Хмельницький національний університет
Факультет програмування та комп'ютерних і телекомунікаційних систем
Кафедра кібербезпеки та комп'ютерних систем і мереж

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Метод ідентифікації особистості на основі операцій несуворої відповідності та вагових коефіцієнтів

Назва теми

Галузь знань 12 – Інформаційні технології

Спеціальність 123 –Комп'ютерна інженерія

КРМКІ. 015070.19.01.10 ПЗ

Виконав: студент 2 курсу, група КІ1м-19-1 Підпис Мозолюк В.О.

Керівник доц., к. т. н, доцент кафедри КбКСМ Підпис Орленко В.С.

Нормоконтролер доц., к. т. н, доцент кафедри КбКСМ Підпис Муляр І.В.

До захисту допускаю:
Зав. кафедри КбКСМ, к.т.н., доц Підпис Кльоц Ю.П.

4 12 2020 р.

Хмельницький, 2020

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ПРОГРАМУВАННЯ ТА КОМП'ЮТЕРНИХ І ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ

Кафедра КІБЕРБЕЗПЕКИ ТА КОМП'ЮТЕРНИХ СИСТЕМ І МЕРЕЖ

Освітній рівень МАГІСТР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА ПІДГОТОВКИ МАГІСТРА

ЗАТВЕРДЖУЮ
Завідувач кафедри кібербезпеки та
комп'ютерних систем і мереж
к.т.н. доцент Кльоц Ю.П.
"4" 09 2020 року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Мозолюку Віталію Олександровичу
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Метод ідентифікації особистості на основі операцій несуворої відповідності та вагових коефіцієнтів
Науковий керівник Орленко Вікторія Сергіївна, к.т.н.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджена наказом № 118 ректора університету додаток №23 від 01.09.2020
2. Строк подання студентом проекту (роботи) на кафедру 3.12.2020
3. Вихідні дані до проекту (роботи) Провести дослідження існуючих моделей, методів, а також алгоритмів використовуваних в інформаційно-пошукових системах. Розробити модель представлення релевантності подібності рядків та модель наближеного пошуку інформації при опрацюванні пошукових рядків. Розробити алгоритм оптимізації запису інформації в бази даних та алгоритм пошуку інформації в базах даних за реквізитами особистості. Провести дослідження ефективності запропонованих алгоритмів
4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)
Дослідження інструментарію ідентифікації об'єктів інформаційними системами.
Розробка моделей опису процесу ідентифікації об'єктів в базах даних.
Розробка алгоритмів наближеного пошуку інформації в базах даних.
Використання алгоритмів наближеного пошуку інформації на підприємствах
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) 1,2.Тема, мета магістерської роботи, об'єкт, предмет, задачі дослідження, наукова новизна, практична цінність, апробація роботи. 3. Порівняння алгоритмів пошуку за ефективністю та швидкістю. 4. Модель представлення релевантності подібності рядків. 5. Графік прийняття рішення при N= {1..5}. 6. Удосконалений метод оптимізації запису інформації в бази даних. 7. Алгоритм оптимізації запису інформації в бази даних. 8. Метод формування пошукового індексу за реквізитами особистості. 9. Алгоритм пошуку інформації в базах даних за реквізитами особистості 10. Архітектура системи «Оптимізація інформації при запису в бази даних». 11.Висновки.

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, да	
		завдання видав	зав
Відповідальний за оформлення КРМ	Муляр І.В., доцент, к.т.н		

7. Дата видачі завдання: «_1_» лютого 2020 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи
1	Ґрунтовне ознайомлення з предметною галуззю	2.02.2020
2	Визначення структури магістерської роботи	2.03. 2020
3	Робота над першим розділом магістерської роботи	1.04. 2020
4	Робота над першою статтею за результатами обробки літературних джерел	1.05. 2020
5	Робота над другим розділом магістерської роботи	1.06. 2020
6	Робота над третім розділом магістерської роботи	1.09. 2020
7	Робота над четвертим розділом магістерської роботи	1.10. 2020
8	Підготовка ілюстративного матеріалу	1.11. 2020
9	Оформлення текстової і графічної частини магістерської роботи	11.11. 2020
10	Попередній захист магістерської роботи	21.11. 2020
11	Захист ДРМ на засіданні ЕК	08.12. 2020

Студент _____ Мозолюк Е
Підпис Ініціали, пр

Керівник проекту (роботи) _____ Орленко В
Підпис Ініціали, пр

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Метод ідентифікації особистості на основі операцій несуворої відповідності та вагових коефіцієнтів».

Автор роботи: Мозолюк Віталій Олександрович

Керівник роботи: ктн., доц. Орленко Вікторія Сергіївна

Пояснювальна записка: 95 сторінок, 13 рисунків, 20 таблиць., 33 посилань, 3 додатки.

ПОШУКОВІ СИСТЕМИ, ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ, ПОШУКОВИЙ ШАБЛОН, БАЗИ ДАНИХ, РЯДОК ПОШУКУ, ІНФОРМАЦІЙНА ВЗАЄМОДІЯ.

Метою дослідження є оптимізації запису інформації в бази даних, на основі використання значення релевантності подібності рядків, в результаті реалізації запропонованої моделі релевантності подібності рядків.

Метою дипломної роботи є вдосконалення методу оптимізації запису інформації в бази даних, забезпечує розпізнавання та виключення дублювання даних при використанні інформаційно – пошукових систем, на основі автоматичного вибору схеми ручної або автоматичної ідентифікації, що дозволяє зберегти інформаційну цілісність, а також знизити зашумленість даних, зумовлену наявністю помилок операторського введення. Запропоновані алгоритми можуть бути використання в якості типових рішень при проектуванні та застосуванні подібних систем для підприємств середнього та малого бізнесу.

4. 12. 2020 Дата

Підпис студента

ANNOTATION

a master's degree work of Mozolyuk Vitaly

entitled «Personality identification method based on non-strict compliance operations and weights».

Mentor: Victoria Orlenko

Total volume of work: 95 pages, 13 figures, 20 tables, 3 appendices, 33 references.

SEARCH ENGINES, INFORMATION SUPPORT, SEARCH TEMPLATE, DATABASES, SEARCH STRING, INFORMATION INTERACTION

The aim of the study is to optimize the recording of information in the database, based on the use of the value of the relevance of the similarity of the lines, as a result of the implementation of the proposed model of the relevance of the similarity of the lines.

The purpose of the thesis is to improve the method of optimizing the recording of information in the database, provides recognition and exclusion of data duplication when using information retrieval systems, based on automatic selection of manual or automatic identification, which allows to preserve information integrity and reduce noise caused by data operator input errors. The proposed algorithms can be used as typical solutions in the design and application of such systems for medium and small businesses.

7. 12. 2020 Date

Signature

ЗМІСТ

ВСТУП

Актуальність теми. Починаючи з середини 20 століття, поводились активні дослідження з проблем присвяченим інформаційному пошуку [3]. З поширенням в кінці 20 століття Інтернет, методи зберігання та пошуку інформації суттєво змінилися, що дозволило поставити питання про проблему використовуваного інформаційного пошуку[4]. Інформаційні технології на сучасному етапі все більше проникають в сфери життя суспільства, пересилання документів в електронному вигляді, а також пошук інформації, встановлення, ідентифікація особистості злочинців, які вчиняють кримінальні правопорушення, стали звичним явищем сьогодення.

На даний час проаналізовано та досліджено багато матеріалів, порівняльних оглядів в області інформаційно - пошукових технологій, але в силу величезної швидкості їх розвитку, зібрана інформація миттєво застаріває. Незважаючи на велику кількість опрацьованих матеріалів, експериментів, технології пошуку постійно змінюються.

На сьогоднішній день проводяться дослідження, присвячені щодо підвищення ефективності пошукових систем за ключовими словами. Основна увага приділяється наступним напрямками розвитку: підвищення якості, отриманої інформації, оцінки релевантності документів; побудова моделі клієнта, з метою видачі більш релевантної інформації пошукової системою конкретному клієнту.

Немаловажним фактором є наскільки надійна і захищена інформаційно – пошукова система від зовнішніх атак, так званих SQL-ін'єкцій. Експертами компанії Positive Technologies в 2018р. були проведені дослідження різних галузей діяльності підприємств. Атаки типу SQL-ін'єкції продовжують непреривно зростати. Так чином критично небезпечна вразливість SQL-ін'єкцій піднялася з шостого місця на четверте, тобто склала 62% проти 48% в 2017 році. Лідером за кількістю знайдених вразливостей є банківська галузь (89%), на другому місці -

телекомунікаційна галузь (80%). Далі йдуть промисловість (71%) і інформаційні технології. Останнє десятиріччя XX ст. характеризується широкою комп'ютеризацією всіх видів діяльності людства: від традиційних інтелектуальних задач наукового характеру до автоматизації виробничої, комерційної, торгової, банківської та інших видів людської діяльності. Таким чином в умовах ринкової економіки конкурентну боротьбу можуть успішно витримувати тільки ті підприємства, які у своїй діяльності використовують сучасні прогресивні інформаційні технології [3]. Сучасний етап розвитку людства не представляється можливим без ефективного методу об'єднання існуючих інформаційних ресурсів та інформаційно-пошукових систем забезпечення надійного прогресивного розвитку всіх галузей індустрії.

Разом з тим, присутні фактори, які стримують розвиток інформаційно-пошукових систем, до яких можна віднести: проблеми управління якістю інформаційними потоками в базах даних, а також помилки в пошукових запитах та безпосередньо в самих базах даних На сьогодні не існує універсального підходу їх вирішення.

Розробка методів і спеціальних технологій інформаційного пошуку, з використанням при цьому, нетривіальних рішень, у тому числі з використанням оптимізації потоку даних в базах даних на сьогоденні стає актуальною задачею.

Об'єкт дослідження – технологія процесу пошуку даних в базах даних інформаційно – пошуковими системами.

Предмет дослідження − є застосування моделей, методів підвищення рівня якісного та ефективного інформаційного забезпечення підрозділів підприємства за рахунок проведення оптимізації інформації, в базах даних.

Мета і завдання дослідження дипломної роботи. Метою дослідження є оптимізації запису інформації в бази даних, на основі використання значення релевантності подібності рядків, в результаті реалізації запропонованої моделі релевантності подібності рядків.

Відповідно до поставленої мети в дипломній роботі поставлені, і вирішені наступні задачі:

1. Провести дослідження існуючих моделей, методів, а також алгоритмів використовуваних в інформаційно-пошукових системах.

2. Розробити модель представлення релевантності подібності рядків та модель наближеного пошуку інформації при опрацюванні пошукових рядків.

3. Розробити алгоритм оптимізації запису інформації в бази даних та алгоритм пошуку інформації в базах даних за реквізитами особистості.

4. Провести дослідження ефективності запропонованих алгоритмів.

Методи дослідження. Для вирішення задач поставлених в дипломній роботі використовуються методи системного аналізу, імовірнісних графів, випадкових процесів і математичної статистики, теорії ймовірності, теорії прийняття рішень, методів комп'ютерного аналізу, математичного моделювання, методів модульного і структурного програмування.

Наукова новизна одержаних результатів:

1. Моделі представлення релевантності подібності рядків, дозволяє надати, для рядків пошуку, кількісну оцінку їх подібності.

2. Удосконалений метод оптимізації запису інформації в бази даних, забезпечує розпізнавання та виключення дублювання даних при використанні інформаційно – пошукових систем, на основі автоматичного вибору схеми ручної або автоматичної ідентифікації, що дозволяє зберегти інформаційну цілісність, а також знизити зашумленість даних, зумовлену наявністю помилок операторського введення.

Практична цінність результатів дипломної роботи. Запропоновані алгоритми під час виконання дипломної роботи, можуть бути використання в якості типових рішень при проектуванні та застосуванні подібних систем для підприємств середнього та малого бізнесу.

Достовірність наукових положень, висновків отриманих в дипломній роботі результатів підтверджується коректною постановкою задач, результатами

моделювання та апробацією результатів отриманих на конференціях, коректністю використовуваного математичного апарату. Отримані в ході виконання дисертаційного дослідження результати не суперечать раніше отриманим даним, описаним в літературі іншими авторами.

Особистий внесок. Всі дослідження, викладені в дипломній роботі, проведені автором в процесі наукової діяльності. Результати, які виносяться на захист, отримані автором особисто, запозичений матеріал позначений в роботі посиланнями.

Апробація роботи. За темою дипломної роботи ОКР «Магістр» опубліковано 1 теза та 1 стаття.

Структура і обсяг роботи. Дипломна робота ОКР «Магістр» складається зі вступу, основної частини, що містить 4 розділи, висновків і списку використаних джерел. Загальний обсяг роботи - 95 сторінок. Робота містить 13 рисунків та 20 таблиць. Список використаної літератури включає 33 бібліографічних джерела.

