АҚПАРАТТЫҚ-КОММУНИКАЦИЯЛЫҚ ТЕХНОЛОГИЯЛАР
ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ
INFORMATION AND COMMUNICATION TECHNOLOGIES

АҚПАРАТТЫҚ-КОММУНИКАЦИЯЛЫҚ ТЕХНОЛОГИЯЛАР
ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ
INFORMATION AND COMMUNICATION TECHNOLOGIES

DOI 10.51885/1561-4212_2025_2_123
МРНТИ 20.53.19 01

Г. Жомартқызы[1], К.А. Павлов[1], М.Ж. Базарова[2]
[1]Восточно Казахстанский технический университет им. Д. Серикбаева,
г. Усть-Каменогорск, Казахстан
*E-mail: gzhomartkyzy@edu.ektu.kz**
E-mail: pavlov.kanstantin@gmail.com
[2]Восточно Казахстанский университет им. С. Аманжолова, г. Усть-Каменогорск, Казахстан
E-mail: madina9959843@gmail.com

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ОЦЕНКИ СХОДИМОСТИ СТРУКТУР ДАННЫХ С ИСПОЛЬЗОВАНИЕМ МАТЕМАТИЧЕСКИХ МНОЖЕСТВ И АЛГОРИТМА ЛЕВЕНШТЕЙНА

МАТЕМАТИКАЛЫҚ ЖИЫНДЫ ЖӘНЕ ЛЕВЕНШТЕЙН АЛГОРИТМІ НЕГІЗІНДЕ ДЕРЕКТЕР ҚҰРЫЛЫМДАРЫНЫҢ ЖИНАҚТЫЛЫҒЫН БАҒАЛАЙТЫН САЛЫСТЫРМАЛЫ ТАЛДАУ

COMPARATIVE ANALYSIS OF THE EVALUATION OF THE CONVERGENCE OF DATA STRUCTURES USING MATHEMATICAL SETS AND THE LEVENSHTEIN ALGORITHM

Аннотация. *В данной работе описано сравнение двух методов оценки сходимости различных наборов данных: математических множеств и алгоритма Левенштейна. Приведено описание общей методологии вычисления сходимости данных: анализ первичных данных, очистка данных, построение множества данных, применение алгоритма Левенштейна и расчет сходимости. Приведены этапы оценки сходимости данных с общей структурой. Цель оценки сходимости данных с общей структурой состоит в том, чтобы определить, насколько хорошо модель или теория объясняет наблюдаемые данные. Для улучшения показателей сходимости данных был выбран алгоритм Левенштейна из набора алгоритмов на основе семантического поиска. Во второй части статьи проводится сравнительный анализ этих двух методов на примере нескольких наборов данных.*

Перед выполнением вычислений сходимости данных была проведена предварительная очистка с использованием стандартных методов интеллектуального анализа данных (data mining), реализованных на языке программирования Python. Также была выполнена нормализация полей в структуре данных, что было особенно важно при работе с реляционными базами данных для обеспечения корректной интерпретации и сопоставимости результатов. Приведены результаты сравнительного анализа представленых методов и дан ряд рекомендаций по выбору наиболее эффективного метода оценки сходимости структур данных в зависимости от конкретных требований и условий.

Ключевые слова: *репликация, сходимость данных, ERP системы, алгоритм Левенштейна, множества, базы данных.*

Аңдатпа. *Бұл жұмыста әртүрлі деректер жиынтығының жинақтылығын бағалайтын екі әдістің салыстырмасы сипатталған: математикалық жиынтық және Левенштейн алгоритмі. Мақаланың бірінші бөлімінде мәліметтер жиынтығын талдау, тазалау және құру кезеңдерінен тұратын мәліметтер құрылымдарының жинақтылығын бағалайтын жалпы әдіснамасы сипатталады. Ортақ құрылымды деректер жинақтылығын бағалау кезеңдері берілген. Ортақ құрылымды деректер жинақтылығын*

