УДК 004.912:004.8

АЛГОРИТМЫ СРАВНЕНИЯ ТЕКСТОВОЙ ИНФОРМАЦИИ

К.С. Крез
Ассистент и аспирант кафедры проектирования информационно-компьютерных систем БГУИР
k.krez@bsuir.by

Е.Н. Шнейдеров
Доцент кафедры проектирования информационно-компьютерных систем БГУИР, кандидат технических наук, доцент, проректор по учебной работе
alexvikt.minsk@gmail.com

В.И. Голушко
Студент специальности информационные системы и технологии (в бизнес-менеджменте) кафедры проектирования информационно-компьютерных систем БГУИР
vadimgolushko2004@gmail.com

К.С. Крез

Окончила Белорусский государственный университет информатики и радиоэлектроники. Область научных интересов: корреляционный анализ цифровых следов пользователей, нейронные сети, разработка и анализ структуры хранения данных.

Е.Н. Шнейдеров

Окончил Белорусский государственный университет информатики и радиоэлектроники. Область научных интересов связана с разработкой феноменологических моделей и научно-методических основ компьютерного проектирования радиоэлектронных средств, технического обеспечения безопасности и создания электронных систем безопасности.

В.И. Голушко

Является студентом Белорусского государственного университета информатики и радиоэлектроники по специальности «Информационные системы и технологии (в бизнес-менеджменте)», дневная форма обучения. Профессиональные интересы связаны с разработкой программного обеспечения и изучением различных направлений в сфере информационных технологий.

Аннотация. В статье рассматриваются алгоритмы сравнения текстовой информации, применяемые в задачах обработки естественного языка. Выполнен обзор подходов, ориентированных на анализ лексического, структурного и семантического сходства текстов, включая статистические методы, расстояние Левенштейна, алгоритмы локально-чувствительного хеширования и фонетические подходы. В статье рассмотрены архитектуры, принципы работы и области применения моделей Word2Vec, GloVe, FastText, BERT и Doc2Vec. На основании проведённого анализа установлено, что выбор оптимального алгоритма зависит от требований к точности, вычислительной эффективности и специфики обрабатываемых данных. Особо отмечается высокая эффективность модели BERT в задачах сравнения текстов.

Ключевые слова: алгоритмы сравнения текстов; BERT; TF-IDF; семантический анализ

Введение. Определение степени сходства между текстами представляет собой одну из ключевых задач в области автоматической обработки естественного языка. Эта задача возникает в различных контекстах, таких как подготовка рефератов, дипломных работ, отчетов, машинный перевод, установление авторства, выявление академического плагиата, и других, где требуется оценить уровень подобия между двумя текстовыми документами. В подобных сценариях требуется определение меры близости между двумя текстовыми документами на основе их лексических, структурных и семантических характеристик. Цель

статьи заключается в проведении обзора существующих алгоритмов сравнения текстовой информации и анализе принципов их работы. При анализе текстового сходства в современных алгоритмах обычно учитываются следующие группы признаков:

– формат файла текстового процессора Microsoft Word DOCX;

– взаимное расположение слов и устойчивых сочетаний, включая порядок следования и совместную встречаемость [1];

– синтаксические и семантические связи, существующие между словами [2];

– предметная область и тематическая направленность текстов [3].

Основная часть. Для сравнения текстов применяются различные методы, отличающиеся уровнем анализа и типом используемых признаков. К наиболее распространённым из них относятся модели векторного пространства, включая TF-IDF и методы распределённых представлений; редакционные расстояния, оценивающие количество операций, необходимых для преобразования одной последовательности в другую; методы локально-чувствительного хеширования, обеспечивающие эффективное сопоставление текстов с учётом их структурных характеристик; а также алгоритм Soundex, основанный на сопоставлении текстовых единиц по фонетическому сходству.

Особое распространение получили алгоритмы, использующие векторные представления слов, предложений и текстов. Их основная идея заключается в отображении текста в пространство признаков, в котором степень сходства между объектами определяется на основе расстояния или меры близости между соответствующими векторами [4]. При данном подходе компоненты векторного представления фиксируют существенные характеристики текста, включая частотные показатели употребления терминов: каждому слову ставится в соответствие отдельное измерение, а сами документы описываются как векторы в многомерном признаковом пространстве. При этом близкое расположение точек в векторном пространстве интерпретируется как свидетельство высокой степени сходства между текстами. Существенным различием между методами данного класса является выбор метрики, определяющей способ вычисления расстояния или близости между векторами.