1 ДОСЛІДЖЕННЯ ІНСТРУМЕНТАРІЮ ІДЕНТИФІКАЦІЇ ОБ'ЄКТІВ ІНФОРМАЦІЙНИМИ СИСТЕМАМИ

1.1 Аналіз сучасних інформаційно - пошукових систем та їх класифікація

На ранніх стадіях еволюції інформаційних пошукових систем враховувалося мала кількість факторів, що впливають на ранжування у видачі результатів пошуку, таким чином знаючи базові принципи роботи пошукових систем, можна було досить легко маніпулювати результатами.

В результаті, пошукові системи були змушені ускладнювати свої алгоритми. Кількість чинників, що враховуються зросли в сотні, і навіть тисячі разів. На сучасному етапі, незважаючи на бурхливий розвиток технологій в області інформаційно-пошукових систем, саму інформацію про ефективність та якість сучасних пошукових системах отримати досить непросто.

Починаючи з середини 20 століття, проводились активні дослідження з проблем присвяченим інформаційному пошуку [3]. З поширенням в кінці 20 століття Інтернет, методи зберігання та пошуку інформації суттєво змінилися, що дозволило поставити питання про проблему використовуваного інформаційного пошуку[4].

Інформаційні технології на сучасному етапі все більше проникають в сфери життя суспільства, пересилання документів в електронному вигляді, а також пошук інформації, пошук, встановлення, ідентифікація особистості злочинців, які вчиняють кримінальні правопорушення, стали звичним явищем сьогодення.

На даний час проаналізовано та досліджено багато матеріалів, порівняльних оглядів в області інформаційно - пошукових технологій, але в силу величезної швидкості їх розвитку, зібрана інформація миттєво застаріває. Незважаючи на велику кількість опрацьованих матеріалів, експериментів, технології пошуку постійно змінюються. Для пошукових систем постійний та непреривний розвиток - це боротьба за клієнтів, а більш точно, трафік і його монетизація.

Інформаційний пошук - пошук даних, пошук документів а також пошук фактів. Інформаційно-пошукова система - програмно-апаратна система зі спеціальним програмним забезпеченням призначена для зберігання, пошуку і надання відповідному клієнтові корисної інформації. Відповідно до поставленої задачі в дипломній роботі під терміном інформаційно-пошукової системи будемо визначати як документальна інформаційно-пошукова система – ідентифікації особистостей, призначена для відшукання відповідної інформації (документів), що містять необхідну корисну оператору інформацію.

Розглянемо принципи функціонування інформаційно-пошукової системи ідентифікації особистостей. Загальний алгоритм функціонування інформаційно-пошукова система наведений на рис. 1.1. Процеси, що протікають в ІПС є пошук документів, інформації ідентифікації особистостей основні признаки яких відображенні в запиті а також індексування документів.

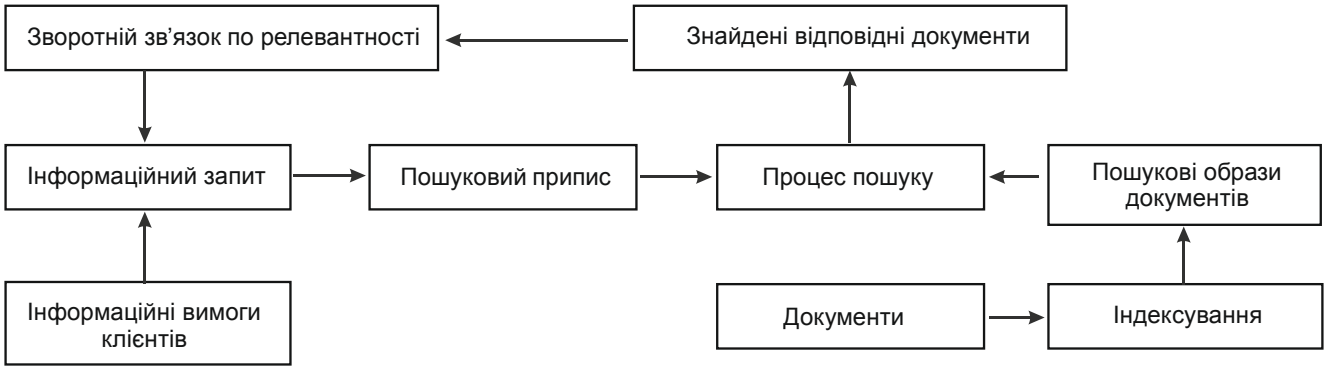


Рисунок 1.1 - Алгоритм функціонування інформаційно-пошукова система.

Процес інформаційного пошуку в базі даних відбувається за наступним алгоритмом. Клієнт надає свої інформаційні вимоги у вигляді спеціального форматованого тексту - інформаційного запиту до інформаційно-пошукової система. Пошукова система формує з інформаційного запиту клієнта пошуковий припис, переводячи запит клієнта на зрозумілу для інформаційно-пошукової систему мову. Інформаційно-пошукова мова представляє собою формальну мову Хомського, яка використовується всередині інформаційно-пошукової системи для представлення запиту клієнта а також для збереження відповідних документів. Результатом пошуку ІПС вибрані з масиву документів та інформація, яка

змістовно релевантна запиту клієнта, відповідає інформаційним запитам клієнта, зазначених в запиті. Розглядають формальну релевантність, як відповідність, яка визначається програмно – апаратним шляхом (алгоритмічно) співставленням порівняння пошукового припису з пошуковим образом документа. Критерій, який визначає видачу документа (пошукової інформації) - формальне правило, яке визначає ступінь формальної релевантності пошукового припису інформаційного документа і пошукового запиту, згідно за яким приймається відповідне рішення про видачу деякого документа у відповідь на інформаційний запит клієнта.

В процесі індексування в базі даних, для кожного інформаційного документа, що зберігається в системі бази даних, будується відповідний пошуковий образ. На сьогодні розглядаються два основні підходи в ІПС до побудови пошукового образу - приписуючи і виводі індексування. У першому варіанті в процесі індексування інформаційному документу ставлять у відповідність набір ключових слів з деякою класифікаційної системи ІПС, і наступним кроком інформаційний документ поміщається в загальну класифікацію. При використання другого варіанту з інформаційного документа вибирається набір ключових слів і призначаються пошуковим образом інформаційного документа, з яким в подальшому буде працювати інформаційно-пошукова система [6].

Великий вплив на те, як буде функціонувати інформаційно-пошукова система, залежить від типу пошукового образу, типу індексування а також вибраний варіант до оцінки релевантності.

Розглянемо основні моделі пошуку використовувані в інформаційно-пошукових системах

Модель пошуку, в основі якої лежить метод, з використанням класифікації. Як приклад такої моделі можна привести - тематичний каталог бібліотеки. Інформаційні документи такої інформаційно-пошукової системи розподіляються за темами, представленими у вигляді організованої ієрархічної класифікації, в даному випадку використовується перший варіант індексування. Відношення між класами використовується відношення включення. Інформаційні класи між собою

не перетинаються. Інформаційно-пошукові системи з такою організацією документів слід відзначити те, що список запитів до пошукової системи визначено заздалегідь у вигляді класифікатора тем і пошукова система не дозволяє клієнтам проводити пошук необхідної інформації по перетинанню класів і також не дозволяє впорядкування інформаційних документів за релевантністю.

У векторній моделі [6] документи $d \in D$ ідентифікуються в ІПС за допомогою множини незалежних між собою атрибутів *A*, наприклад ключових слів (в даному випадку використовується другий варіант типу індексування). Кожен інформаційний документ представляється у вигляді вектора атрибутів $d_i = \{a_{i1}, a_{i2}, ..., a_{in}\}$, де a_{ij} - представляє собою вагу *j*-го атрибута в інформаційному документі. Аналогічним чином, пошуковий запит $q = \{q_1, q_2, ..., q_t\}$, де q_j - представляє собою вагу *j*-го атрибута в пошуковому запиті. В даному випадку формальна релевантність визначається наступним чином, як скалярний добуток двох векторів d_i і q. До переваг векторної моделі можна віднести упорядкування інформаційних документів за релевантністю пошукового запиту. Найбільш використовуваною на практиці варіант векторної моделі, в якому формальна релевантність в даному варіанті обраховується за формулою TFIDF [6]. Для обчислення формальної релевантності використовується при цьому статистична інформація про всі інформаційні документи, які зберігаються в пошуковій системі.

В пошуковій системі, яка використовує булевську модель [10] пошуковий запит представляється у формі логіки висловлювань, в даному випадку в якості висловлювань в ІПС використовуються ключові слова, типу висловлювання істинно, або хибне. Висловлювання рахується істинно, якщо ключове слово пошукового запиту входить до складу інформаційного документа. В даному випадку критерієм видачі інформаційного документа є істинність заданого в пошуковому запиті ключового слова. До недоліків методу слід віднести неможливість упорядкувати результати пошукового запиту за релевантністю.

Для оцінки ефективності інформаційно-пошукових систем їхньої роботи, крім використовуваних стандартних параметрів, які задіяні для оцінки ефективності електронно-обчислювальних систем, також додатково використовуються спеціальні параметри для оцінки ефективності та якості роботи пошукових систем. Серед них основними є наступні параметри [7]: точність видачі інформації пошуковому запиту – визначається як відношення загального числа виданих релевантних інформаційних документів до загальної суми виданих релевантних і нерелевантних інформаційних документів; втрати інформації пошуку - відношення числа невиданих релевантних інформаційних документів до загальної суми виданих релевантних і невиданих релевантних інформаційних документів; повнота видачі запитаної інформації – визначається як відношення загальної суми виданих релевантних документів до загального числа виданих релевантних документів і невиданих релевантних документів; інформаційний шум – визначається як відношення виданих нерелевантних документів до загальної суми виданих релевантних і нерелевантних інформаційних документів; чутливість - відношення виданих релевантних інформаційних документів до загальної суми виданих релевантних і невиданих релевантних інформаційних документів; специфічність - відношення невиданих нерелевантних інформаційних документів до загальної суми виданих нерелевантних і невиданих нерелевантних інформаційних документів.

Як правило на практиці для порівняння пошукових систем використовуються усереднені графіки на яких представленні залежності параметра повноти від параметра точності [7].

Також для порівняння пошукових систем використовуються однозначні оцінки [4]. Як приклад можливо привести оцінку Е-мера [5], яка надає можливість уникнути порівняння пар параметрів повнота, точність за рахунок введення відношення значимості їх параметрів: $E(d) = 1 - \frac{\left(d^2 + 1.0\right) \cdot T \cdot P}{d^2 T + P}$, де T - точність, P - повнота, d – відношення значимості параметрів повноти і точності.

Розглянемо основні класи інформаційно-пошукових систем для пошуку інформації в базах даних.

Пошукова система, в основі якої використовується класифікації інформаційних документів. Загальний алгоритм роботи пошукової система, в основі якої використовується класифікації представлена на рис. 1.2.

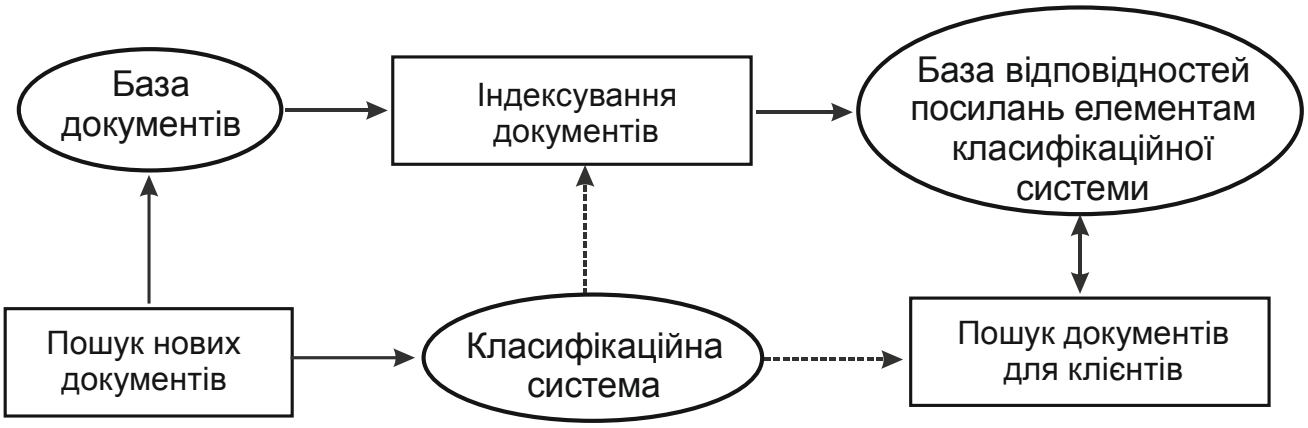


Рисунок 1.2 - Алгоритм роботи пошукової система, в основі якої класифікації

Головною відмінністю роботи пошукової система, в основі якої використовується класифікації є поява програмного блоку процесу пошуку нових інформаційних документів. Використання пошукової системи побудованої на класифікації ефективно в тому випадку, коли класифікаційна пошукова система побудована під вузьку предметну області. До недоліків класифікаційної пошукової системи можливо віднести: для якісного пошуку необхідно заносити всі інформаційні документи для їх подальшого індексування і зберігати їх в базі даних, що приведе до великого об’єму інформації, і високому навантаженні на мережу а також в необхідності постійно оновлювати в базі даних інформацію; пошук інформаційних документів клієнтів може здійснюватися тільки по використовуваній пошуковій системі відповідної класифікаційної системи.

