

12. A. Šubrťová, D. Futschik, J. Čech, M. Lukáč, E. Shechtman, and D. Sýkora, «ChunkyGAN: real image inversion via segments», in European conference on computer vision (ECCV'22), Tel Aviv, Israel, 2022, pp. 189-204. doi: 10.1007/978-3-031-20050-2_12.
13. S. Battiato, G. Puglisi, and G. Impoco, «Vectorialisation of raster colour images», in Conferenze Nazionale del Gruppo del Colore, Bologna, Italy, 2012, pp. 11-8.
14. A. Thyssen, Resize and Scaling. Examples of ImageMagick Usage (Version 6), 2009 [Online]. Available: <http://www.imagemagick.org/Usage/resize>. Accessed on: November 15, 2022.
15. A. Thyssen, Resize and Scaling. Examples of ImageMagick Usage (Version 7), 2009 [Online]. Available: <http://www.imagemagick.org/Usage/resize>. Accessed on: November 15, 2022.
16. Byrne E. Image Processing Onramp. MATLAB URL: <https://matlabacademy.mathworks.com/details/image-processing-onramp/imageprocessing>. Accessed on: December 1, 2022.
17. P. Milanfar, *Superresolution imaging*. Boca Raton, USA: CRC Press, 2011.
18. O.N. Romanyk, and M.D. Obidnyk, «Odyn iz pidkhodiv do pidvyshchennia shvydkodii zafarbuвання» [«One of approaches to increase the speed of coloring»], *Naukovi pratsi Donetskooho natsionalnoho tekhnichnoho universytetu – Scientific papers of State higher educational institution «Donetsk national technical university»*, vol. 21(183), pp. 116-121, 2011. (Ukr.)

Рецензент: О.І. Проніна
канд. техн. наук, доц., ДВНЗ «ПДТУ»

Стаття надійшла 28.06.2023
Стаття прийнята 18.07.2023

УДК 004.896

doi: 10.31498/2225-6733.47.2023.299974

© Гончаренко Т.Д.¹, Проніна О.І.²

ПОРІВНЯННЯ АЛГОРИТМІВ ОЦІНКИ ВІДСТАНІ МІЖ СЛОВАМИ ДЛЯ ПОШУКУ СХОЖИХ РЕЧЕНЬ

У статті розглядається розробка системи пошуку схожих пропозицій на основі оцінки відстані між словами. Алгоритми неточного порівняння дають змогу пошуковим системам враховувати контекст запиту, зважаючи на можливі помилки або варіації написання слів. Це особливо важливо в умовах, коли користувачі можуть використовувати різні форми вираження однієї і тієї ж ідеї. Такі алгоритми стають ключовим елементом у створенні інтелектуальних пошукових систем, здатних розуміти суть запиту та надавати релевантні результати, навіть якщо введення містить помилки. Розроблене програмне забезпечення може бути застосовано в інформаційному пошуку, обробці природної мови, виявленні плагіату, геноміці та багатьох інших. Розглянуті в статті методи й алгоритми можуть знаходити широке застосування у сферах, де необхідний високий рівень точності в інтерпретації та зіставленні текстових даних. В інформаційному пошуку вони допомагають поліпшити якість результатів, пропонуючи користувачеві більш релевантні варіанти відповідей на його запити, навіть якщо вони містять друкарські помилки або граматичні помилки. В обробці природної мови алгоритми можуть використовуватися для аналізу і розуміння людської мови, що є ключовим аспектом у створенні чат-ботів, систем машинного перекладу та інтелектуальних асистентів. При виявленні плагіату ці алгоритми здатні точно визначати схожість текстів, що важливо в академічних і дослідницьких колах. У геноміці ці методи можуть застосовуватися для зіставлення генетичних послідовностей, що має важливе значення в

¹ магістрант, ДВНЗ «Приазовський державний технічний університет», м. Дніпро

² канд. техн. наук, доцент, ДВНЗ «Приазовський державний технічний університет», м. Дніпро, ORCID: 0000-0001-7085-8027, pronina.lelka@gmail.com

дослідженнях з біоінформатики. Таким чином, розроблене програмне забезпечення пропонує багатогранне застосування в різних галузях науки і техніки, де потрібен глибокий аналіз і розуміння текстових даних.