бағалаудың мақсаты – модель немесе теория бақыланатын деректерді қаншалықты дұрыс бейнелейтінін анықтау. Деректер жинақтылығын бойынша көрсеткіштерді жақсарту үшін семантикалық іздеуге негізделген алгоритмдер жиынтығынан Левенштейн алгоритмі таңдалды. Мақаланың екінші бөлігінде бірнеше деректер жиынтығының мысалымен жүзеге асырылатын осы екі әдістің салыстырмалы талдауы келтіріледі. Деректер жинақтылығын есептеуді орындамас бұрын, Python бағдарламалау тілінде енгізілген стандартты деректерді интеллектуалды талдау әдістерін қолдану арқылы алдын ала тазалау жүргізілді. Деректер құрылымындағы өрістерді нормалау да орындалды, бұл нәтижелердің дұрыс интерпретациялануын және салыстырылуын қамтамасыз ету үшін реляциялық мәліметтер базасымен жұмыс істегенде маңызды болды. Ұсынылған әдістердің салыстырмалы талдау нәтижелері көрсетіледі және және сонымен қатар нақты талаптар мен шарттарға байланысты деректер құрылымдарының жинақтылығын бағалаудың ең тиімді әдісін таңдау бойынша бірқатар ұсыныстар беріледі,

Түйін сөздер: *репликация, деректерді жинақтылығы, ERP жүйелер, Левенштейн алгоритмі, жиынтықтар, мәліметтер базасы.*

Abstract. *This paper describes a comparison of two methods for evaluating the convergence of different data sets: mathematical sets and the Levenshtein algorithm. The first part of the paper describes a general methodology for evaluating the convergence of data structures, including the steps of analyzing, cleaning, and constructing data sets. The stages of assessing data convergence with a common structure are presented. The goal of assessing data convergence with a common structure is to determine how well a model or theory explains the observed data. To improve data convergence metrics, the Levenshtein algorithm was selected from a set of algorithms based on semantic search. In the second part of the paper, a comparative analysis of the two methods is performed using several data sets as examples. Before performing data convergence calculations, preliminary data cleaning was carried out using standard data mining techniques implemented in the Python programming language. In addition, normalization of fields within the data structure was performed, which was particularly important when working with relational databases to ensure accurate interpretation and comparability of the results. The results of the comparative analysis of the presented methods are given and a number of recommendations are given for choosing the most effective method for evaluating the convergence of data structures depending on specific requirements and conditions.*

Keywords: *replication, data convergence, ERP systems, Levenshtein algorithm, sets, databases.*

Постановка проблемы. Метрика сходимости структур данных (индекс разнообразия) в реляционных БД имеет узконаправленный характер и в первую очередь показывает, насколько эффективно и точно могут быть созданы представления на основе типов данных полей в таблицах (Рыбанов, 2019), однако данная метрика не учитывает их семантический смысл (Hedge, 1993). Проблема создания общей структуры данных на основе уже имеющихся наборов, которые прошли все этапы предобработки и нормализации, встает, например, в области репликации, где данные загружаются из различных источников. Конкретным примером является концепция ETL (Pishghadam, 2011), которая описывает этапы работы с информацией для успешной репликации между различными информационными системами или источниками информации. В ней на этапе извлечения информации встает проблема создания эффективной структуры для хранения собранной информации, и на данный момент такая структура является унифицированной (Piattini et al., 2001).

Основой для таких хранилищ, использующих ETL-подход, служат хранилища данных Hadoop, Apache Hive, NoSQL базы данных и различного рода реляционные базы данных. Поэтому в большинстве случаев при возникновении такого рода задач специалистам приходится создавать пользовательские структуры данных, со своим набором полей, такие как таблицы и представления в реляционных базах данных, кластеры данных Hadoop и различного рода облачные хранилища.

Рассмотренный подход к репликации данных устанавливает общие положения и наборы операций, необходимых для осуществления процесса репликации в целом. Программный подход к репликации, в частности репликация с помощью API middleware (Lin et al., 2005), используется для улучшения производительности загрузки данных, их чтения и записи в целевую систему, вследствие чего имеет более организованную первоначальную структуру данных, однако и такой подход не лишен недостатков.

Например, в случае с Master-slave репликацией (ведущий-ведомый) (Cecchet et al., 2008), относящейся к API middleware подходу, доступ к информации возможен только для чтения на ведомых узлах, а обновления передаются на ведущий. В этом случае может возникать неконсистентность данных, влияющих на всю структуру получаемых данных.