Пошукова система, в основі якої пошук реалізується за ключовими словами, інформаційних документів. Загальний алгоритм роботи пошукової система, в основі якої пошук реалізується за ключовими словами представлена на рис. 1.3.

Пошуковий запит в інформаційних системах такого типу формується у вигляді рядка ключових слів, які повинні мати місце в документі пошуку.

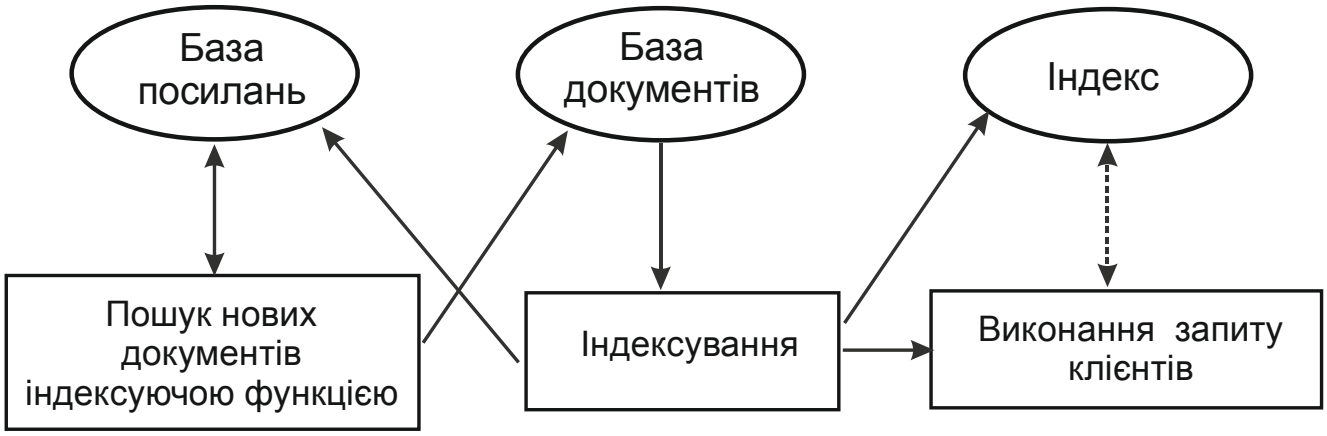


Рисунок 1.3 - Алгоритм роботи інформаційної системи пошуку за ключовими словами

Основними процесами в інформаційній системі пошуку за ключовими словами є пошук нових інформаційних документів індексуючою функцією, також індексування знайдених інформаційних документів і виконання пошукового запиту клієнта. Індексуюча функція представляє собою спеціальне програмне забезпечення (автономний процес), задача якої полягає в постійному, періодичному оновленні та поповненні бази даних інформаційних документів.

Індексуюча функція збирає інформаційні документи в базу документів, а посилання - в базу посилань. Періодично, індексуюча функція перевіряє зібрану в базі даних інформацію на коректність і цілісність. По інформаційним документам пошукова система будується індекс, який в подальшому використовується для ефективного пошуку документів за ключовими словами. В подальшому для економії дискового простору пошукова система зберігає короткий пошуковий образ.

До переваг інформаційних систем пошуку за ключовими словами систем є простота їх використання. До недоліків таких систем можна віднести: на пошуковий запит, клієнт отримує багато нерелевантною інформації, так як по списку ключових слів достатньо складно точно сформулювати інформаційні потреби клієнта, а так як клієнти надають найчастіше короткі запити, так як для формулювання розширеного запиту необхідність знання спеціального синтаксису, а у кожній пошуковій системі власний синтаксис; індексуючі функції сильно

завантажують мережу пошукової системи, інформацію в базі даних необхідно постійно оновлювати; можливість роботи клієнта пошукової системи тільки в інтерактивному режимі.

На сьогоднішній день проводяться дослідження, присвячені щодо підвищення ефективності пошукових систем за ключовими словами. Основна увага приділяється наступним напрямками розвитку: підвищення якості, отриманої інформації, оцінки релевантності документів; побудова моделі клієнта, з метою видачі більш релевантної інформації пошаткованої системою конкретному клієнту.

Розглянемо особливості роботи з інформаційними пошуковими системами. Використання інформаційних пошукових систем може стати причиною проникнення на комп'ютер клієнта шкідливої програми. Видаючи результати по пошуковому запиту клієнта, пошукові інформаційні системи можуть видавати адреси заражених сайтів. Також інформаційно-пошукові системи видають посилання на релевантні сайти, але при цьому не відповідають за достовірність інформації. Задача інформаційно-пошукових систем - максимально точно і швидко відповісти на пошуковий запит клієнта, тому виникає потреба не довіряти інформаційним документам, які видаються користувачу. Отримана інформація пошуку, можуть містити некоректну або відверто неправдиву інформацію, може ввести в оману клієнта, адже не всі джерела, перевіряються і пишуться компетентними фахівцями.

Розглянемо основні рекомендації щодо безпечного використання інформаційно-пошукових систем: виникає необхідність обов'язкового використання і оновлення антивірусних засобів – програмне забезпечення, яке дозволяє видаляти і виявляти шкідливі програми, також відновлювати заражені файли, запобігати проникненню вірусів на комп’ютер клієнта; виникає необхідність перевірки достовірності отриманої інформації пошукових систем.

Таким чином проведене дослідження надає опис основних функцій які використовуються для аналізу пошукового запиту інформаційно-пошуковою

системою: морфологічний пошук, логічні операції, цитатний пошук, пошук з корекцією помилок, використання синонімів, пошук з синтаксичним розбором слова змішаний цитатний пошук, пошук введеної фрази як цитати, але між введеними словами можуть бути присутніми інші слова і т.д. Деякі з перерахованих опцій мають свої специфічні налаштування. Крім того, є також можливість використання словника незначущих слів, причому в програмному забезпеченні є список цих слів, також можна здійснювати пошук пріоритетних слів по самостійно заповненому словнику.

1.2 Дослідження та класифікація систем управління базами даних

Система управління базами даних представляє програмно-апаратний комплекс, з використанням якого можна розробляти та проектувати бази даних, проводити над даними різні типи операцій: додавати, оновлювати, витягувати, видаляти, редагувати. Сучасні системи управління базами даних (СУБД) надають гарантії збереження, безпеку зберігання даних, їх цілісність, а також надає права доступу до адміністрування баз даних.

Переважна більшість призначених для клієнта інформаційно-пошукових систем розраховані на роботу з базами даних, в яких зберігається не тільки текстова інформація, але і музика, графіка, а також будь-який інший тип даних. Потреба у використанні баз даних зростає постійно, в зв’язку з цим з'явилася велика кількість сучасних СУБД, для роботи не потрібна додаткова спеціальна технічна оснащеність. Таким чином, актуальність тематики по СУБД зумовлена зростанням затребуваності сучасними інформаційно-пошуковими технологіями і базами даних.

СУБД також це інформаційне середовище для управління збереженими в базах даних відомостями про об'єкти реальної предметної області. В розділі розглянуті основні СУБД, недоліки та їх переваги, також проведений аналіз та класифікація сучасних СУБД, призначених для роботи і автоматизації роботи

організацій. На сьогоднішній день існує безліч систем управління базами даних, використовувані для роботи з різними пошуковими системами. Таким чином вибір конкретної СУБД залежить від задач ІПС та визначається декількома факторами, але головна - можливість роботи СУБД з побудованої моделлю даних. Таким чином з найважливіших характеристик СУБД є тип моделі (реляційна, об'єктно-орієнтована, пост - реляційна), яка підтримується конкретною СУБД. База даних для інформаційно-пошукової систем повинна бути реляційної, яка є на даному етапі найбільш поширеною. Крім використовуваної моделі СУБД є також важливим показником вартість ліцензії для проектування баз даних а також вартість підтримки в процесі експлуатації СУБД. Також необхідно звернути увагу на мінімальні технічні вимоги для успішної оперативної і ефективної роботи СУБД.

До базових задач СУБД належать створення, зміна та доступ до інформації/даних, яка зберігається в БД. При виборі СУБД необхідно враховувати - зручність використання, інтерфейс який надається СУБД, ступінь масштабування СУБД, а також можливості інтеграції з програмними продуктами, з якими СУБД буде контактувати. Крім наведених факторів, також має значення вартість систем управління та їх підтримка. Необхідно враховувати можливість розширення організації, відповідні механізми управління базами даних також повинні мати можливість покращуватися і вдосконалюватися. На сучасному етапі великі організації використовують різноманітні інформаційно – пошукові системи - офіційні сайти компаній, банківського обслуговування, електронні торгові майданчики, портали державних послуг, системи підприємств управління ресурсами і т. д.

Експертами компанії Positive Technologies в 2018р. були проведені дослідження по тисяча сто дев'яносто чотири web-додатків і клієнт-серверних систем з різних галузей діяльності підприємств. Атаки типу SQL-ін'єкції продовжують неперервна зростати. Так чином критично небезпечна вразливість SQL-ін'єкцій піднялася з шостого місця на четверте, тобто склала 62% проти 48%

в 2017 році. Лідером за кількістю знайдених вразливостей є банківська галузь (89%), на другому місці - телекомунікаційна галузь (80%). Далі йдуть промисловість (71%) і інформаційні технології. Останнє десятиріччя XX ст. характеризується широкою комп'ютеризацією всіх видів діяльності людства: від традиційних інтелектуальних задач наукового характеру до автоматизації виробничої, комерційної, торгової, банківської та інших видів людської діяльності. Таким чином в умовах ринкової економіки конкурентну боротьбу можуть успішно витримувати тільки ті підприємства, які у своїй застосовують у своїй діяльності використовують сучасні прогресивні інформаційні технології [3]. Використання інформаційних технологій в поєднанні з прогресивними технологіями матеріального виробництва, дозволяють істотно підвищувати продуктивність праці і якість продукції і в той же час значно скорочувати терміни постановки на виробництво нових виробів, що відповідають запитам і очікуванням споживачів. В першу чергу вище сказане відноситься до складної наукомісткої продукції.

Ведення будь-якої інформаційно-пошукової системи, і особливо у випадку коли пошукова система технічна, системи характеризується великою кількістю різноманітних накладами на них параметрів, які необхідно враховувати при проектуванні подібних пошукових систем.

Таким чином також важливо вибрати відповідну для досягнення поставленої ети не тільки відповідну СУБД, але при цьому і технологію яка дозволить успішно виконати таку складну поставлену задачу [5].

Базу даних можна представити як набір структурований даних. Дані, в даному випадку, організовані таким чином для моделювання відповідних аспектів реальності предметної області, а також для підтримки процесів, що вимагають надання необхідних даних [3].

Більшість сучасних систем управління базами даних засновані на реляційній моделі представленої інформації управління базами даних. Реляційна модель представлена таким чином, що кожен запис відповідної таблиці містить

інформацію, яка відноситься тільки конкретного об'єкту. У реляційних базах представлені дані не дублюються, а зв'язуються по відповідних полях з іншими таблицями представлення даних по даному об'єкту. В СУБД серед інших функцій виділимо три основні: - визначення даних, надана можливість визначення даних відповідно до предметної області яка представлена для автоматизації, тип інформації яка буде зберігатися у базі даних, при цьому задати структуру запису, тип даних які входять в запис, а також вказати, тип реляційного зв’язку між таблицями бази даних, як дані по даному об'єкту пов'язані між собою. Також є можливість задати формати, обмеження, критерії перевірки представлених даних в БД; обробка даних, дані можна об'єднувати з пов’язаними даними з інших таблиць для витягу повної інформації по даному об'єкту, а також роботи відповідні обчислення, отримувати підсумкові значення; виконувати управління даними, надавати відповідний доступ до даних БД, кому дозволено перегляд даних, корегування, видалення та додавання нових записів в БД, також при цьому задаються правила колективного доступу користування даними БД.

Серед сучасних СУБД можна виділити Oracle, Microsoft SQL Server, IBM DB2, Microsoft Access, Postgre SQL, MySQL, і SQLite. СУБД в своїй основі використовують деякі стандарти для взаємодії, як SQL і ODBC, OLE DB, для підтримки проектів які використовують в собі декілька баз даних. Системи управління базами даних повинна забезпечувати ефективну колективну роботу з даними, забезпечувати правильність відповідну продуктивність, доступність та безпеку і конфіденційність даних, надавати користувачам у міру необхідності відповідні дані.