Ключові слова: текстовий аналіз, система пошуку, схожі речення, відстань між словами, порівняння текстових фрагментів, косинусна схожість, відстань Левенштейна, дипломний проект.

T.D. Goncharenko, O.I. Pronina. Comparison of algorithms for estimating the distance between words for finding similar sentences. The article in question delves into the intricacies of a newly developed system designed for the identification of similar sentences. This system operates on the principle of measuring the distance between words, utilizing algorithms that are adept at recognizing inexact matches. Such algorithms are vital as they enable search engines to process queries with a deeper understanding of context. They take into account potential discrepancies or variations in word spellings, a feature that becomes critically important when users employ diverse expressions to convey identical ideas. These algorithms form the backbone of intelligent search engines, equipping them with the ability to grasp the core intent of a query. This ensures the delivery of pertinent results, even when the input is marred by spelling or typographical errors. The software crafted as a result of this development finds its application across a broad spectrum of fields including information retrieval, natural language processing, plagiarism detection, and genomics, to name a few. The methodologies and algorithms highlighted in the article have significant implications in domains that demand a high degree of precision in text interpretation and comparison. In the realm of information search, these algorithms are instrumental in enhancing result quality, offering users more accurate responses to their queries, regardless of the presence of errors in spelling or grammar. In natural language processing, the algorithms play a pivotal role in analyzing and interpreting human language. This capability is fundamental for the development of advanced chatbots, machine translation systems, and intelligent digital assistants. Their application in plagiarism detection is equally noteworthy. Here, the algorithms demonstrate an exceptional ability to ascertain the degree of similarity between texts, a function that holds immense value in academic and research settings. In the field of genomics, these methods are employed for the intricate task of mapping genetic sequences, a vital process in bioinformatics research. In conclusion, the software developed as per the article presents a versatile tool, finding relevance in various scientific and technological arenas. Its ability to conduct a thorough analysis and comprehend textual data is unparalleled, marking a significant advancement in these fields.

Key words: text analysis, search system, similar sentences, distance between words, comparison of text fragments, cosine similarity, Levenshtein distance, diploma project.

Постановка проблеми. Область алгоритмів неточного порівняння рядків – це сфера, яка досліджує та розробляє методи для визначення ступеня схожості між текстовими рядками, з урахуванням можливих помилок, вставок, вилучень чи змін символів. Ця предметна область є ключовою в контексті аналізу текстової інформації в різних галузях, включаючи інформаційний пошук, обробку природної мови, плагіат-виявлення, геноміку та багато інших.

Область алгоритмів неточного порівняння рядків є критично важливою в контексті опрацювання текстової інформації, де неминучі друкарські помилки, варіації написання слів і різні форми вираження схожих ідей. У сучасних пошукових системах, наприклад, актуальність цієї царини проявляється в прагненні забезпечити точність і повноту результатів пошуку під час введення користувачем запитів із можливими орфографічними помилками.

Алгоритми неточного порівняння дають змогу пошуковим системам врахувати контекст запиту, враховуючи можливі помилки або варіації написання слів. Це особливо важливо в умовах, коли користувачі можуть використовувати різні форми вираження однієї й тієї самої ідеї. Такі алгоритми стають ключовою ланкою у створенні інтелектуальних пошукових систем, здатних розуміти зміст запиту і надавати релевантні результати, навіть якщо введення містить помилки.

Аналіз останніх досліджень та публікацій. В області алгоритмів вимірювання відстані між реченнями є безліч робіт.

У роботі [1] авторами було досліджено різні метрики відстані між рядками для задачі зіставлення імен. Особлива увага приділялася завданням, пов'язаним із зіставленням імен в онлайн-просторі. Автори провели порівняльний аналіз ефективності різних метрик, включно з косинусною схожістю і метрикою Жаккара, а також запропонували свої рекомендації щодо вибору метрик залежно від конкретних сценаріїв використання.

Робота [2] В.І. Левенштейна, опублікована 1966 року, представила один із фундаментальних алгоритмів у галузі алгоритмів неточного порівняння рядків. Автор запропонував алгоритм редакційної відстані, який вимірює мінімальну кількість операцій (вставок, вилучень, замінів), необхідних для перетворення одного рядка в інший. Цей алгоритм виявився ключовим для багатьох завдань, включно з виправленням помилок і пошуком схожих фрагментів тексту.