Одним из ключевых подходов в рамках векторного представления текста является назначение каждому слову определённого веса, отражающего его значимость в документе и во всём тексте.

При данном подходе расчёт метрики опирается на предварительное определение информационного веса каждого слова в тексте. Наиболее известным методом, реализующим такой принцип, является Term Frequency – Inverse Document Frequency (далее – TF-IDF). В его основе лежат два показателя: Term frequency (далее – TF), отражающий частоту появления термина в конкретном тексте, и Inverse document frequency (далее – IDF), характеризующий обратную частоту его встречаемости в коллекции документов. Для более точного расчета весов значения TF и IDF перемножаются, что называется взвешиванием TF-IDF. Логика метода TF-IDF основана на том, что тексты, в которых определенное слово встречается с близкой частотой, считаются схожими [5]. Однако TF-IDF не учитывает контекст употребления слов [6]. Векторное представление текстов строится на основе гипотезы «мешка слов». Мера TF-IDF вычисляется как произведение частоты слова в тексте и обратной частоты слова во всём наборе документов и определяется по формуле 1.

Формула для определения показателя имеет следующий вид:

$$TF\_IDF = TF \cdot IDF, \tag{1}$$

где TF – частота слова в конкретной категории / документе / коллекции (в зависимости от того, какие данные анализируются), IDF – обратная частота документа (популярность слова).

Частота слова в категории определяется по формуле 2:

$$TF = \frac{n_t}{\sum_{i=1}^{k} n_i}, \tag{2}$$

где n_t – количество отдельных слов в категории / документе / коллекции, $\sum_{i=1}^{k} n_i$ – общее количество всех слов в категории / документе / коллекции.

Обратная частота документа (также часто называют инверсией частоты) определяется по формуле 3:

$$IDF = \ln \frac{n_c}{\sum_{j=1}^{m} n_j}, \tag{3}$$

где n_c – количество категорий / документов / коллекций всего, $\sum_{j=1}^{m} n_j$ – количество категорий / документов / коллекций, в которых содержится интересующее слово.

TF-IDF не учитывает семантическую близость текстов в различных документах. Иными словами, при совпадении смыслового содержания, выраженного различными лексическими средствами, данный подход демонстрирует эффективность преимущественно на уровне лексико-статистических совпадений. Вместе с тем в задачах попарного сравнения документов, где существенное значение имеют точные формулировки и прямые текстовые совпадения, указанная особенность не всегда должна рассматриваться как недостаток метода. Дополнительно следует отметить, что TF-IDF не позволяет локализовать конкретные изменения и различия между сравниваемыми документами, а формирует лишь обобщённую количественную оценку степени их сходства.

Помимо TF-IDF в задачах анализа текстовых данных находят применение методы векторного представления, ориентированные на более полное моделирование семантических характеристик текста. К числу таких методов относятся Word2Vec, GloVe, FastText, BERT, Doc2Vec и Sense2Vec.

Word2Vec – модель распределённого представления слов, разработанная Google, предназначенная для построения векторных эмбеддингов слов. Получаемые векторы отражают семантические и контекстные связи между лексическими единицами, вследствие чего слова с близким значением имеют близкое расположение в векторном пространстве.

Global Vectors for Word Representation (далее – GloVe) представляет собой модель построения векторных представлений слов, основанную на анализе глобальной статистики их совместной встречаемости во фрагментах текстов. В отличие от Word2Vec, которая в большей степени ориентирована на использование локального контекста, модель GloVe объединяет преимущества статистического подхода и распределённого представления слов.

FastText является модификацией модели Word2Vec, разработанной компанией Facebook, в которой слово рассматривается как совокупность символьных n-грамм. Такой подход позволяет учитывать морфологические особенности слов и формировать векторные представления не только для слов, присутствующих в обучающем фрагменте, но и для ранее не встречавшихся лексических единиц.

Bidirectional Encoder Representations from Transformers (далее – BERT) представляет собой контекстно-зависимую языковую модель, основанную на архитектуре Transformer и разработанную компанией Google. За счёт двунаправленного механизма обучения модель способна учитывать как левый, так и правый контекст слова, что обеспечивает более полное выявление смысловых связей в тексте и высокую результативность при решении широкого круга задач обработки естественного языка.

