправляются через анализ структуры документа и извлечение точных диапазонов;

— отсутствующие источники обнаруживаются через поиск неподтвержденных утверждений;

— призрачные цитаты выявляются через анализ цепочек цитирования [3, 13].

3.3.2 Автоматический парсинг и извлечение метаданных

Для проверки корректности источников необходимо автоматизированно извлекать метаданные из PDF и других документов. Системы парсинга, такие как CERMINE, GROBID и PDFDataExtractor, используют компьютерное зрение и обработку естественного языка для распознавания структуры документа и

извлечения текста, авторов, названия, года публикации, DOI (Digital Object Identifier), диапазонов страниц, и списка литературы. Извлеченные метаданные затем сопоставляются с библиографическими базами данных [13].

3.3.3 Проверка контекста цитирования и соответствия источнику

Анализ контекста цитирования включает проверку того, действительно ли утверждение в цитирующем тексте соответствует содержимому исходного документа. Для решения этой задачи применяются методы NLP:

— выделение синтаксических единиц вокруг ссылки (несколько предложений до и после цитирования);

— извлечение соответствующих фрагментов из исходного документа, соответствующих ключевым термам из цитирующего предложения;

— сравнение семантической схожести между контекстом ссылки и релевантными фрагментами источника с использованием косинусной меры близости;

— оценка соответствия на основе порога схожести для определения, является ли цитирование корректным [13, 14].

4 Экспериментальное исследование и оценка эффективности

4.1 Метрики оценки качества

В качестве основных показателей качества работы алгоритмов используются полнота, точность и F-мера. Для оценки эффективности систем обнаружения используются следующие метрики:

— точность — доля верно обнаруженных случаев плагиата из всех случаев, помеченных как плагиат. Высокая точность означает низкий процент ложных срабатываний.

— полнота — доля верно обнаруженных случаев из всех действительно существующих случаев плагиата. Высокая полнота означает, что система пропускает мало реальных случаев плагиата.

— F-мера — гармоническое среднее точности и полноты, дающее единую метрику качества работы алгоритма;

4.2 Учет порядка слов

Помимо метрик полноты и точности, важно учитывать, как методы ведут себя с точки зрения порядка слов и вычислительной сложности. Методы на основе предложений (Long Sent, Heavy Sent) в значительной степени зависят от синтаксической структуры и потому лучше выявляют парафраз в пределах предложений, но чувствительны к сильной переразметке текста. Лексические сигнатуры (TF · IDF, TF · RIDF, Opt Freq, I-Match) и шингловые методы (мегашинглы, Log Shingles) реализуют модель «мешка слов» или локальных шинглов и менее чувствительны к порядку предложений, но зачастую хуже различают семантический парафраз. С точки зрения вычислительных затрат методы на основе шинглов и лексических сигнатур проще и масштабируемее, тогда как нейросетевые семантические модели и стилометрический анализ требуют значительно больших ресурсов, но потенциально обеспечивают лучшую чувствительность к сложным формам плагиата.

4.3 Результаты сравнительного исследования

Сравнение проводилось на коллекции русскоязычных веб-документов РОМИП путем поиска заимствований в текстах научных статей. Тексты включали

как полные дубликаты, так и парафразы и частичные переиспользования. Система проверяла каждый алгоритм на единообразных наборах тестовых пар документов. Тексты включали как полные дубликаты, так и парафразы и частичные переиспользования. Коллекция содержала 1250 научных статей из различных областей (информатика, математика, физика) общим объемом более 5000 страниц. Система проверяла каждый алгоритм на единообразных наборах тестовых пар документов, включая пары с известным заимствованием и пары без заимствований для оценки ложных срабатываний.

Алгоритм	**Полнота**	**Точность**	F**-мера**
Long Sent	0.84	0.80	0.82
TF	0.60	0.94	0.73
Opt Freq	0.59	0.94	0.73
TF · RIDF	0.59	0.95	0.73
Heavy Sent	0.62	0.86	0.72
TF · IDF	0.54	0.96	0.69
Lex Rand	0.50	0.97	0.66
Descr Words	0.44	0.77	0.56
Log Shingles	0.39	0.97	0.56
Megashingles	0.36	0.91	0.51
MD5	0.23	1.00	0.38

Таблица 4.1 — Сравнение метрик качества алгоритмов обнаружения заимствований в порядке убывания рейтинга

Результаты показывают, что алгоритм выбора длинных предложений (Long Sent) показал наилучшие результаты по F-мере (0.82), сочетая высокую полноту (0.84) и точность (0.80). Успех метода объясняется тем, что длинные предложения содержат больше уникальной информации и реже подвергаются существенным модификациям при парафразировании, что делает их надежными индикаторами заимствования.

Лексические методы (TF, TF · IDF, TF · RIDF, Opt Freq) демонстрируют высокую точность (0.94–0.97), но умеренную полноту (0.54–0.60). Высокая точность обусловлена тем, что эти методы выявляют документы с существенным совпадением ключевых слов, что минимизирует ложные срабатывания. Однако низкая полнота указывает на чувствительность к парафразированию: при замене слов синонимами или изменении формулировок методы пропускают за-

имствования.