Методы исследования. В работе предлагается общая модель для сравнения нескольких структур тестовых данных для создания общей структуры с использованием математических множеств и алгоритма Левенштейна, который призван улучшить сходимость полей из нескольких наборов данных, а также увеличить индекс подобия.

Показатель сходимости данных отражает степень соответствия данных между различными таблицами или источниками в рамках создания общей структуры в реляционных базах данных (Рыбанов, 2020).

Показатель сходимости данных может быть определен как мера того, насколько полно и точно данные из разных источников совпадают в отношении определенного набора атрибутов или полей. Чем выше значение сходимости данных, тем более точно данные соответствуют друг другу и тем более надежна общая структура базы данных.

Цель оценки сходимости данных с общей структурой состоит в том, чтобы определить, насколько хорошо модель или теория объясняет наблюдаемые данные, и использовать эту информацию для улучшения модели или теории в будущем (Moshyedi, 2019).

Оценка сходимости данных с общей структурой включает в себя следующие этапы:

1. Формулирование модели: определение математической модели или теории, которая предсказывает общую структуру данных.
2. Сбор данных: сбор данных, которые будут использоваться для проверки модели.
3. Оценка соответствия: оценка того, насколько хорошо данные соответствуют модели. Это может включать в себя сравнение прогнозов, сделанных на основе модели, с фактическими данными, а также анализ остатков - расхождений между прогнозами и фактическими данными.
4. Оценка значимости: оценка значимости результатов и определение, насколько вероятно, что различия между моделью и данными являются статистически значимыми или случайными.
5. Интерпретация результатов: анализ результатов и интерпретация их в контексте исходной модели или теории.

Для построения общей структуры наборов данных в качестве хранилища загруженных данных предлагается использовать простейший способ выявления общих полей – пересечение математических множеств. Данный способ является простейшим в программной реализации и может быть применен без каких-либо специальных знаний в этой области. Для улучшения показателей сходимости данных из набора алгоритмов на основе семантического поиска был выбран алгоритм Левенштейна по ряду причин:

1. В исследовании (Anju et al., 2016) алгоритм Левенштейна показал хорошие результаты точности на наборе простых структурированных данных, коими являются наименования полей данных из ERP систем.
2. Гибкость. Алгоритм Левенштейна позволяет измерять семантическую разницу между строками, учитывая различные операции с данными, такими как вставка, удаление и замена символов. Это позволяет наиболее гибко производить сравнение и поиск строк, даже если они отличаются несколькими операциями редактирования.
3. Простота реализации. Алгоритм Левенштейна относительно прост для понимания и реализации. Он широко применяется в динамическом программировании и может быть реализован как рекурсивная функция или в виде таблицы для сохранения промежуточных результатов. Это делает его доступным для широкого круга разработчиков и позволяет легко интерпретировать данный алгоритм на языке Transact-SQL.

Перед вычислением сходимости нужно провести очистку данных, воспользовавшись стандартными инструментами data mining, на любом из языков программирования, к примеру – Python. Также поля в структуре данных должны быть нормализованы, в случае если они используются в реляционных базах данных. В целом общая методология для вычисления сходимости может выглядеть следующим образом:

1. Анализ первичных данных – данный этап представляет собой анализ первичных полей и их типов, сформированных в результате файловой выгрузки, либо имеющихся в системе реляционных таблиц.

2. Очистка данных – этот этап направлен на первичную обработку данных путем удаления пустых и неинформативных полей, дубликатов, разделения агрегированных и ссылочных данных.

3. Построение множества данных – на этом этапе формируются математические множества, содержащие в себе основные поля сравниваемых структур.

4. Применение алгоритма *Левенштейна и расчет сходимости* – на данном этапе сформированная с помощью математических множеств структура проверяется на сходимость еще раз, но уже с использованием алгоритма Левенштейна.

Показатель сходимости данных в общем случае может быть определен на основе индекса сходства Жаккарда (Tavor et al., 2020) с той лишь разницей, что в базах данных учитываются не все уникальные атрибуты таблицы, а общее их количество в созданном представлении:

$$DC = \frac{m}{n} \times 100 \quad (1)$$

где DC – сходимость данных; m – количество записей с совпадающими значениями по выбранным атрибутам; n – количество полей в общей структуре.