Класифікація БД може включати тип вмісту інформації: документ тексту, мультимедійні або статистичні об'єкти, області застосування баз даних, автоматизація підприємств, бухгалтерський облік, банківська справа, музичні композиції, виробництво, фільми, або страхування.

На даному етапі при проектуванні інформаційно-пошукових систем систем можна використовувати різні типи систем управління базами даних, які

розрізняються вимогами до технічних та обчислювальних ресурсів та їх можливостями для успішного функціонування системи. Систем управління базами даних можна рознести на два класи: персональні (режим одно користувальницького використання) і СУБД багато користувальницького використання, розраховані на одночасному обслуговуванні багато користувачів. СУБД dBASE, MS Access, FoxPro відносяться до першого класу, і орієнтовані на роботу на персональному комп'ютері (ПК). Спочатку СУБД першого класу підтримували роботу з базами даними одного клієнта. СУБД першого класу виконується як одна програма, при цьому таблиці такої бази даних представляються окремими розрізненими файлами на дисковому просторі ПК. В зв’язку з розвитком локальних мереж, перед розробниками першого класу СУБД виникла необхідність пристосовувати даний клас СУБД до роботи в локальній мережі, в якій стало необхідним організувати доступ до бази даних з декількох ПК, включених в мережеве середовище. При цьому відповідні файли бази даних розміщуються на так званому файловому сервері. Виникає необхідність в розміщенні на всіх робочих місцях копії програми-СУБД а також прикладної програми, при цьому характеристики ПК робочого місця можуть істотно впливати на характеристики їх виконання. Помилка у виконанні однієї з копії є буде помічена іншими копіями. При виконанні запитів до БД СУБД може або виробляти пошук даних у віддалених файлах на файловому сервері, або копіювати всі файли, в яких ведеться пошук в свою локальну файлову систему. У першому випадку виникають проблеми одночасного доступу до даних при їх зміні. Дані, над якими виробляються зміни, повинні бути заблоковані. Засоби файлового серверу дозволяють виконувати блокування на рівні файлів, але не на рівні записів, що істотно знижує ефективність паралельної роботи з базою даних багатьох користувачів. У другому ж випадку, по-перше, потрібна передача по мережі великих обсягів інформації, а по-друге, виходить, що різні робочі місця працюють з різними копіями даних і ці копії можуть стати неідентичних [7].

Розглянемо роботу та особливості СУБД другого класу. Даний клас СУБД створювалися для виконання роботи з використанням на великих комп'ютерах і при цьому для забезпечення паралельної роботи кінцевих користувачів. В архітектуру систем керування базами даних другого класу входить ядро, яке постійно знаходиться в пам'яті сервера системи, а також включає велику кількість програм-агентів, задача яких обслуговування пошукових запитів кінцевих клієнтів і локальних прикладних програм. При використанні СУБД другого класу ядро системи керування базами даних і сама база даних повинні розміщуватися на одному персональному комп'ютері (сервері). В даному випадку використовується одна копія СУБД, яка управляє при цьому однією копією бази даних. Таким чином єдина керуюча система управління базами даних дозволяє в даному випадку ефективно організувати паралельний доступ до бази даних багатьох користувачів, а також розв’язати своєчасно у разі виникнення між ними конфліктів. При виникненні помилки в роботі СУБД другого класу, вона своєчасно буде локалізована і ефективно виправлена автоматично самою СУБД. При використанні СУБД другого класу в умовах роботи в мережі в даному випадку ядро системи управління базами даних буде виконувати пошукові запити клієнтів на вибірку відповідних даних і при цьому тільки результати вибірки будуть передаватися по мережі. Оскільки швидкодія сучасних використовуваних дискових спеціалізованих систем значно вища, ніж при цьому швидкість передачі відповідних даних по мережі, тому зменшення обсягу переданих результатів пошукового запиту істотно збільшує ефективність роботи даного класу систем. При цьому не накладається ніяких обмежень на масштаб мережі, агенти можуть бути пов'язані з ядром СУБД через будь-яку мережу і будь-які протоколи передачі даних. Розраховані на паралельну роботу кінцевих агентів СУБД другого класу також мають незаперечні переваги в таких аспектах, як безпека, надійність, доступність також цілісність збережених даних в базі даних. СУБД другого класу розраховані на паралельну роботу кінцевих агентів системи управління базами даних з самого початку проектування використовували в даному в якості

інтерфейсу мову запитів SQL, звідси їх одна альтернативних назв - SQL-сервери. Останнім часом мова запитів SQL стає доступною і в персональних СУБД першого класу, але при цьому вони не включають засоби забезпечення паралельного доступу та безпеки, цілісності даних які зберігаються в базі даних. СУБД першого класу не в стані забезпечити роботу даних механізмів.

При виборі відповідної бази даних при проектуванні інформаційно-пошукової системи важливо використати БД, яка найбільшою мірою відповідає висунутим вимогам до ІПС вимогам, в даному випадку стає питання визначити, яка модель буде використовуватися при автоматизації предметної області (автоматизація технологічних процесів, документообігу). При виборі системи керування базами даних, першу чергу необхідно керуватися значеннями наступних факторів: максимальне число кінцевих агентів, для паралельної роботи з базою даних; при цьому необхідно також враховувати характеристики клієнтського програмного забезпечення; програмно-апаратні механізми та компоненти сервера баз даних; програмно-апаратне забезпечення серверної операційної системи; рівень кваліфікації обслуговуючого персоналу.

Розглянемо особливості та способи використання ORD (об'єктно-реляційних баз даних, а також (об'єктно-реляційна система управління базами даних), ОРСУБД в деякій мірі схожа на реляційну систему управління базами даних, але з об'єктно-орієнтованої модель бази даних має свої особливості: об'єкти, класи, наслідування, множинне наслідування – перераховані механізми безпосередньо підтримується ORD в схемах баз даних і також SQL мовою запитів. Крім того, наряду з реляційними системами, ОРСУБД підтримує розширення моделі даних з використанням відповідних користувацьких типів даних і методів забезпечення роботи з даними.

Об'єктно-реляційна модель представлення даних в базі даних займає проміжне місце між реляційними системами баз даних і ООСУБД (об'єктно-орієнтованими базами даних) забезпечують золоту середину між ними. В системах з об'єктно-реляційним представленням даних в базі даних, використовується по

суті такий підхід, що і з реляційним представленням даних в базі даних: дані зберігаються відповідним чином в базі даних, є можливість маніпулювати паралельно даними запитами в SQL мові запитів, в той час ООСУБД знаходиться на іншому полюсі, в якому використовувана база даних є постійним сховищем об'єктів предметної області програмного забезпечення написаної на одній з мов об'єктно-орієнтованого програмування, з використанням програмування API для зберігання, редагування та видалення об'єктів, що представляють задану предметну область і також коректну підтримку пошукових запитів кінцевих агентів. [15].

Сучасний етап розвитку людства не представляється можливим без ефективного методу об'єднання існуючих інформаційних ресурсів та інформаційно-пошукових систем забезпечення надійного прогресивного розвитку всіх галузей індустрії. Важливою категорією використовуваних систем управління базами даних, в залежності від яких в великій степені визначається ефективність роботи будь-якого підприємства, компанії або організації.

Наведемо основні приклади використовуваних на даному етапі різних типів баз даних представлення даних. Більшості з них в дослідженнях було приділено особливу увагу, на вимогу кінцевих клієнтів. Деякі з типів баз даних існують як в спеціалізованих СУБД, при цьому функціональність деяких з них включена в сучасні існуючі системи управління базами даних загального призначення [13]: активна база даних представлення інформації – представляє базу даних, особливістю якої є включення обробки подій в архітектуру системи, які реагують на події які виникають як всередині, так і поза даної бази даних представлення інформації. Можливі області застосування активних баз даних представлення інформації включають моніторинг безпеки, збору статистики, оповіщення а також авторизації. Більшість використовуваних сучасних на даному етапі реляційних баз даних включають активні спеціальні функції реагування на обробку виниклих подій бази даних у вигляді тригерів БД; хмарні бази даних збереження інформації – представляють бази даних, які опираються в процесі роботи на використання

хмарних технології. Хмарна база даних збереження інформації і більша частина системи управління базами даних знаходяться віддалено, в так називаємій «в хмарі», а їх використання кінцевими агентами відбувається в даному випадку через Веббраузер і відкриті API; інформаційне сховище збереження даних - сховища різних архівних даних з оперативних робочих баз даних і часто отримуванні даних від зовнішніх джерел, таких як в вчасності від фірм маркетингових досліджень. Оперативні дані при передачі їх в інформаційне сховище трансформуються відповідним чином, таким чином інформаційне сховище стає центральним джерелом зберігання даних для використання керівниками підрозділів та іншими кінцевими агентами, які не мають прямого доступу до оперативних даних. Операції в які виконуються над даними в інформаційному сховищі, як правило, пов'язані з об'ємною обробкою збережених даних і, в даному випадку, це виявляється не достатньо зручно і неефективно; розподілена база даних збереження інформації, розподілені бази даних зазвичай відносяться до модульної архітектури систем управління базами даних, даний підхід надає можливості інтегрувати співпрацювання з однієї СУБД, в той же час управління базою даних в даному випадку розподілено і функціонує з використанням декількох ПК і використанням різних об'єктів; документо-орієнтовані бази даних - представляють собою спеціальне програмне забезпечення, даний тип баз даних призначений для зберігання, управління і пошуку документоорієнтованої (слабоструктурованих даних) інформації. Документо-орієнтовані бази даних представляють одну з основних категорій баз даних, так званих NoSQL. Популярність використовуваного терміна «документоорієнтованих баз даних» («сховище документів») найбільш виросла з використанням терміну NoSQL. Використання документо-орієнтованих баз даних представляться зручним при роботі з документами і виконання відповідних операцій зберігання, редагування, управління і витягування відповідних документів; вбудована база даних - «прихована» від кінцевого користувача і практично в даному випадку не вимагає поточного обслуговування. Вбудована

база даних - широка категорія, яка включає в себе сучасні технології систем управління базами даних з різними цільовими ринками і властивостями. Вбудована база даних - підмножина вбудованих засобів баз даних представлення інформації використовується при експлуатації вбудованих систем в режимі реального часу, таких як пристрої електроніки та телекомунікаційні перемикачі; бази даних кінцевих користувачів - бази даних даного типу зберігають дані, розробленими кінцевими користувачами. Прикладами баз даних кінцевих користувачів - електронні таблиці, збірники документів, презентацій, мультимедійних та інших типів файлів; федеративні бази даних (об'єднання баз даних) представляють собою інтегровану базу даних, яка включає різнотипні бази даних, зазвичай кожна зі своєю системою управління базами даних. Федеративна база даних використовується і обробляється як єдина БД об'єднана по базам даних системою управління (FDBMS), система дозволяє прозоро інтегрувати кілька автономних систем управління базами даних, при цьому є можливість об’єднання, різних типів СУБД (гетерогенна баз даних). Власники відповідних баз даних забезпечують з’єднання між собою використовуючи мережу, і при цьому можуть бути географічно децентралізовані. Використання федеративної бази даних потребує в даному випадку використання проміжного розподілу загальних секретних ключів, який потребує використання атомного протоколу фіксації, може бути використаний двофазний протокол фіксації, який забезпечує відповідний доступ до інформації в гетерогенних базах даних; граф бази даних- це свого роду NoSQL БД, яка в своїй архітектурі використовує графові структури з вузлами, ребрами і властивостями листків для надання, редагування та зберігання інформації в базі даних документів. Граф (головний вузол) даного типу баз даних, може зберігати будь-який граф представлення даних, може також відрізняється від спеціалізованих систем баз даних, таких як ієрархічних та мережевих баз даних; гіпермедіа бази даних дозволяють переглядати інформацію в базі даних, паралельно з виконанням пошукових операцій та використання спеціального програмного забезпечення гарантують еквівалент індексу використовуваної бази