У статті [3] G. Navarro представив огляд і порівняння різних методів пошуку наближених збігів рядків. Автор розглянув методи, засновані на деревах і префіксних деревах, а також надав аналіз їх застосовності в різних сценаріях. Робота стала важливим ресурсом для дослідників, які прагнуть зрозуміти й вибрати оптимальний метод пошуку збігів рядків залежно від контексту.

Робота [4] E. Ukkonen запропонувала алгоритми для наближеного пошуку схожих текстових фрагментів. Автор досліджував різні методи вимірювання відстані між рядками і запропонував їх застосування в контексті пошуку схожих текстів. Це дослідження поклало основу для розробки ефективних алгоритмів пошуку збігів у текстових даних.

У статті [5] M. Jaro було представлено метрику Jaro, розроблену спеціально для порівняння імен у задачах зіставлення записів. Автор запропонував формулу розрахунку метрики, що враховує загальні символи і порядок символів в іменах. Цей підхід виявився ефективним у задачах порівняння імен, наприклад, при зіставленні записів у базах даних.

Робота [6] G. Landau і U. Vishkin розглядає ефективні методи пошуку апроксимованих збігів у рядках. Автори запропонували алгоритми, що забезпечують лінійний час виконання, що є важливим аспектом в обробці великих обсягів текстової інформації. Робота стала важливим внеском у розробку ефективних алгоритмів пошуку збігів в умовах великих даних.

У роботі [7] Moffat A. та Zobel J. «Самоіндексування інвертованих файлів для швидкого пошуку тексту» автори заглиблюються у сферу самоіндексування інвертованих файлів як засобу досягнення швидкого пошуку тексту. Основна увага в дослідженні приділяється розробці та використанню методів самоіндексації в інвертованих файлових структурах для підвищення ефективності процесів текстового пошуку.

Робота [8] G. Myers описує алгоритм порівняння рядків з лінійним часом виконання, відомий як алгоритм з найменшими відмінностями (An O(ND) Difference Algorithm). Автор пропонує метод, який оптимально опрацьовує випадки вставок і вилучень у рядках, що є важливим у контексті пошуку збігів з можливими змінами.

У статті [9] A. Monge і C. Elkan представляють алгоритми порівняння текстових даних, сфокусовані на завданнях зіставлення записів та інтеграції інформації. Автори надають аналіз ефективності різних методів зіставлення записів і пропонують рекомендації для вибору відповідного методу залежно від конкретних вимог завдання.

Робота [10] S. Wu і U. Manber описує алгоритм пошуку схожості в тексті, який дозволяє помилки, такі як вставки і видалення, при ефективному виконанні пошуку. Автори представляють метод, який забезпечує ефективність під час пошуку збігів в умовах неточних даних, що є важливим аспектом під час обробки реальних текстових даних.

Мета дослідження. Метою даної роботи є підвищення ефективності порівняння строк за рахунок розробки методів та математичної моделі що лежить в основі.

Виклад основного матеріалу. Далі були розглянуті алгоритми схожих речень на основі оцінки відстані між словами. Спочатку було детально вивчено й описано різні алгоритми для вимірювання відстані між рядками.

Алгоритм Левенштейна, також відомий як редакційна відстань, використовували для вимірювання мінімальної кількості односимвольних операцій (вставка, видалення, заміна), необхідних для перетворення одного рядка в інший. Основа алгоритму полягала в матриці, де рядки відповідали символам першого рядка, а стовпці – символам другого рядка. Кожен осередок

матриці містив число, що представляло мінімальну відстань між підрядками першого і другого рядка, сформованими відповідними рядками і стовпчиками.

Наступний алгоритм Хеммінга описувався як міра відмінності між двома рядками однакової довжини. Він обраховував кількість позицій, на яких відповідні символи відрізнялися. Відстань Хеммінга порівнює два рядки однакової довжини та обчислює кількість позицій, у яких відповідні символи відрізняються. Він, в основному, використовується для двійкових рядків і не застосовується до рядків різної довжини.