Можно также выразить m и n через отдельные счетчики для каждого атрибута:

$$m = \sum_{k=0}^{i} min(a, b), \quad (2)$$

$$n = \sum_{k=0}^{i} |a - b|, \quad (3)$$

где a и b – счетчики значений выбранного атрибута для каждого источника данных;

Чем больше сходимость данных к единице, тем эффективнее получается итоговая структура, что впоследствии сказывается на масштабировании данных.

Множество данных, может быть построено как программным способом, так и вручную.

Результаты и их обсуждения. Основой для построения множества данных послужили стандартные структуры для хранения бухгалтерских документов из нескольких популярных ERP-систем – SAP (таблица ACDOCA) и 1С (Метаданные объекта «Документ»).

На основе части стандартных полей из данных структур было построено новое множество, представляющее предварительную общую структуру путем поименного сравнения полей с использованием языка программирования Python (рис. 1).

Общую структуру данных можно представить множеством S, а наборы данных 1С и SAP могут быть представлены множествами A и B соответственно, что изображено на рис 1. Тогда можно определить, что набор данных SAP является подмножеством множества S, если каждый элемент множества A является элементом множества S. Аналогично, набор данных 1С, каждый элемент которого является элементом множества B, является подмножеством множества S.

Таким образом, можно записать:

$A \subseteq S$, где A – множество данных 1, S – множество общей структуры;

B ⊆ S, где B – множество данных 2, S – множество общей структуры;

Для определения соответствия между множествами A и S можно вычислить коэффициент пересечения множеств A и S (1):

$$A \cap S/S * 100\% . \quad (4)$$

Рисунок 1. Пересечение математических множеств наборов данных 1С и SAP

Примечание – составлено автором

Набор данных A содержит 28 полей, из которых 11 соответствуют полям в общей структуре S.

$$11/ 22 * 100 = 50 \% .$$

Аналогично для определения соответствия между множествами B и S можно вычислить коэффициент пересечения множеств B и S:

$$B \cup S/S * 100\& \% . \quad (5)$$

Набор данных B содержит 14 полей, из которых 7 соответствуют полям в общей структуре S:

$$7 / 22 * 100 = 31{,}8 \% .$$

Результаты определения сходимости данных для построенных множеств сведены в

табл. 1, где знаком «+» отмечены поля, пересекающиеся с общей структурой.

Таблица 1. Индекс сходимости структур на основе математических множеств

	Поля общей структуры																					DC
Набор данных	10	Балансовая единица	Тип докум.	Дата докум.	Дата пров.	Дата перес.	Коэф. для к	Имя пользов.	Валюта док.	Номер пак.	Позиция	Счет г.к.	Приз. суб. д	Знач. суб. д	Кор. счет	Приз. суб. к	Знач. суб. к	Сум. в вал. т	Сум. в вал. б	Сум. в вал. о	Уник. идент.	Номер	
1С	+										+	+			+					+	+	+	0,5
SAP	+	+	+	+	+			+			+	+			+			+				+	0,31
Примечание – составлено автором																							

После расчета сходимости данных для математических множеств можно улучшить результат оценки, применив алгоритм Левенштейна (Карахтанов, 2010), основанный на применении расчета минимальных расстояний до ближайшего поля, связанного с исходным одним семантическим смыслом.

Алгоритм Левенштейна определяет минимальное количество операций вставок, удалений и замены символов, необходимых для превращения одной строки в другую.

Алгоритм может быть обобщен для сравнения нескольких наборов данных путем нахождения минимального редакционного расстояния между каждой парой наборов.

Матрица, создаваемая в результате работы алгоритма Левенштейна, является двумерным массивом размерностью (m+1) на (n+1), где m и n - длины сравниваемых строк или наборов данных.

Чем меньше расстояние Левенштейна для пары полей, тем более вероятно их совпадение (рис. 2).