даних для підтримки пошуку та здійснення одночасно інших видів діяльності; гіпертекст бази даних. У базі будь-яке слово або фрагмент тексту, що представляє об'єкт, наприклад іншу частину тексту, статті, фотографії або фільму, може бути пов'язаний з цим об'єктом; гіпертекстові бази даних, використання яких особливо корисне для організації та упорядкування великих обсягів розрізненої інформації; In-Memory Database - бази даних збереження інформації в пам'яті представляють базу даних, інформація яких представлена в оперативній пам'яті, але зазвичай паралельно зберігається резервна копія представленої інформації в енергонезалежному сховищі даних. Такий тип баз даних використовуються, в випадку коли критичним є час відгуку; бази знань - особливий вид бази даних які використовуються для управління знаннями, також даний тип надає засоби та механізми для збору, видалення та організації інформації в вигляді правил; оперативні бази даних -зберігають всі необхідні дані про діяльність організації. По великому рахунку кожна велика бізнес компанія використовує бази даних такого типу(бази даних клієнтів); паралельні бази даних. Мета використання паралельних бази даних підвищити продуктивність, час відгуку за рахунок розпаралелювання задач, серед яких: підключення/відключення кінцевих агентів, завантаження даних, побудова індексів, виконання та оптимізація пошукових запитів. Паралельні бази даних підвищують ефективність обробки і введення/виведення результатів запиту за рахунок використання паралельної одночасної роботи декількох центральних процесорів (CPU) (використання багатоядерних процесорів) і також використання спеціальних методів паралельного зберігання інформації. При використанні паралельній механізму обробки операції виконуються одночасно, на відміну від обробки інформації послідовно; системи без поділу - масова паралельна обробка В даному випадку під кожен процесор відводиться своя оперативну та дискову пам'ять. При використанні даного типу база даних інформація розподілена між усіма дисковими пристроями. Така архітектура, таким чином забезпечує більш високий рівень масштабованості, оптимальну продуктивність, в разі коли дані бази даних будуть знаходитися на

одному кластері. Розподіленість потоку даних знижує продуктивність; паралельні системи управління базами даних використовують в своєму розпорядженні велику кількість сучасних допоміжних технологій, які в свою чергу дозволяють підвищити ефективність та оптимізувати обробку складних запитів за рахунок використання при цьому технології розпаралелювання, таких операцій, сортування, вибірка, з'єднання, і т.д. При виборі системи управління базами даних необхідно пам’ятати про безпеку при зберіганні даних в базі даних. При цьому необхідно притримуватися наступним умовам безпеки при проектуванні бази: контроль доступу, шифрування, представлення інформації адекватною структурою даних.

1.3 Дослідження алгоритмів пошуку інформації

Цифрові матеріали включають зростаючі текстові документи, структуровані і неструктуровані зображення, бази даних, звукові і графічні матеріали, програмне забезпечення та веб-сторінки. Збільшення темпів створення цифрової інформації привело до необхідності аналізу структури вхідних файлів і більш швидкого пошуку і обробки інформації. З цією метою, досліджені властивості алгоритмів штучного інтелекту в аналізі нетрадиційно структурованих великих даних. Було встановлено, що необхідно використовувати алгоритми на основі штучного інтелекту для вирішення проблем, пов'язаних з поліпшенням якості пошуку, збільшенням обсягу даних і інтенсивності призначених для користувача запитів. Також аналізуються алгоритми пошуку, їх недоліки та можливі варіанти використання для їх застосування з метою максимізації їх переваг.

Класичні моделі використовувані для опису інформаційного пошуку розглядають інформаційні документи як множину ключових слів (терми, словоформи), для представлення цих документів. Терм – семантика, з використанням якої описується основний зміст інформаційного документа. Формально опис моделі представлення інформаційного пошуку складається з

наступних частин $M_{ФО} = \langle D, Q, F, R(q, d_i) \rangle$ де, D- множина використовуваних різних типів, для представлення інформаційних документів; Q - множина використовуваних типів для опису інформаційних потреб кінцевого агента, пошукових запитів; F - загального каркаса, область в рамках якої буде відбувається загальне моделювання описів документів і пошукових запитів, а також опис можливих взаємозв'язків між ними; $R(q, d_i)$ - функція ранжирування, відповідній парі «і-й документ (d_i) – пошуковий запит (q)» повертає деяке ціле число.

Серед моделей інформаційного пошуку документів можливо виділити три класи: теоретико-множинні моделі інформаційного пошуку, в основі яких лежить використання теорію множин. Класичний приклад представлення моделей цього класу є булева модель, в рамках якої інформаційні документи і пошукові запити представляються у вигляді множин термів; імовірнісні моделі. Каркасом в основі таких моделей використовується теорія ймовірностей, результатом оцінки релевантності інформаційного документа пошуковому запиту кінцевого агента береться ймовірність того, чи кінцевий агент визнає інформаційний документ істинно релевантним; алгебраїчні моделі. У випадку використання цих моделей інформаційні документи і пошукові запити описуються в даному випадку у вигляді алгебраїчних векторів в багатовимірному просторі. В основі опису таких моделей використовуються алгебраїчні методи.

Необхідно відмітити що серед розглянутих моделей, набули найбільшого поширення алгебраїчні моделі, оскільки їх використання в практичних додатках більш ефективне від інших розглянутих моделей. В розглянутих публікаціях станом на сьогодні пропонуються нові моделі для опису інформаційного пошуку, але по своїй природі вони часто є гібридними і мають властивості розглянутих моделей з різних класів. Необхідно також зазначити, що практично всі сучасні алгоритми пошуку інформаційних документів в своїй основі використовують теорію графів. Інформаційний пошук не структурованої інформації можна

представити графом, в якому сторінки є вузлами, а при цьому посилання на інші сторінки - дугами.

Деякі інформаційно - пошукові системи для підвищення ефективності пошуку документів використовують морфологічний аналіз, який дозволяє при цьому здійснювати пошук документів більш якісно.

Розглянемо класичний алгоритм морфологічного аналізу текстів.

1. Інформаційний пошук всіх варіантів аналізованої словоформи, яка піддається аналізу.

2. Для кожного варіанту знайденої словоформи (основи), відсортованої по спаданню довжин основи, починаючи з найдовшої словоформи, здійснюється бінарний пошук в інверсному списку основ. У випадку якщо варіант основи словоформи в наведеному списку відсутній, то необхідно відшукати найбільш близькі словникові словоформи, які мають в даному випадку максимальне по довжині загальне закінчення. Позиція першої найближчої словоформи (основи) і міра її подібності - число символів які виявилися однаковими в основі і також довжина закінчення - запам'ятовуються для подальшого аналізу.

3. Для всіх варіантів словоформ (основ) проводяться наступні обчислення: для всіх визначених лексем, інформаційного документа які мають в основі однакову міру подібності (в нашому варіанті довжина загального закінчення словоформи основи), проводимо морфологічний аналіз по вибраній лексемі; якщо розглянутий варіант словоформи (основи) не співпадає ні з однією з найближчих словникових словоформ основ, то можливо зробити наступний висновок, що аналізоване слово в даному варіанті з даними варіантом словоформи основи в словнику відсутнє. У цьому випадку за спрощеним варіантом основи, закінчення і лексеми, відповідної найближчій словникової основі, генерується гіпотетична лексема - модель словозміни для цього невідомого слова. У разі успішної генерації ця гіпотеза подається на вхід морфологічного аналізатора по лексемі; успішні варіанти розбору запам'ятовуються у вигляді {лексема (текст статті), варіанти розбору}; якщо результат є гіпотезою і при цьому така ж гіпотеза вже є, то вона не

запам'ятовується повторно. Замість цього збільшується лічильник продуктивності цієї гіпотези; якщо серед лексем з однаковою поточної мірою подібності є хоча б один варіант розбору, то перехід до п. 5 з успішним результатом. Якщо варіантів розбору немає, то довжина необхідного загального закінчення словоформи основи зменшується. Якщо після циклічного зменшення довжина необхідного загального закінчення словоформи основи стала менше двох, то перехід до п. 5 з відмовою; інакше - перехід до п. 3.

4. Проводиться уніфікація гіпотез по парадигмам (оскільки формат при використанні даного алгоритму допускає неоднозначний опис парадигми) і також проводиться фільтрація гіпотез по продуктивності. Якщо продуктивність розглянутої гіпотези менше максимальної продуктивності гіпотези в п'ять разів, то гіпотеза, яка розглядається в даному варіанті відсіюється.

5. Кінець алгоритму морфологічного аналізу текстів.

Крім використання так званих морфологічних словників в інформаційно - пошукових системах для підвищення релевантності пошуковому запиту кінцевих агентів по вузькоспеціалізованим темам також в такому випадку дуже часто застосовуються тематичні тезауруси (словники).

Розглянуті алгоритми пошуку документів відрізняються від класичних використовуваних алгоритмів пошуку інформації. Це обумовлено також тим фактом, що використовувані класичні моделі пошуку документів розробляються виходячи з передумов, поставлених та обумовлених пошуком релевантних інформаційних документів в Інтернеті. В результаті такого підходу виникає ряд проблем, при пошуку інформаційних документів пов'язаних з ранжируванням результатів отриманої інформації про посилання на інформаційні документи. Необхідно також відзначити, що представлені вище моделі відрізняються ще й тим, що пошук здійснюється одночасно по декільком сегментам потоку інформації, що в даному випадку накладає додаткові обмеження на використання існуючих моделей опису інформаційного пошуку документів.

Використання інтерактивного алогоритму пошуку [12] в інформаційно-пошукових системах забезпечує більш гнучку, в порівнянні з використовуваними іншими методами пошуку затребуваної інформації, реакцію на запити і потреби клієнта. В основі алгоритму інтерактивного пошуку лежить так званий зворотній зв'язок за релевантністю - метод інтерактивного пошуку інформації клієнтом в пошуковій системі, при якому інформаційно-пошукова система надає клієнтові початкові результати обробки пошукового запиту, а клієнт в свою чергу вказує які з виданих інформаційних документів релевантні, а які необхідно відкинути, після чого пошукова система піддає запит корегуванню і продовжує більш детальний пошук. Алгоритм інтерактивний пошуку будується як правило на основі векторної моделі. У випадку пошуку документів висококваліфікованим клієнтом пошукова система здатна надати клієнтові значно кращі результати пошуку, ніж автономний пошук документів.

Результати проведеного дослідження та аналізу параметрів таких як ефективність, швидкодія та об’єм дискового простору необхідний для виконання поставленої задачі використовуваних алгоритмів в інформаційно- пошукових системах показано в табл. 1.1.

Алгоритм створення ефективного запиту виглядає наступним чином: необхідно точно сформувати задачу пошуку. Для отримання корисної необхідної інформації, в першу чергу, потрібно зрозуміти, на яке саме питання необхідно відповідь; в другу чергу необхідно обмежити область пошуку. Отримання результатів пошукового запиту може відрізнятися в залежності від використання інформаційно- пошукової системи; виникає необхідність в правильному підборі ключових слів, тобто слова і фрази, які відносяться до теми інформаційного пошуку. Ключові слова при цьому поділяють ділять на високо-, середньо- і низькочастотні, це залежить від частоти запиту і визначається на основі статистики інформаційно – пошукової системи; для використання більш ефективного інформаційного пошуку виникає необхідність в використанні мови запитів.

Таблиця 1.1 – Порівняльний аналіз пошукових алгоритмів

Алгоритм	Ефективність	Швидкість	Розмір дискового простору	Підсумок
розширеної вибірки	висока	висока тільки при розмірі словника до500 тис. записів	низький	не підходить, в зв'язку з низь-кою швидкістю на словниках від 5 млн. записів
n-грамм	висока	середня, ліній-но залежить від довжини рядків можливо збільшити з допомогою хешування і індексування	високий	підходить
дерева пошуку	низька	висока	середній	підходить як спосіб індексування
відстань між рядками	низька	вище середнього	низький	підходить як спосіб сортуван-ня, ранжируван-ня результатів ін-шого алгоритму
хешування по-сигнатурі	нижче середнього	вище середнього	середній	підходить
послідовний перебір	висока тільки при пошуку з малою кількістю помилок по великому масиву тексту або при порів - нянні на повну відповідність	вище середнього	низький	підходить

На даний час розроблено безліч моделей і алгоритмів інформаційного пошуку документів, які в свою чергу використовуються інформаційно - пошуковими системами.

1.4 Постановка задачі

Таким чином на основі проведеного аналізу та дослідження отримані наступні результати: алгоритм послідовного перебору поступаються в швидкості обробки інформації словниковим алгоритмам, які по результатам дослідження є ефективнішими, але при порівнянні алгоритмів при роботі з рядками пошуку довжина яких перевищує 30 символів алгоритм послідовного перебору істотно ефективніший і значно виграє в швидкості. Також, серед розглянутих алгоритмів, слід приділити увагу сигнатурним алгоритмам а також алгоритмам n-грам, trie-дерев. Дані алгоритми забезпечують непогане співвідношення між швидкістю та ефективністю пошуку та розміром індексу.

Для досягнення мети, поставленої в дипломній роботі, необхідно вирішити наступні задачі:

1. Розробити модель опису процесу ідентифікації об'єктів в базах даних.
2. Розробити модель представлення релевантності подібності рядків.
3. Розробити алгоритм оптимізації запису інформації в бази даних.
4. Розробити алгоритм ідентифікації особистості в базах даних.
5. Розробити алгоритм пошуку інформації в базах даних за реквізитами особистості.
6. Провести дослідження ефективності запропонованих алгоритмів.