Doc2Vec – расширение Word2Vec, предназначенное для построения векторных представлений документов, абзацев и предложений. Использование данного подхода позволяет выполнять сравнение текстовых фрагментов на уровне семантической близости, а не только по совпадению отдельных слов.

Sense2Vec – модель векторного представления, ориентированная на учёт многозначности слов. В рамках данного подхода для различных значений одной и той же лексемы формируются отдельные векторы на основе контекста её употребления, что способствует более точному моделированию семантики.

В обзоре [7] рассмотрены применения векторных представлений текстов.

Современное развитие данного подхода связано с несколькими направлениями совершенствования моделей представления текста.

Во-первых, осуществляется расширение векторных представлений слов путём интеграции дополнительных типов лингвистической информации, включая именованные сущности, морфосинтаксические характеристики и языковые модели.

Во-вторых, ведётся разработка методов, учитывающих многозначность слов и словосочетаний, что повышает точность семантической интерпретации.

В-третьих, наблюдается переход от дискретных векторных представлений к непрерывным вероятностным моделям, описываемым функциями распределения плотности. Наряду с этим разрабатываются специализированные представления, адаптированные к отдельным предметным областям, а также многоязычные модели, обеспечивающие обработку текстов на нескольких языках в рамках единого признакового пространства.

Метод редакционного расстояния, наиболее известной реализацией которого является расстояние Левенштейна, представляет собой способ количественной оценки различий между двумя символьными последовательностями. Данный метод основан на вычислении минимального числа элементарных преобразований, необходимых для преобразования одной последовательности в другую. В качестве таких преобразований рассматриваются вставка, удаление и замена символов.

К элементарным операциям в рамках данного подхода относятся:

– вставка/удаление символа;

– замена одного символа на другой.

Алгоритм производит подсчет операций, необходимых для преобразования одного набора символов (текста, документа) в другой. Данный метод назван в честь советского математика Владимира Левенштейна, который рассматривал это расстояние в 1965 году [8].

Расстояние Левенштейна применяется в следующих областях:

– сравнение текстовых данных – выявление различий между файлами и документами;

– коррекция ошибок – исправление опечаток в поисковых запросах, базах данных, системах ввода текста, а также при OCR и обработке речи;

– биоинформатика – анализ схожести генетических последовательностей (генов, хромосом, белков) [9].

К числу недостатков расстояния Левенштейна относится высокая чувствительность к перестановке слов, даже если она не изменяет смысл текста, а также зависимость значения метрики от длины сравниваемых единиц. В результате короткие, но несвязанные слова могут демонстрировать малое редакционное расстояние, тогда как длинные и частично совпадающие слова – большее, что снижает точность метода при оценке смысловой близости.

Локально-чувствительное хеширование (далее – TLSH) – уникальный набор символов, рассчитанный по определенному алгоритму и соответствующий определенному набору данных. Существует ряд алгоритмов (хеш-функций), используемых для расчета хеш-суммы: CRC32, SHA256, MD5. В случае, если файл подвергается изменению,

изменяется и хеш-сумма этого файла. Для полностью идентичных файлов хеш-сумма будет равной.

Для сравнения документов большого объёма, имеющих большое количество разделов, может применяться локально-чувствительное хеширование. Алгоритмы локально-чувствительного хеширования помещают схожие данные в одну корзину с высокой вероятностью [10]. Вместе с тем эффективность таких методов зависит от способа разбиения текста на признаки и выбора параметров хеширования поэтому два раздела могут быть определены как разные, если в одном из них есть пробел, а в другом его нет, даже если оба файла содержат одинаковый текст.

SoundEx – алгоритм, который позволяет сравнивать набор слов по их звучанию [11]. Изначально алгоритм был ориентирован на американский вариант английского языка, однако впоследствии появились его модификации для других языков, включая русский.

Иногда этот алгоритм не способен обнаружить сходство между очень близкими фамилиями: например, «Levinson» получит код L152, а «Lewinson» – код L525. Кроме того, Soundex плохо работает в ситуациях, когда произношение сильно расходится с написанием, что в английском языке бывает нередко. По этой причине его прямое применение к русскоязычным текстам сопровождается значительными погрешностями и обычно требует предварительной адаптации или замены более подходящими фонетическими алгоритмами.