Алгоритм Soundex, фонетичний метод індексації слів за їхнім звучанням в англійській мові, використовувався для кодування гомофонів в один і той самий код. У рамках алгоритму першу букву слова зберігали, а решту букв перетворювали на цифри за певними правилами. Послідовні дублікати чисел видаляли, виключаючи першу літеру, і формували кінцевий код, що складається з першої літери слова і трьох цифр, отриманих із літер, що залишилися. Цей метод був особливо корисний у базах даних і системах пошуку для спрощення пошуку за фонетичною схожістю.

Далі Відстань Жаккара – це міра різниці між двома струнами однакової довжини. Він підраховує кількість позицій, на яких відповідні символи відрізняються. Подібність Жаккара оцінює подібність між двома наборами шляхом вимірювання розміру їх перетину відносно їх об'єднання. Хоча спочатку він застосовувався до наборів, він був адаптований для використання в різних областях, включаючи обробку природної мови.

Косинусна подібність – це показник, який використовується для вимірювання схожості двох векторів, особливо в контексті просторів великої розмірності, таких як текстові дані. У випадку слів ми часто представляємо їх як вектори в просторі, де кожен вимір відповідає унікальному терміну або слову. Косинус подібності двох векторів обчислюється на основі косинуса кута між ними.

У контексті неточного порівняння двох слів, це зазвичай передбачає оцінку їхньої подібності чи спорідненості за значенням, а не точних збігів. У обробці природної мови одним із поширених способів представлення слів є використання вбудованих слів, які відображають слова в безперервні векторні простори.

Подібність косинусів особливо корисна в цьому контексті, оскільки вона залежить не від величини векторів, а лише від їх орієнтації. Це робить його ефективним для порівняння напрямку векторів, що представляють слова у багатовимірному просторі, фіксуючи семантичну подібність між словами.

Після розгляду алгоритмів була розроблена система для перевірки ефективності алгоритмів. На початку роботи система здійснює импорт текстових документів, приховуючи від користувача деталі обробки та форматування тексту. Наступним кроком є токенизація тексту, тобто його розбиття на окремі речення з використанням неявних методів. Цей процес також може включати видалення стоп-слів для підвищення точності аналізу, оскільки вони часто зустрічаються в тексті, але не несуть значущої інформації про зміст.

Далі система переходить до порівняння відповідних речень із двох текстів, застосовуючи обрану метрику схожості, як-от відстань Левенштейна або косинусна схожість. Речення, визнані схожими, виводяться в текстові поля, а також формується статистика, що включає кількість схожих речень і загальну відстань між текстами.

Завершальний етап роботи системи – це виведення результатів і статистики. На цьому етапі створюються текстові віджети, в які записуються речення, визнані алгоритмом схожими. Крім того, алгоритм формує статистичну інформацію, що містить кількість схожих речень, загальну дистанцію між текстами та інші показники, корисні для аналізу. Виведення результатів здійснюється з урахуванням форматування, роблячи інформацію більш читабельною і зрозумілою для користувача.

Таким чином, розроблена система забезпечує комплексний підхід до аналізу текстів, надаючи користувачам точні та інформативні результати порівняння текстів на рівні окремих речень.

Після розроблення програмного забезпечення з реалізацією перерахованих вище алгоритмів було проведено кілька експериментів. Мета цих експериментів – оцінити, наскільки добре кожен алгоритм впорається із завданням визначення схожості та виявлення відмінностей між двома текстами, які загалом схожі, але містять деякі ключові відмінності. Це дасть нам змогу зрозуміти, які алгоритми більш чутливі до змін у тексті та як вони можуть бути використані для

виявлення й аналізу відмінностей у текстових даних. Такий підхід має особливе значення в завданнях, пов'язаних із перевіркою на плагіат, редагуванням текстів, а також в інших додатках, де важливо визначити ступінь зміни текстового змісту.

Результати одного з експериментів наведено в таблиці 1.

Таблиця 1

Результати експерименту

Алгоритм	Відстань	Точність	Час виконання
Левенштейн	20	100%	00.000999 секунд
Хеммінг	5	90%	00.004005 секунд
Жаккар	7.62	50%	00.000999 секунд
Косинусна подібність	7.81	30%	00.082002 секунд
Soundex	800	10%	00.005005 секунд

Під час проведення експериментів було зібрано дані, що відображають результати роботи різних алгоритмів опрацювання тексту, які було представлено у вигляді трьох графіків: відстані, точності та часу виконання (рис. 1-3).