2 РОЗРОБКА МОДЕЛЕЙ ОПИСУ ПРОЦЕСУ ІДЕНТИФІКАЦІЇ ОБ'ЄКТІВ В БАЗАХ ДАНИХ

2.1 Проведення дослідження порівняльних характеристик сучасних систем управління базами даних

Незважаючи на те, що всі сучасні системи управління базами даних виконують одну і ту ж задачу - надавати кінцевим агентам можливість редагувати, створювати та отримувати доступ до даних, що зберігається в базах даних. Функції і можливості кожної системи управління базами даних можуть значно відрізнятися. При порівнянні затребуваних на сьогодні СУБД, необхідно враховувати, зручність інтерфейсу, масштабованість та інтеграція з іншими програмними продуктами, які вже використовуються в організації. Також, під час вибору СУБД необхідно брати до уваги вартість системи та можливості технічної підтримки, яка надається розробником.

На сьогодні існує декілька затребуваних системи управління базами даних, як платних, так і безкоштовних, які можна рекомендувати для використання на промисловому підприємстві та різних організаціях. Розглянемо деякі СУБД.

Oracle 12c. Існує кілька версій даного програмного продукту для задоволення потреб конкретної організації. Актуальна версія Oracle на даний час - 12c – основне призначення використання в хмарних середовищах і може бути розміщена на одному або декількох серверах, це дозволяє управляти базами даних, які містять мільярди і більше записів. Деякі з функцій новітньої версії Oracle включають в себе grid framework і використання як фізичних, так і логічних структур. Це означає, що фізичне управління даними не впливає на доступ до логічних структур. Крім того, безпека в даній версії доведена до найвищого рівня, так як робота кожної транзакції ізольована від інших.

До переваг СУБД Oracle 12c слід віднести: найсвіжіші інновації та вражаючий функціонал впроваджені в даному продукті, оскільки компанія Oracle

тримає планку в порівнянні з іншими розробників СУБД; СУБД від Oracle є високонадійною, фактично це еталон надійності серед подібних систем.

До недоліків СУБД Oracle 12c слід віднести: вартість Oracle може виявитися непомірно високою, особливо для невеликих організацій; система може зажадати значних ресурсів вже відразу після установки, тому можливо буде потрібно модернізувати обладнання для впровадження Oracle.

Ідеально підходить для великих організацій, які працюють з величезними базами даних і різноманітними функціями.

MySQL - одна з найпопулярніших баз даних для розробки веб-додатків. Фактично, є стандартом для веб-серверів, які працюють під управлінням операційної системи Linx. MySQL - безкоштовний пакет програмного забезпечення, обновлення MySQL виходять постійно, розширюючи функціонал і покращуючи при цьому безпеку. Також існують спеціальні платні версії, призначені для комерційного використання. У безкоштовному пакеті перевага надається на швидкість і надійність, а не на повноту функціоналу, який може стати одночасно і перевагою так і недоліком - в залежності від області застосування.

MySQL поширюється як під GNU General Public License, так і під власною комерційною ліцензією. Крім цього, розробники створюють функціональність за замовленням ліцензійних користувачів. Саме завдяки такому замовленню майже в перших версіях з'явився механізм реплікації. MySQL дозволяє вибирати різні движки для системи зберігання, які дозволяють змінювати функціонал інструменту і виконувати обробку даних, які зберігаються в різних типах таблиць. Завдяки відкритій архітектурі і GPL-ліцензуванню, в СУБД MySQL постійно з'являються нові типи таблиць. СУБД має простий у використанні інтерфейс, і пакетні команди, які дозволяють зручно обробляти величезні обсяги даних. Система неймовірно надійна і не вимагає підвищених вимог до апаратних ресурсів.

До переваг СУБД MySQL слід віднести: розповсюджується безкоштовно; прекрасно документована; пропонує багато функцій, також у безкоштовній версії; пакет MySQL включений в стандартні репозиторії найбільш поширених дистрибутивів операційної системи Linux, що дозволяє встановлювати її достатньо просто; підтримує набір призначених для користувача інтерфейсів; може працювати з іншими базами даних, включаючи DB2 і Oracle.

До недоліків СУБД MySQL слід віднести: доведеться витратити багато часу і зусиль, щоб змусити MySQL виконувати нескладні завдання, хоча інші системи роблять це автоматично, наприклад: створювати інкрементні резервні копії; відсутня вбудована підтримка XML чи OLAP; для безкоштовної версії доступна тільки платна підтримка.

Ідеально підходить для організацій, яким необхідний надійний безкоштовний інструмент управління базами даних.

Microsoft SQL сервер - програмний продукт управління базами даних, движок якої працює на хмарних серверах, а також локальних серверах, причому можна комбінувати типи застосовуваних серверів одночасно. Microsoft адаптувала SQL сервер для операційної системи Linux. До особливостей Microsoft SQL сервер є temporal data support, яка дозволяє відстежувати зміни даних з протягом часу. Microsoft SQL-сервер підтримує dynamic data masking (динамічне маскування даних), яка гарантує, що тільки авторизовані користувачі будуть бачити конфіденційні дані.

До переваг СУБД Microsoft SQL сервера слід віднести: продукт простий у використанні; поточна версія працює швидко і стабільно; движок надає можливість регулювати і відслідковувати рівні продуктивності, які допомагають знизити використання ресурсів; є можливість отримати доступ до візуалізації на мобільних пристроях; він добре взаємодіє з іншими продуктами Microsoft.

До недоліків СУБД Microsoft SQL сервера слід віднести: ціна для юридичних осіб виявляється неприйнятною для більшості організацій; навіть при ретельному

налаштування продуктивності SQL Server здатний використати всі доступні ресурси.

Ідеально підходить для: великих організацій, які вже використовують ряд продуктів Microsoft.

PostgreSQL є безкоштовних затребуваним варіантів СУБД, в більшості використовується для ведення баз даних веб-сайтів. В даний час PostgreSQL добре розвинена, і дозволяє користувачам управляти як структурованими, так і неструктурованими даними. Може бути використаний на більшості основних платформ, включаючи Linux. Прекрасно справляється з завданнями імпорту інформації з інших типів баз даних за допомогою власного інструментарію. Движок БД може бути розміщений в ряді середовищ, в тому числі віртуальних, фізичних і хмарних.

До преваг СУБД PostgreSQL сервера слід віднести: є масштабованою і здатна обробляти терабайти даних; підтримує формат json.

До недоліків СУБД PostgreSQL сервера слід віднести: документація досить туманна; конфігурація може збентежити непідготовленого користувача; швидкість роботи може падати під час проведення пакетних операцій або виконання запитів читання.

Ідеально підходить для організацій з обмеженим бюджетом, але кваліфікованими фахівцями.

DB2 має можливості NoSQL, і може читати JSON і XML-файли. Система DB2 розроблялася для серверів компанії IBM модельного ряду iSeries, працює на Windows, Linux і Unix. Діалект мови SQL, що використовується в DB2 строго декларативний, система забезпечена багатофазовий оптимізатором, які будують за цими декларативним конструкціям план виконання запиту. Відсутня мова збережених процедур, і, таким чином, все направлено на підтримку декларативного стилю написання запитів. Мова SQL DB2 при цьому є обчислювальна повною. Оптимізатор DB2 широко використовує статистику розподілу даних в таблицях, тому один і той же запит на мові SQL може бути

відкомпільований в абсолютно різні плани виконання в залежності від статистичних характеристик даних, які він обробляє. В рамках концепції підвищення рівня інтеграції засобів безпеки в комп'ютерній системі, DB2 не має власних засобів аутентифікації кінцевих агентів, інтегруючись із засобами операційної системи або спеціалізованими серверами безпеки. В рамках DB2 здійснюється тільки авторизація користувачів. DB2 є єдиною реляційної СУБД загального призначення, що має реалізації на апаратно-програмному рівні. Сучасні версії DB2 забезпечують розширену підтримку використання даних в форматі XML, в тому числі операції з окремими елементами документів XML. Остання версія DB2 також забезпечує вдосконалені функції аварійного відновлення, сумісності та аналітики.

До преваг СУБД DB2 слід віднести: Blu Acceleration дозволяє грамотно задіяти ресурси для об'ємних баз даних; може використовуватися в хмарному сховищі, на фізичному сервері; декілька задач можуть виконуватися одночасно за допомогою використання планувальника задач; коди помилок і коди завершення дозволяють легко відстежити, які завдання виконуються або виконалися за допомогою планувальника завдань.

До недоліків СУБД DB2 слід віднести: ціна за межами бюджету багатьох фізичних осіб і невеликих організацій; базова підтримка доступна тільки протягом трьох років.

Підходить для великих організацій, які планують задіяти максимум з наявних ресурсів і для оброки великих баз даних.

Якщо привести порівняння сучасних систем управління базами даних з точки зору їх поширеності, то можна бачити картину станом на 2015 рік (рис. 2.1).

На основі проведеного дослідження робимо наступний висновок що при розробці та плануванні веб-сервера в першу чергу необхідно звернути увагу на систему управління базами даних MySQL, яка залишається популярною і затребуваною на протязі вже більше п’ятнадцяти років.

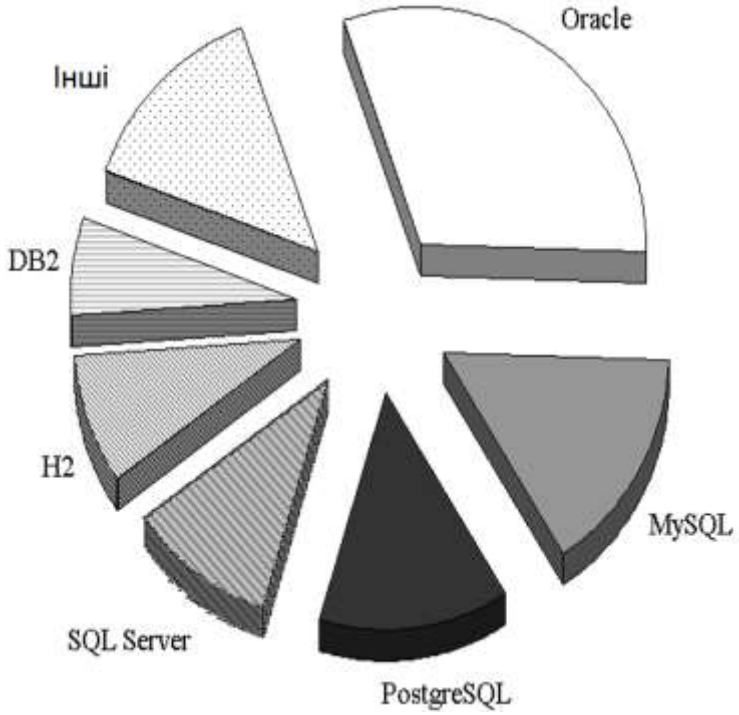


Рисунок 2.1 - Картина поширеності СУБД станом на 2015рік

Картина поширеності СУБД станом на 2019 рік прийняла наступний вид (рис. 2.2).

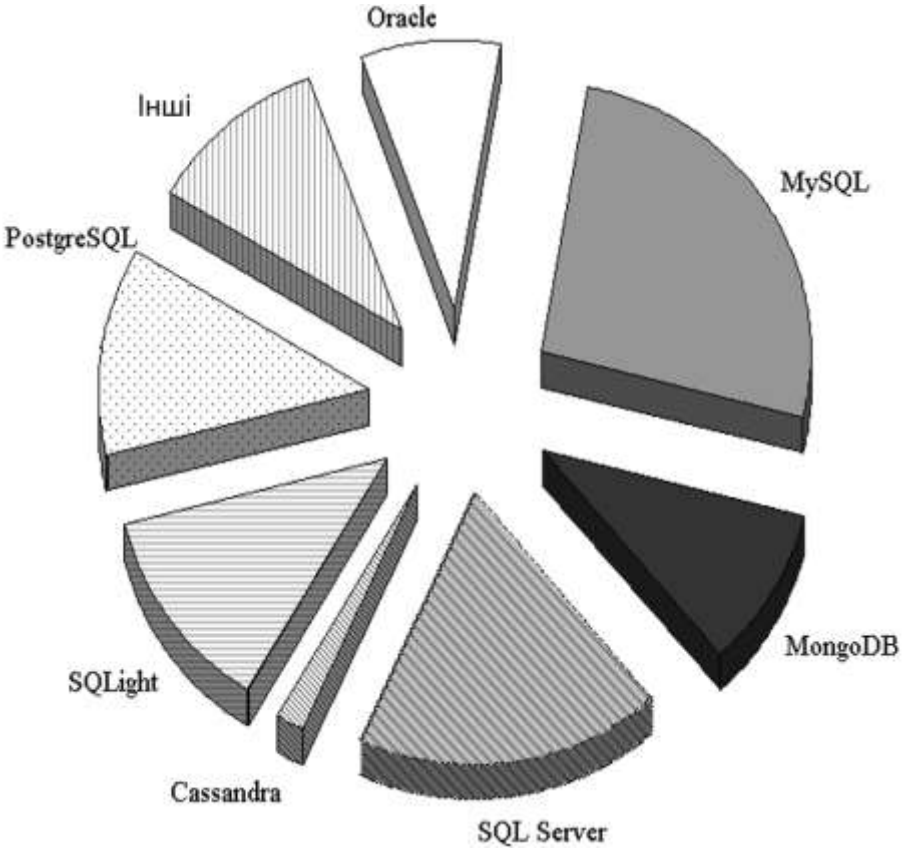


Рисунок 2.2 - Картина поширеності СУБД станом на 2019рік

Альтернативними варіантами можуть будуть PostgreSQL або Maria DB. Слід зауважити, що перераховані систему управління базами даних є безкоштовні. У тому випадку, якщо виникає необхідність впровадження СУБД на підприємство або велику компанію для здійснення автоматизації документообігу, то в даному випадку непоганим варіантом є СУБД Oracle Database, функціонал і продуктивність даної системи управління базами даних дійсно вражають. Унікальною в даному випадку є система управління базами даних MongoDB, вона також можна бути рекомендована для вирішення поставленої задачі. При обмеженому фінансуванні для здійснення автоматизації документообігу можна розглянути варіант використання СУБД PostgreSQL.