1 Тип данных – категория текстовых данных, для обработки которых предназначен алгоритм. Выделяются следующие типы:

– короткие строки – текстовые последовательности малой длины, для которых важно точное посимвольное сравнение; такие методы эффективны на коротких данных, но менее пригодны для длинных текстов;

– тексты произвольной длины – слова, предложения и абзацы, обрабатываемые на основе токенов или n-грамм; данные методы универсальны по длине текста, однако ограниченно учитывают семантику;

– документы – объёмные структурированные тексты, анализ которых требует учёта не только лексических совпадений, но и смыслового содержания;

– слова и контексты – отдельные слова или их окружение в тексте, анализируемые на основе статистики совместной встречаемости;

– слова (онтологии) – слова, связанные через иерархии понятий (например, WordNet);

– длинные тексты – тексты из множества предложений, где сравнение требует агрегации результатов на уровне абзацев или предложений (а не символов / слов).

2 Учет контекста – свойство алгоритма учитывать окружение слова или фрагмента, но не обязательно его смысловую связь.

3 Чувствительность к опечаткам – степень, в которой алгоритм реагирует на ошибки в написании слов (опечатки, перестановки символов, лишние / пропущенные буквы). Чувствительность к опечаткам делится на:

– высокая (алгоритм реагирует на малейшие изменения символов);

– умеренная (частично игнорирует перестановки или замены);

– низкая (ориентирован на семантику, а не на символы).

4 Семантическое понимание – способность алгоритма учитывать сходство значений, а не только поверхностное совпадение символов или слов.

5 Внешние ресурсы – дополнительные данные или инструменты, требуемые для работы алгоритма.

Корпус – большой структурированный набор текстов, используемый для статистического анализа языка (частотность слов, контексты, семантика). WordNet – лексическая база данных английского языка, где слова связаны в иерархии (синонимы, гиперонимы, гипонимы).

Таблица 1. Сравнение алгоритмов

Алгоритм	Тип данных	Учет контекста	Чувствительность к опечаткам	Семантическое понимание	Внешние ресурсы	Примеры использования
Расстояние Левенштейна	Короткие строки	Нет	Высокая	Нет	Нет	Проверка орфографии (MS Word), системы исправления ошибок в базах данных
Jaro-Winkler	Короткие строки	Нет	Умеренная	Нет	Нет	Сопоставление записей в CRM-системах (Salesforce), идентификация пользователей.
N-граммы	Текст любой длины	Частично	Зависит от n	Нет	Нет	Поиск плагиата (Turnitin), кластеризация коротких текстов
LSA	Документы	Да	Низкая	Да	Требует	Тематическое моделирование (Gensim), рекомендательные системы (Amazon)
ESA	Высокая	Да	Низкая	Да	Корпус	Семантический поиск (IBM Watson), анализ научных статей
PMI	Слова/ контексты	Да	Низкая	Да	Требует	Определение коллокаций (Google N-gram Viewer), анализ тональности отзывов
Wu-Palmer	Слова (онтологии)	Нет	Высокая	Да	Корпус	Оценка эссе (ETS), классификация медицинских терминов (UMLS)
Lin	Слова (онтологии)	Нет	Средняя	Да	WordNet	Дизъюнкция слов в NLP-библиотеках (NLTK), анализ юридических документов
Resnik	Слова (онтологии)	Нет	Низкая	Да	WordNet + корпус	Семантическая кластеризация (Apache OpenNLP), поиск синонимов
SoftTFIDF	Текст	Частично	Умеренная	Нет	Нет (или корпус для TF-IDF)	Обработка заявок в банках (сопоставление клиентских данных), очистка дубликатов в CRM
Monge-Elkan	Длинные тексты	Нет	Низкая	Нет	Нет	Сравнение резюме с вакансиями (LinkedIn), анализ текстовых шаблонов
BERT	Текст (любой длины)	Да	Низкая	Да	Предобучен	Поисковые системы (Google), чат-боты (ChatGPT), анализ тональности в соцсетях (Twitter)

Заключение. Проведённый анализ показал, что алгоритмы сравнения текстовой информации существенно различаются по принципам работы, уровню представления текста и области практического применения. Такие алгоритмы как расстояние Левенштейна, SoundEx и методы локально-чувствительного хеширования, ориентированы преимущественно на анализ формальных характеристик текста и позволяют эффективно выявлять совпадения на уровне символов, слов или структурных элементов документа. Однако данные методы обладают ограниченной способностью учитывать семантические связи между словами и устойчивость к изменениям формулировок.