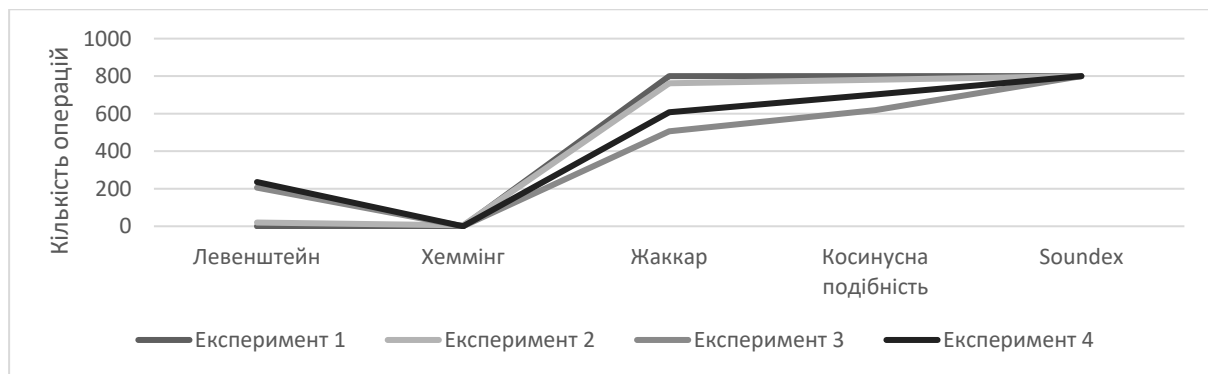


Рис. 1 – Графік результатів відстань

На графіку відстані спостерігалася тенденція, згідно з якою алгоритм Левенштейна показав найменшу відстань, що вказує на його високу чутливість до змін у тексті. Водночас, алгоритм Soundex продемонстрував значно більшу відстань, що свідчить про його обмежену спроможність до аналізу повних текстів у даному контексті. Алгоритми Жаккара та косинусної подібності показали проміжні значення відстані, що відображає їхню помірну чутливість до текстових змін.

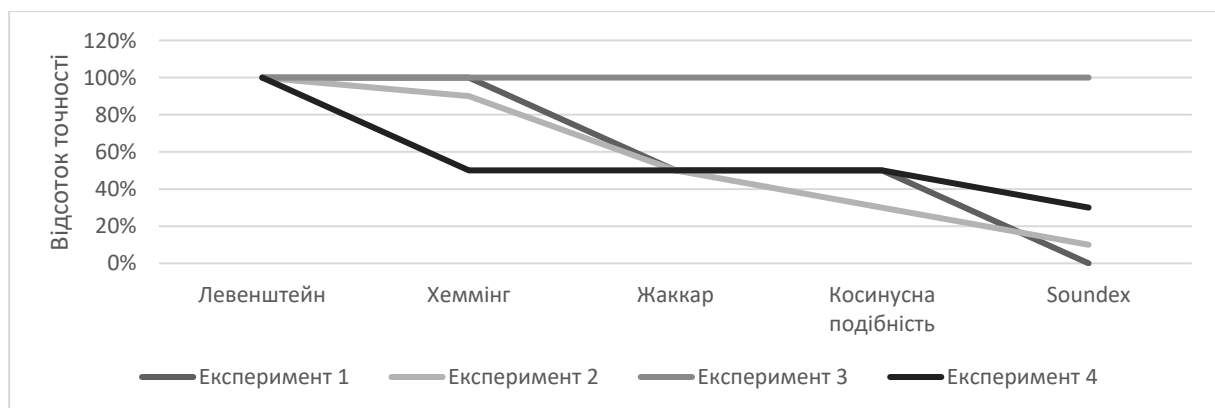


Рис. 2 – Графік результатів точності

На графіку точності було помітно, що алгоритм Левенштейна досяг 100% точності, що підтверджує його ефективність у визначенні ступеня змін між двома текстами. Алгоритм Хеммінга також показав високу точність, хоча його застосування було обмежене порівнянням текстів однакової довжини. На відміну від цього, алгоритми Жаккара, косинусної подібності та Soundex продемонстрували нижчу точність, що вказує на їхні потенційні обмеження в завданнях точного порівняння текстів зі змінами.