На сьогодні існує декілька затребуваних систем управління базами даних, вибираючи з яких, гарантовано можна знайти ту СУБД, яка буде відповідати поставленим вимогам конкретного промислового підприємства або великої компанії. Завдяки тому, що є також безліч відмінних безкоштовних варіантів, для невеликих організацій і компаній можна знайти інструмент управління базою даних, який буде відповідати їхнім критеріям. Якщо промислове підприємства або велика компанія вимагає більш насиченого функціоналом рішення систему управління базами даних, існує достатня кількість платних пропозицій систему управління базами даних.

2.2 Розробка моделі представлення релевантності подібності рядків

Для вирішення поставленої задачі в дипломній роботі виникла необхідність розробки моделі релевантності подібності рядків. В якості базової інформації для вирішення поставленої задачі розглянемо алгоритм задачею якого є порівняння рядків, а як результат роботи повертає близькість порівняних рядкових значень у відсотках. Для реалізації моделі представлення релевантності подібності рядків використаємо даний алгоритм. Вхідними даними моделі використаємо наступні параметри: два рядки яких виникла потреба в порівнянні, а також параметр з яким

буде виконуватися порівняння *N*- максимальна довжина порівнюваних рядків. Вихідний параметр моделі відсоток подібності релевантності, в даному випадку значення параметра 0% повну розбіжність підрядків, а 100% повне співпадання порівнюваних рядків.

При цьому порівняння підрядків виконується наступним чином: визначаємо два рядка, довжина яких обмежується вхідним параметром *N*. У випадку співпадання довжин рядків знаходимо співпадання першого рядка підрядків в підрядках іншого рядка, а також обчислюємо суму співпадань другого рядка підрядків з підрядками які входять в свою чергу до першого рядка. В результаті виконаних дій отримаємо відношення отриманих сум, в процентного вигляді, результат запам'ятовується як тимчасовий коефіцієнт релевантності. Обчислюємо середнє значення всіх тимчасових коефіцієнтів релевантності і отримуємо вихідний параметр моделі у вигляді релевантності всіх вхідних рядків у відсотках. Модель опису релевантності для взятих окремо двох рядків S_{t1} *і* S_{t2}, відповідних довжин l_1 *і* l_2 і при цьому задаємо максимальну довжину використовуваних підрядків *N* буде наступний вигляд визначимо:

1. Формуємо матрицю, елементами якої є набори всіх варіантів підрядків довжина яких не перевищує *N*:

$$G_j(i)=\left\{g_{j1}(i),...,g_{jk}(i),...,g_{jn}(i)\right\};\quad j=1,2;\quad i=\overline{1,N};\quad n=l_j-i+1, \qquad (2.1)$$

де *i* - довжина підрядка вхідного рядка; *j* – порядковий номер вхідного рядка; *n* – загальна кількість підрядків довжина яких *i* в *j*-му рядку.

2. Кожному елементу матриці $G_j(i)$ ставимо відповідним чином відфільтровану множину $G_j^*(i)$, елементи множин не повинні пересікатися з елементами матриці $G_j(i)$, таким чином видаляємо повторювані елементи з матриці $G_j(i)$ в результаті виконаних дій отримуємо множину $G_j^*(i)$:

$$G_j^*(i)=\left\{g_{j1}^*(i),...,g_{jk}^*(i),...,g_{jn}^*(i)\right\};\qquad j=1,2;\quad i=\overline{1,N};\quad m\le l_j-i+1, \qquad (2.2)$$

де m – в даному випадку загальна кількість підрядків які не повторюються заданої довжини в j-му рядку.

3. Значення вихідного параметра моделі релевантності $FR=\left(l_1,l_2,N\right)$ обчислено наступним чином:

$$FR=\left(l_1,l_2,N\right)=\frac{\sum_{i=1}^{N} fr(i)}{N}, \tag{2.3}$$

$$fr(i)=\frac{\left|{}^{*}G_1(i)\right|+\left|{}^{*}G_2(i)\right|}{\left|G_1(i)\right|+\left|G_2(i)\right|} \tag{2.4}$$

$${}^{*}G_j(i)=G_j(i)\cap G_j^{*}(i), \tag{2.5}$$

де $g_j(i)\in {}^{*}G_j(i)\Rightarrow \exists\, g_k^{*}(i): g_j(i)=g_k^{*}(i)$, тобто елементи матриці ${}^{*}G_j(i)$ формуються з елементів матриці $G_j(i)$, для яких є відповідні елементи у матриці $G_j^{*}(i)$. $G_j(i)$ - масив елементів підрядків довжина яких i відповідного рядка l_j; $\left|G_j(i)\right|$ - кількість в елементах матриці знаходиться підрядків $G_j(i)$; $G_j^{*}(i)$ - матриця, яка формується з матриці $G_j(i)$ при цьому в ній відсутні повторяємі підрядки матриці $G_j(i)$; $\left|G_j^{*}(i)\right|$ - кількість в елементах матриці знаходиться підрядків $G_j^{*}(i)$; N - максимальна довжина відповідного підрядка розглянутого рядка.

Після того як модель опису релевантності отримана, переходимо до визначення наступних задач а саме: виникає необхідність обчислення значення оптимального значення параметра N; розробити оптимальний метод розбиття набору рядків на відповідні підрядки.

Розглянемо оптимальний метод розбиття набору рядків на відповідні підрядки. На сьогоднішній день використовується класичний алгоритм розбиття набору рядків на грами. Уже більше 40 років алгоритм n-грамної індексації

використовується в інформаційно – пошукових системах для розбиття рядків. В основі алгоритма *n*-грамної індексації (словникова *n*-грамна індексація) використовуються наступні обмеження використання алгоритму: якщо рядок *str1* при реалізації алгоритму входить в рядок *str2* в результаті виконання операцій редагування не більше ніж за *k* елементарних кроків, то в даному випадку при будь-якому, представленні *str1* у вигляді різних розглянутих конкатенацій з *k+1* – им розглянутим рядком, один з представлених рядків буде в розглянутому варіанті підрядком *str2*. Розглянуту закономірність є можливість підсилити, прийнявши до уваги, що в даному варіанті серед представлених рядків можна стверджувати що існує такий підрядок, різниця в позиціях його розміщення в рядках *str1* і *str2* не більше *k*.

Таким чином, в результаті розглянутого алгоритму *n*-грамної індексації, можна зробити наступний висновок - задача розбиття набору рядків на відповідні підрядки зводиться до задачі пошуку слів, які в даному випадку містить заданий підрядок який розглядається алгоритмом. Розглянемо алгоритм порівняння рядків за схемою розглянутою вище: в якості вхідних параметрів задаймо відповідно два рядки "макс" і "марс", максимальна довжина в даному випадку підрядка = 4. В даному випадку запропонований підхід до розбиття рядка на підрядки обчислює набір всіх варіантів можливих комбінацій підрядків з довжиною яка не повинна перевищувати заданої, в нашому випадку це значення - 4, і далі обчислюємо загальну кількість співпадань підрядків першого рядка з розглянутим іншим. Результатом виконання алгоритму отримуємо відношення загального числа співпадань до число розглянутих варіантів, тобто коефіцієнта подібності рядків для заданої фіксованої довжини *N* і повертається в якості вихідного параметра моделі. Кінцевим результатом є середнє значення всіх обчислених коефіцієнтів (табл.2.1).

Результати наведені в табл. 2.1 демонструють алгоритм обчислення коефіцієнта коефіцієнта подібності двох порівнювальних рядків на релевантність.

Таблиця 2.1 - Обчислення релевантності рядків «макс» і «марс» для *N*=4

Підрядки порівняння	Підрядок іншого рядка	Співпадання так/ні	Число співпадань	Число варіантів
Проводимо порівняння рядка «макс» з рядком «марс» для *N*=1				
м	м, а, р, с	так	3	4
а	м, а, р, с	так		
к	м, а, р, с	ні		
с	м, а, р, с	так		
Проводимо порівняння рядка «макс» з рядком «марс» для *N*=2				
ма	ма, рс	так	1	3
ак	ма, рс	ні		
кс	ма, рс	ні		
Проводимо порівняння рядка «макс» з рядком «марс» для *N*=3				
мак	мар, арс	ні	0	2
акс	мар, арс	ні		
Проводимо порівняння рядка «макс» з рядком «марс» для *N*=4				
макс	марс	ні	0	1
Загальний результат			4	10

Для приведених рядків «макс» і «марс» і при заданій максимальній довжині підрядка $N = 4$, отримали значення релевантності (коефіцієнта подібності), величиною 4/10, або 0,4. Якщо при роботі алгоритму використовувати підрядки зі значенням N меншої довжини, то в результаті отримаємо зовсім інші коефіцієнти подібності: наприклад, при використанні підрядків зі значенням $N= 1$ результатом отримаємо наступний $3/4 = 0,75$.

Розглянемо інший спосіб розбиття рядка на підрядки, який під час обчислення здатен враховувати початок і кінець рядка, позначимо початок і кінець рядка символом «*»(табл..2.2).

В результаті обчислення даного варіанту отримали релевантність, величина якої склала 7/16 = 0,43, що відповідає більш точній оцінці ніж в попередньому варіанті - 40%. При обчисленні даного варіанта змінився не тільки коефіцієнт подібності релевантності, зріс об'єм грамів з 10 до 15, що може значно вплинути на швидкість відзнайдення інформації. Якщо в даному варіанті враховувати початок\кінець рядка, виникає необхідність в розширенні алфавіту, що призведе в свою чергу до збільшення розміру інформації і також буде негативним чином впливати на швидкість роботи алгоритму пошуку відповідної інформації. Ефективність алгоритму починає значно падати вже починаючи з $N>=6$, що також є негативним впливом на роботу алгоритму.

Перейдемо до вирішення задачі знаходження оптимальної довжини підрядка в даному випадку –*N*. Як було показано раніше збільшення максимальної довжини підрядка –*N* призводить до значного зменшення швидкодії алгоритму пошуку необхідної інформації. Але в той же час при цьому пошук необхідної інформації стає більш точним. Таким чином рекомендується в якості –*N* оптимального значення використовувати діапазон значень 1 .. 5, при цьому необхідно враховувати довжини слів які розглядаються в даному випадку. Для підтвердження справедливості вище сказаного проведемо експеримент: витягнемо всі пошукові поля які використовуються при формуванні пошукового запиту (реквізити-фірм, реквізити клієнта, коментарі і т.д.) в загальну таблицю, при цьому проведемо сортування по довжині рядків. Далі для кожного блоку однакових довжин витягуємо з бази даних до ста відповідних записів довжиною яка не перевищує 50 символів в запису. Проводимо обчислення для кожного відповідного блоку однакових довжин і таким чином отримаємо ступінь подібності відповідних пар рядків застосувавши при цьому декартовий добуток.

Проводимо фільтрацію слів, слова, які отримали ступінь подібності і подолали поріг >= 0.35, заносимо їх в лог ідентифікації, з результатом отриманого ступеня подібності.