Список литературы

[1] Errecalde M., Ingaramo D., Rosso P. A new AntTree-based Algorithm for Clustering Short-text Corpora // Journal of Comput-er Science and Technology. – 2010. – V. 10. – № 1. – P. 1–7.

[2] Hayes R., Pisano G., Wheelwright S. Operations, Strategy, and Technical Knowledge. Hoboken, NJ: Wiley, 2007. Ferrrandez и др. Deep vs. Shallow Semantic Analysis Applied to Textual Entailment Recognition // Advances in Natural Language Pro-cessing 5th International Conference on NLP, FinTAL 2006 Turku, Finland, August 23-25, 2006 Proceedings

[3] Yue Wang, Hongsong Li, Haixun Wang, Kenny Q. Zhu. Toward Topic Search on the Web // JMIR Publications, 2012, P. 28. 20. Zampieri M., A

[4] Joaquin Perez-Iglesias – Integrating the Probabilistic Model BM25: BM25F into Lucene // Cornell University free distribu-tion platform[arXiv.org], 2009, P. 7.

[5] Christopher D. Manning, Prabhakar Raghavan & Hinrich Schütze. An Introduction to Information Retrieval // Cambridge Uni-versity Press, 2009, P. 19.

[6] Automatic Keyword Extraction from Individual Documents [Электронный ресурс]. – Режим доступа: https://www.researchgate.net/publication/227988510Automatic_Keyword_Extraction-dfrom_Individual_Docments.

[7] Харламов А. А., Гордеев Д. И. Дистрибутивная VS сетевая семантика в диалоговых системах // Проблемы искусственного интеллекта, 2019, №2 (13), С. 93. 6. Ч

[8] Левенштейн В. И. Двоичные коды с исправлением выпадений, вставок и замещений символов // Докл. АН СССР, 1965, Т. 163, №4, С. 845.

[9] Shama Rani, Jaiteg Singh – Enhancing Levenshtein's Edit Distance Algorithm for Evaluating Document Similarity // Communi-cations in Computer and Information Science, 2017, P. 75.

[10] Loïc Paulevé, Hervé Jégou, Laurent Amsaleg – Locality sensitive hashing: A comparison of hash function types and querying mechanisms // HAL, 2011, P. 11.

[11] Wagner R.A., Fischer M.J. The string-to-string correction problem / Journal of the ACM, 1974. Vol. 21, № 1, pp. 168–173.

[12] Zeeshan Bhatti, Ahmad Waqas, Imdad Ali Ismaili, Dil Nawaz Hakro, Waseem Javaid Soomr // Phonetic based SoundEx & ShapeEx algorithm for Sindhi Spell Checker System, 2014, P. 7.

Авторский вклад

Крез Карина Сергеевна – постановка цели и задач исследования, проведение литературного обзора, анализ и систематизация алгоритмов сравнения текстовой информации, написание черновика рукописи, подготовка сравнительной таблицы, общая координация проекта.

Евгений Николаевич Шнейдеров – научное руководство, проверка методологии, пересмотр и редактирование содержания статьи, утверждение окончательного варианта рукописи для публикации.

Вадим Иванович Голушко – сбор и обработка данных, анализ алгоритмических подходов, участие в написании отдельных разделов рукописи, оформление текста и списка литературы.

ALGORITHMS FOR COMPARING TEXT INFORMATION

K.C. Krez
Assistant and graduate student in the Department of Information and Computer Systems Design at BSUIR

E.N. Shneiderov
Associate Professor of the Department of Design of Information and Computer Systems of BSU-IR, Candidate of Technical Sciences, Associate Professor, Vice-Rector for Academic Affairs

V.I. Golushko
Student of the information systems and technologies (in business management) specialty of the Department of Information and Computer Systems Design at BSUIR

Abstract. This article examines algorithms for comparing textual information used in natural language processing tasks. It provides an overview of approaches focused on analyzing the lexical, structural, and semantic similarity of texts, including statistical methods, Levenshtein distance, locality-sensitive hashing algorithms, and phonetic approaches. The article discusses the architectures, operating principles, and application areas of the Word2Vec, GloVe, FastText, BERT, and Doc2Vec models. Based on the conducted analysis, it is established that the selection of the optimal algorithm depends on requirements regarding accuracy, computational efficiency, and the specific characteristics of the data being processed. Particular emphasis is placed on the high effectiveness of the BERT model in text comparison tasks.

Keywords text comparison algorithms; BERT; TF-IDF; semantic analysis.