Алгоритм MD5 показывает идеальную точность (1.00), но крайне низкую полноту (0.23), что объясняется его способностью обнаруживать только точные побайтовые копии документов. Любое минимальное изменение текста приводит к полному изменению хеш-суммы.

Метод мегашинглов уступает более простым алгоритмам (F-мера 0.51), что объясняется строгими требованиями к совпадению супершинглов: для признания документов похожими необходимо совпадение хотя бы одного мегашингла из 15, что делает метод чувствительным к локальным модификациям текста.

Методы, основанные на шинглах и мегашинглах, хорошо подходят для масштабных систем, так как требуют минимальной памяти и времени, но их точность и полнота ниже, чем у методов на основе предложений [1].

4.4 Ограничения исследования

Проведенное экспериментальное исследование имеет ряд ограничений, которые необходимо учитывать при интерпретации полученных результатов:

— эксперимент проведен только на русскоязычных текстах научных статей, что ограничивает обобщаемость результатов на тексты других языков и стилей;

— коллекция ограничена научными статьями из областей информатики, математики и физики; методы могут демонстрировать иную эффективность на текстах других жанров (художественная литература, публицистика, техническая документация);

— не тестировались кросс-языковые заимствования и переводной плагиат, которые требуют специализированных подходов;

— размер коллекции (1250 документов) может быть недостаточен для полной оценки масштабируемости алгоритмов на больших массивах данных;

— не проводилось тестирование современных нейросетевых методов (BERT, Siamese networks) на той же коллекции, что ограничивает возможность прямого сравнения с классическими подходами.

ЗАКЛЮЧЕНИЕ

Проблема обнаружения некорректных заимствований требует применения интегрированного подхода, сочетающего методы из различных классов:

— высокоскоростные методы обнаружения (отпечатки, LSH, шинглы) для быстрого отсеивания явно неподозрительных документов;

— статистические и лексико-семантические методы (TF · IDF, опорные слова, расстояния, N-граммы) для детального сравнения;

— современные нейросетевые методы (BERT, Siamese networks, LSTM) для выявления сложных парафраз и переводов;

— стилометрический анализ для выявления внутридокументных разрывов стиля;

— автоматизированные системы верификации источников и метаданных.

Экспериментальные исследования показывают, что ни один единственный метод не может быть универсален. Комбинация различных подходов, правильно настроенных и взвешенных в зависимости от контекста, обеспечивает высокую точность и полноту при обнаружении различных типов нарушений академической этики.

Наилучшие результаты по F-мере (0.82) показывает алгоритм Long Sent, основанный на выборе длинных предложений. Лексические методы обеспечивают высокую точность (до 0.97), но умеренную полноту. Методы на основе шинглов и мегашинглов эффективны для масштабных систем, но требуют тщательной настройки параметров. Дальнейшее развитие в этой области идет в направлении использования более мощных языковых моделей, лучшей интеграции методов верификации ссылок, разработки более эффективных алгоритмов кросс-языкового поиска плагиата, и создания специализированных систем для различных доменов и типов документов [1].

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Зеленков Ю.Г., Сегалович И.В. Сравнительное исследование методов определения нечетких дубликатов для Web-документов // Труды 9-й Всероссийской научной конференции RCDL2007. — Переславль-Залесский, 2007.

2. Гришин В.Д. Методы анализа и поиска заимствований в тексте // Актуальные проблемы гуманитарных и естественных наук. — 2018. — № 6.

3. Авдеева Н.В., Ледовская В.М. Некорректные заимствования в диссертациях: способы их обнаружения // Высшее образование в России. — 2015. — № 6. — С. 22–29.

4. Implementation of Winnowing Algorithm Based K-Gram to Identify Plagiarism // MATEC Web of Conferences. — 2018. — Vol. 154.

5. A Robust Document Identification Framework through f-BP Fingerprint // Applied Sciences. — 2021. — Vol. 7, No. 8.

6. Research on Text Similarity Measurement Hybrid Algorithm with TF-IDF Method // Applied Mathematics. — 2022.

7. Levenshtein Distance, Sequence Comparison and Biological Applications // MIT Open Courseware. — 2020.

8. On the Sentence Embeddings from BERT for Semantic Textual Similarity // Proceedings of EMNLP. — 2020.

9. Phrase-BERT: Improved Phrase Embeddings from BERT with an Application to Corpus Exploration // Proceedings of EMNLP. — 2021.

10. Hybrid approach of BERT extraction with deep Siamese Bi-LSTM for semantic text similarity // Scientific Reports. — 2022.

11. An LSTM-based Plagiarism Detection via Attention Mechanism and a Population-Based Approach for Pre-training Parameters with Imbalanced Dataset // arXiv preprint. — 2021.

12. Intrinsic Plagiarism Detection // Proceedings of the ACL Workshop. — 2015.

13. AutoIE: Automated Information Extraction from Scientific Literature // arXiv preprint. — 2024.

14. CiteCheck: Accurate Citation Faithfulness Detection via Semantic Graph Representation Learning // arXiv preprint. — 2025.