Теперь необходимо пересчитать оценку сходимости, с учетом найденных полей. В итоге для набора данных SAP число полей, которые соответствуют полям из общей структуры, увеличилось с 11 до 14, что составляет (1):

$$14/\ 22\ *\ 100 = 63{,}6\ \%.$$

Набор данных 1С теперь имеет 10 полей, что на 3 единицы больше, чем до применения алгоритма (2):

$$10\ /\ 22\ *\ 100 = 45{,}4\ \%.$$

Применив алгоритм Левенштейна к имеющемуся набору данных, результаты расчета показали рост показателей для наборов 1С – 0,13 и SAP – 0,14 соответственно.

Стоит отметить, что в работе алгоритма иногда появляются корреляции в виде минимальных расстояний Левенштейна для полей, не связанных семантическим смыслом, однако процент нахождения таких полей сводится к уровню погрешности (Yujian et al., 2007; Lindsay et al., 1986; Veiga et al., 2002; Ankorion, 2005; Srikanth et al., 2012).

Это связано с тем, что существуют словосочетания с одинаковой парой слов, в которых заложен один и тот же смысл, меняющийся в результате его связки с другим словом.

Примером могут являться строки «номер пакета» и «номер документа», в которых слово номер несет в себе тот же смысл, который меняется при появлении пары. В таких случаях требуется ручная корректировка.

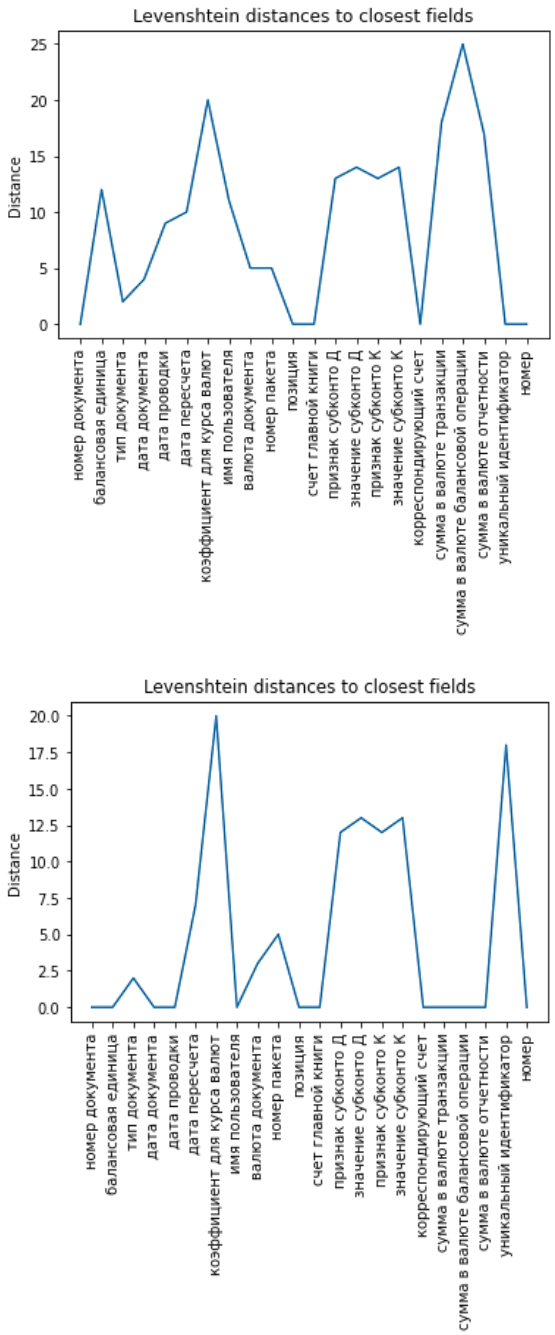

Рисунок 2. Расстояние Левенштейна до ближайших совпадений для структуры полей 1С и SAP
Примечание – составлено автором

Вывод: Процесс создания общей структуры, будь то хранение извлеченных данных из различных источников, или же временное хранилище перед загрузкой данных в целевую систему, является обязательным звеном в любой концепции репликации данных. Представлена модель изучения первичных структур ERP-систем в целях создания общей структуры для хранения извлеченных данных с применением полу-автоматизированного метода математических множеств и программного алгоритма Левенштейна, позволяющего

улучшить результат метрики согласованности данных. По результатам, полученным в данной работе, использование алгоритма Левенштейна позволяет добиться более точных результатов сходимости наборов данных. Таким образом, использование математических множеств и алгоритма Левенштейна для расчета показателя сходимости полей нескольких структур на основе оценки сходимости позволяет создавать более точные структуры под конкретные задачи пользователей и улучшить показатель сходимости данных из нескольких наборов.