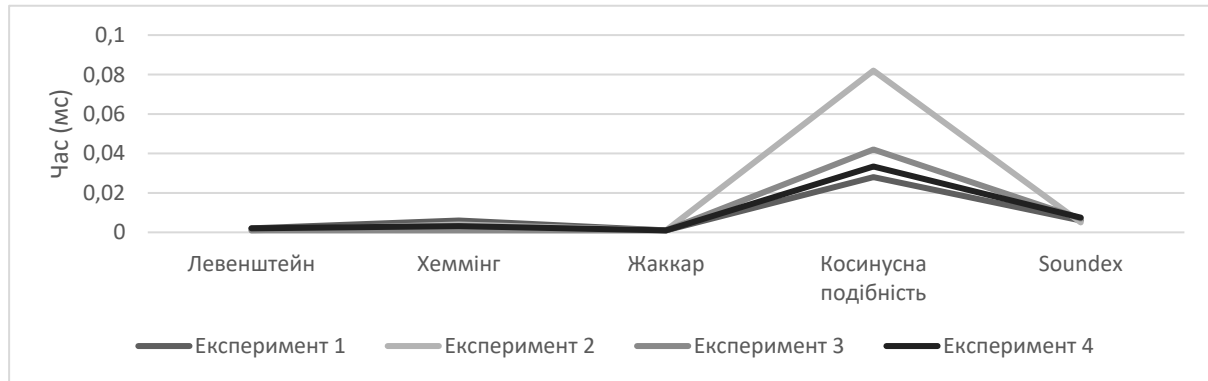


Рис. 3 – Графік результатів часу виконання

Нарешті, на графіку часу виконання алгоритм Левенштейна показав найкращі результати, завершивши завдання за найкоротший термін. Це підкреслює його перевагу у швидкості обробки даних. Алгоритм Хеммінга зайняв трохи більше часу, тоді як косинусна подібність вимагала значно більше часу для виконання того самого завдання. Алгоритм Soundex, незважаючи на свою низьку точність, показав середній час виконання.

У результаті, алгоритм Левенштейна проявив себе як найефективніший у експериментах, демонструючи високу точність і швидкість під час визначення змін у тексті. Ці результати підкреслюють важливість вибору відповідного алгоритму залежно від специфіки завдання і характеру оброблюваних текстових даних. Алгоритм Левенштейна, завдяки своїй гнучкості та ефективності, може бути особливо корисним у додатках, де потрібне швидке і точне порівняння текстів зі змінами.

Висновки

Під час дослідження було продемонстровано, що алгоритми неточного порівняння рядків, як-от алгоритми Левенштейна, Хеммінга, косинусної подібності, Жаккара та Soundex, відіграють ключову роль у розробленні ефективних систем пошуку схожих речень на основі оцінки відстані між словами. Ці алгоритми дають змогу пошуковим системам враховувати контекст запиту, з огляду на можливі помилки або варіації написання слів, що особливо важливо в умовах, коли користувачі можуть використовувати різні форми вираження однієї й тієї самої ідеї.

Алгоритм Левенштейна, що особливо вирізняється своєю здатністю вимірювати мінімальну кількість односимвольних операцій для перетворення одного рядка на інший, демонструє високу точність і ефективність у завданнях порівняння текстів. Алгоритм Хеммінга, хоча й обмежений порівнянням рядків однакової довжини, також показує хороші результати в певних сценаріях. Косинусна подібність, що використовує векторне представлення текстів, ефективна у високорозмірних просторах, особливо в додатках обробки природної мови. Алгоритм Жаккара, що фокусується на подібності між двома наборами, виявляється корисним під час порівняння текстів, представлених у вигляді наборів токенів. Нарешті, Soundex, хоча й обмежений у своєму застосуванні до англійської мови та фонетичного порівняння, пропонує цінні можливості в завданнях, де важлива фонетична схожість слів.

Таким чином, кожен із цих алгоритмів робить свій внесок у розробку систем пошуку схожих пропозицій, водночас їхня ефективність і застосовність залежать від конкретних вимог і характеру оброблюваних текстових даних. У результаті, вибір відповідного алгоритму значною мірою визначається специфікою завдання і вимогами до точності та швидкості обробки даних.