Список литературы

Рыбанов А. А. (2019). Метрики разнообразия типов данных в физической схеме базы данных MySQL // Вестник Адыгейского государственного университета. Серия 4: Естественно-математические и технические науки. – № 4 (251) . – С. 87-90.

Hedge, T. (1993). Key Concepts in ELT: Fluency and Project // ELT Journal. – V 3, – P. 275-277, http://dx.doi.org/10.1093/elt/47.3.275

Pishghadam, R. (2011). Introducing applied ELT as a new paradigm // Iranian EFL Journal. – V. 7. – P. 8-14.

Piattini M., Calero C., Genero M. (2001). Table oriented metrics for relational Databases // Software Quality Journal. № 9 (2), 79–97, DOI:10.1023/A:1016670717863

Lin Y., Kemme B., et al. (2005). Middleware based data replication providing snapshot isolation // Proceedings of the 2005 ACM SIGMOD international conference on Management of data, 419-430. https://doi.org/10.1145/1066157.1066205

Emmanuel Cecchet, George Candea, and Anastasia Ailamaki. (2008). Middleware-based database replication: the gaps between theory and practice. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data (SIGMOD '08). Association for Computing Machinery, New York, NY, USA, 739–752. https://doi.org/10.1145/1376616.1376691

Рыбанов А.А. (2020). Оценка показателей разнообразия типов данных в физических схемах баз данных // Cloud of science. – Т. 7. – № 3. – С. 517-526

Moshyedi A., Kramer T., et al. (2019). A combined semantic search and machine learning approach for address entity resolution. – EasyChair. – № 832.

Anju P. R., Deepika M. P. (2016). Removal of Duplicates and Similarity Checking Using Bi-Gram Approach with Jaccard Index in Cloud Data Storage //International Journal of Computer Science and Information Security. – Т. 14. – № 9. – P. 62.

Tavor Z. Baharav, Govinda M. Kamath, David N. Tse, Ilan Shomorony. (2020). Spectral Jaccard Similarity: A New Approach to Estimating Pairwise Sequence Alignments. – Volume 1. – Issue 6.

Карахтанов Д.С. (2010). Программная реализация алгоритма Левенштейна для устранения опечаток в записях баз данных // Молодой ученый. – № 8-1. – 158-162 стр.

Yujian L., Bo L. (2007). A normalized Levenshtein distance metric //IEEE transactions on pattern analysis and machine intelligence. – Т. 29. – №. 6, 1091-1095.

Lindsay B. et al. (1986). A snapshot differential refresh algorithm //Proceedings of the 1986 ACM SIGMOD international conference on Management of data, 53-60. https://doi.org/10.1145/16894.16860

Veiga L., Ferreira P. (2002). Incremental replication for mobility support in OBIWAN //Proceedings 22nd International Conference on Distributed Computing Systems. IEEE, 249-256.

Ankorion I. (2005). Change data capture efficient ETL for real-time bi //Information Management. – Т. 15. – №. 1. – P. 36.

Srikanth K., Murthy N., Anitha J. (2012). Data Warehousing Concept Using ETL Process For SCD Type-1 // International Journal of Computer Science & Applications (TIJCSA). – Т. 1. – № 10.

Information about authors

Zhomartkyzy Gulnaz – PhD doctor, D. Serikbayev East Kazakhstan technical university, Ust-Kamenogorsk, Kazakhstan, E-mai: gzhomartkyzy@edu.ektu.kz, ORCID:0000-0003-1465-3451, +77055073381

Pavlov Konstantin Andreyevich – master of technical sciences, D. Serikbayev East Kazakhstan technical university, Ust-Kamenogorsk, Kazakhstan, E-mai: pavlov.kanstantin@gmail.com, + 77479751547

Bazarova Madina Zhomartovna – PhD doctor, S. Amanzholova East Kazakhstan University, Ust-Kamenogorsk, Kazakhstan, E-mail: madina9959843@gmail.com, ORCID: 0000-0003-2580-6580, +77055032529