Перелік використаних джерел:

1. Cohen W.W., Ravikumar P. A comparison of string distance metrics for name-matching tasks. *Proceedings of the 2003 International Conference on Information Integration on the Web*, Acapulco, Mexico, 9-10 August 2003. Pp. 73-78.
2. Levenshtein V.I. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*. 1966. Vol 10. № 8. Pp. 707-710.
3. Navarro G. A guided tour to approximate string matching. *ACM Computing Surveys*. 2001. Vol. 33. Iss. 1. Pp. 31-88. DOI: <https://doi.org/10.1145/375360.375365>.
4. Ukkonen E. Algorithms for approximate string matching. *Information and control*. 1985. Vol. 64. Iss. 1-3. Pp. 100-118. DOI: [https://doi.org/10.1016/S0019-9958\(85\)80046-2](https://doi.org/10.1016/S0019-9958(85)80046-2).
5. Jaro M.A. Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. *Journal of the American Statistical Association*. 1989. Vol. 84. No. 406. Pp. 414-420. DOI: <https://doi.org/10.2307/2289924>.
6. Landau G.M., Vishkin U. Fast string matching with k differences. *Journal of Computer and System Sciences*. 1988. Vol. 37. Iss. 1. Pp. 63-78. DOI: [https://doi.org/10.1016/0022-0000\(88\)90045-1](https://doi.org/10.1016/0022-0000(88)90045-1).
7. Moffat A., Zobel J. Self-indexing inverted files for fast text retrieval. *ACM Transactions on Information Systems*. 1996. Vol. 14. № 4. Pp. 349-379. DOI: <https://doi.org/10.1145/237496.237497>.
8. Myers G. An O(ND) difference algorithm and its variations. *Algorithmica*. 1986. Vol. 1. Pp. 251-266. DOI: <https://doi.org/10.1007/BF01840446>.
9. Monge A.E., Elkan C. The field matching problem: Algorithms and applications. *KDD'96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, Portland, USA, 2-4 August 1996. Pp. 267-270.
10. Wu S., Manber U. Fast text searching allowing errors. *Communications of the ACM*. 1992. Vol. 35. Iss. 10. Pp. 83-91. DOI: <https://doi.org/10.1145/135239.135244>.

References:

1. W.W. Cohen, and P. Ravikumar, «A comparison of string distance metrics for name-matching tasks», in Proceedings of the 2003 International Conference on Information Integration on the Web, Acapulco, Mexico, 2003, pp. 73-78.
2. V.I. Levenshtein, «Binary codes capable of correcting deletions, insertions, and reversals», *Soviet Physics Doklady*, vol. 10, № 8, pp. 707-710, 1966.
3. G. Navarro, «A guided tour to approximate string matching», *ACM Computing Surveys*, vol. 33, iss. 1, pp. 31-88, 2001. doi: **10.1145/375360.375365**.
4. E. Ukkonen, «Algorithms for approximate string matching», *Information and control*, vol. 64, iss. 1-3, pp. 100-118, 1985. doi: **10.1016/S0019-9958(85)80046-2**.
5. M.A. Jaro, «Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida», *Journal of the American Statistical Association*, vol. 84, no. 406, pp. 414-420, 1989. doi: **10.2307/2289924**.
6. G.M. Landau, and U. Vishkin, «Fast string matching with k differences», *Journal of Computer and System Sciences*, vol. 37, iss. 1, pp. 63-78, 1988. doi: **10.1016/0022-0000(88)90045-1**.
7. A. Moffat, and J. Zobel, «Self-indexing inverted files for fast text retrieval», *ACM Transactions on Information Systems*, vol. 14, № 4, pp. 349-379, 1996. doi: **10.1145/237496.237497**.
8. G. Myers, «An O(ND) difference algorithm and its variations», *Algorithmica*, vol. 1, pp. 251-266, 1986. doi: **10.1007/BF01840446**.
9. A.E. Monge, and C. Elkan, «The field matching problem: Algorithms and applications», in KDD'96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Portland, USA, 1996, pp. 267-270.
10. S. Wu, and U. Manber, «Fast text searching allowing errors», *Communications of the ACM*, vol. 35, iss. 10, pp. 83-91, 1992. doi: **10.1145/135239.135244**.

Рецензент: Т.О. Левицька
канд. техн. наук, доц., ДВНЗ «ПДТУ»

Стаття надійшла 17.09.2023
Стаття прийнята 20.10.2023