Таблиця 2.2 - Обчислення релевантності рядків «*макс*» і «*марс*» для N=4

Підрядки порівняння	Підрядок іншого рядка	Співпадання так/ні	Число співпадань	Число варіантів
Проводимо порівняння рядка «макс» з рядком «марс» для N=1				
м	м, а, р, с	так	3	4
а	м, а, р, с	так		
к	м, а, р, с	ні		
с	м, а, р, с	так		
Проводимо порівняння рядка «макс» з рядком «марс» для N=2				
*м	*м, ма, ар, рс, с*	так	3	5
ма	*м, ма, ар, рс, с*	так		
ак	*м, ма, ар, рс, с*	ні		
кс	*м, ма, ар, рс, с*	ні		
с*	*м, ма, ар, рс, с*	так		
Проводимо порівняння рядка «макс» з рядком «марс» для N=3				
*ма	*ма, мар, арс, рс*	так	1	4
мак	*ма, мар, арс, рс*	ні		
акс	*ма, мар, арс, рс*	ні		
кс*	*ма, мар, арс, рс*	ні		
Проводимо порівняння рядка «макс» з рядком «марс» для N=4				
*мак	*марс	ні	0	3
макс	марс	ні		
акс*	арс*	ні		
Загальний результат			7	16

Далі проводиться експертна оцінка отриманих результатів, експерти переглядають лог ідентифікації та приймають відповідне рішення щодо переглянутих даних, в даному випадку справедливість виставленої оцінки.

Використовуючи результати експертних оцінок обчислюємо, відношення кількості справедливих виставлених оцінок ідентифікацій до загального числа оцінок логу ідентифікації, отримаємо графік роботи пошукового алгоритму, показаний на рис. 2.3. Подолання при цьому запропонованим варіантом обчислення співпадання рядків порога в 85% є непоганим результатом для даних слів фіксованої довжини.

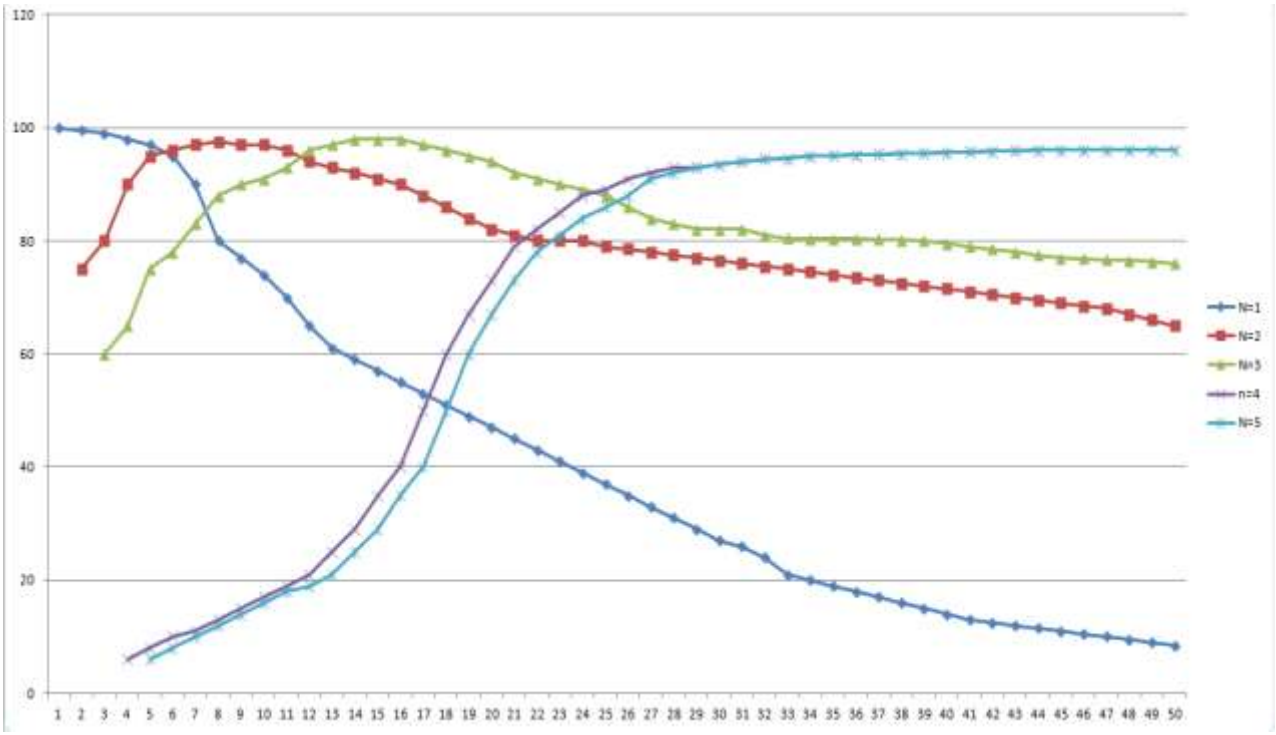


Рисунок 2.3 - Графік прийняття рішення при $N = \{1..5\}$

При проведенні аналізу отриманих результатів бачимо, що для $N > 3$ виявилась характерна помилка яка проявляє себе на коротких словах – як приклад можна навести тестову пару слів «макс» - «марс», відповідно дає 0% подібності релевантності при роботі алгоритму пошуку для $N = 3$ і також для $N = 4$, якщо говорити про дані, які в наявності в БД , то можна навести інший приклад пари слів: *PETRO* - *PETOR* подібність пар тобто релевантність 0% при $N = 4$,та 0.4 при $N = 3$. Значення отримані при роботі алгоритму при значенні максимальної довжини підрядка $N = 4$ в більшій мірі залежать від того, в якому саме місці рядка може знаходиться «помилка» при цьому чим більше грам вона враховує в даному варіанті. Наприклад, коли оцінка подібності проводиться перебуваючи ближче до

центру рядка, в даному випадку оцінка нижча, і навпаки - коли оцінка подібності проводиться подальше від центру, тим менше, тим вище оцінка. Виходячи з результатів обробки проведеного експерименту, можна прийняти наступні висновки:

1. При значенні максимальної довжини підрядка $N \geq 4$ ефективність роботи алгоритму результатом якого релевантність двох рядків приблизно однакова і надає кращі результати на реквізитах більшої довжини - адреси фірм, адреси користувачів а також коментарів.

2. При значенні максимальної довжини підрядка $N = 3$ ефективність роботи алгоритму результатом якого релевантність двох рядків найбільш ефективна при задіяні слів довжина яких приблизно в діапазоні 8 .. 26 символів.

3. При значенні максимальної довжини підрядка $N = 2$ ефективність роботи алгоритму результатом якого релевантність двох рядків найбільш ефективна при задіяні слів довжина яких приблизно в діапазоні 4 .. 18 символів.

4. При значенні максимальної довжини підрядка $N = 2$ ефективність роботи алгоритму результатом якого релевантність двох рядків найбільш ефективна при задіяні слів довжина яких до 6 символів.

Таким чином на основі проведеного дослідження та наведених вище висновків наведемо загальні висновки отримання релевантності двох рядків:

1. При отримання релевантності двох рядків на коротких реквізитах, і при цьому довжина яких не перевищує 30 букв, при даному варіанті найбільш ефективно використовувати так звану сукупну оцінку релевантності, при значеннях максимальної довжини підрядка в діапазоні $N = 1,...,4$. Отриманні результати релевантності оцінки при значеннях максимальної довжини підрядка $N > 5$ можна не враховувати.

2. При використанні лінійного збільшення значення максимальної довжини підрядка $N > 4$, оцінка точності подібності і релевантності практично не змінюється при цьому збільшенні довжини рядка, таким чином можна

стверджувати, при обчисленні реквізитів більшої довжини достатньо використання значення максимальної довжини підрядка при $N=4$. Також на користь даного рішення вказує той факт, що діапазон правильних рішень для використання значення максимальної довжини підрядка при $N=4$ при здійсненні обчислень для слів довжиною більшою ніж 25., а при збільшенні значення максимальної довжини підрядка N ця межа піднімається вгору підвищується, звужуючи при цьому об'єм рядків на якому алгоритм ефективно працює.

Для прискорення роботи алгоритму оцінки подібності та релевантності порівнюваних рядків можна вирішити на початковому етапі пошуку однакових грам. Значення максимальної довжини підрядка $N=4$, в даному випадку тип даних підрядка буде char(4) з довжиною 5 байт. Для прискорення роботи алгоритму оцінки подібності та релевантності порівнюваних рядків необхідно використовувати для зберігання подібних рядків пам'ять менше 5 байт. Таким чином необхідно присвоїти символам цифрові значення метою зменшення виділення об'єму пам'яті під символи, в результаті збільшиться швидкість виконання алгоритму пошуку. Це можна зробити наступним чином:

$$H(a)=\left(ASCII\_CODE(a)-base\right)\bmod m, \qquad (2.6)$$

яка в даному випадку обчисляє різницю коду символам таблиці ASCII і базового значення потім використається ділення по модулю розміру сигнатури. Використання наведеної функція достатньо просто, відносно легко обчислюється, час пошуку в хеші на її основі релеватності порівнюваних рядків значно менший, ніж при використанні для вирішення даної задачі алгоритму послідовного перебору. Запропонований підхід використання хеша, як показали результати проведеного експеримента, досить нерівномірно обчислює та виконує розподіл символів на відповідні групи. При заміні запропонованої хеш-функції (2.6) рівномірною відповідною функцією *H(a)*, яка використовує в своїй основі рівні частоти появи символів в різних елементах сигнатури, швидкість пошуку релевантності при цьому зростає близько в два рази. Для побудови такої хеш-

функції, необхідно обчислити їхні частоти появи відповідних символів в рядках. Для баз даних, які містять більше 2-4 мільйони записів, частоти появи символів легко обчислюються. Після обробки бази даних таблиць з даними клієнтів і при обчисленні частот отримали наступні ймовірностей появи символів (табл. 2.3).

На основі отриманих результатів в результаті проведених обчислень частотної інформації була отримана хеш-функція. Таким чином хешування по сигнатурі в даному випадку добре підходить для «дискового» пошуку, в основі алгоритму якого лежить побудова графа, коли сторінки відповідним чином індексуються і вибираються при цьому безпосередньо з диску, тому під час виконання пошуку обчислюється відносно невеликий об’єм списків, при цьому витягнуті списки займають декілька дискових сторінок.

Таблиця 2.3 - Частота появи символів з рівномірною частотою

Буква	Частота	Буква	Частота	Буква	Частота	Буква	Частота
пропуск	0,175	о	0,090	ї	0,072	а	0,062
і	0,062	т	0,053	н	0,053	с	0,045
р	0,040	в	0,038	л	0,035	к	0,028
м	0,026	д	0,025	п	0,023	у	0,021
я	0,018	и	0,016	з	0,016	ь	0,014
б	0,014	г	0,013	ч	0,012	й	0,010
х	0,009	ж	0,007	ю	0,006	ш	0,006
ц	0,004	щ	0,002	є	0,003	ф	0,002

Розглянемо алгоритм який використовується при побудові побудови префіксного коду:

1. Символи вхідного алфавіта, який подається на алгоритм обчислення релевантності, реалізують послідовний список вільних вузлів. Кожному листку такого списку призначається вага, яка визначається частотою (ймовірністю) входжень символа в рядок.

2. Визначаються два вузла представленого дерева, вага яких при цьому мінімальна.

3. Далі визначається базовий (батьківський) вузол над цими двома вузлами. Вага батьківського вузла обчислюється як сума ваг нащадків.

4. Утворений базовий вузол додається в список перерахованих вільних вузлів, а його нащадки видаляються зі списку.

5. При цьому кожна дуга, що виходить з батька маркується одиницею, інша маркується нулем.

6. Наведені кроки алгоритму, починаючи з 2, будуть повторюватися, поки в представленому списку вільних вузлів в результаті перебору отримаємо один вільний вузол. Він в даному випадку і буде коренем дерева.

Отримане, в результаті наведених дій дерево кодування показано на рис. 2.4. Використаємо отримане дерево кодування, для прикладів розглянутих раніше, отримані результати наведені в табл. 2.4.

Таблиця 2.4 - Слова в префіксному коді з використанням дерева кодування

макс	0	0	1	1	0	1	0	0	1	0	0	1	0	0	0	1	0	0				
марс	0	0	1	1	0	1	0	0	1	1	1	0	1	0	0	1	0	0				
Ротор	1	1	0	1	0	0	1	1	0	0	0	0	0	1	1	1	1	0	1	0		

При використанні даного підходу правило префіксного підходу не буде застосовуватися, по тій причині, що деякі коди представлення символів починають зливатися. Дана проблема вирішується додаванням до кодів символів одиниці. В даному випадку, коди символів приймають однакові значення довжини і тим самим при цьому збільшується довжина загального коду. Для вирішення поставленої проблеми необхідно використовувати не двійковий префіксний код символів, а відповідний десятковий (табл. 2.5). Даний підхід має сенс використання для прискорення роботи алгоритму пошуку.

Проведемо обчислення середньої довжину коду, у випадку використання не двійкового префіксного коду символів, а десятковий префіксний код, наведено в табл. 2.6.

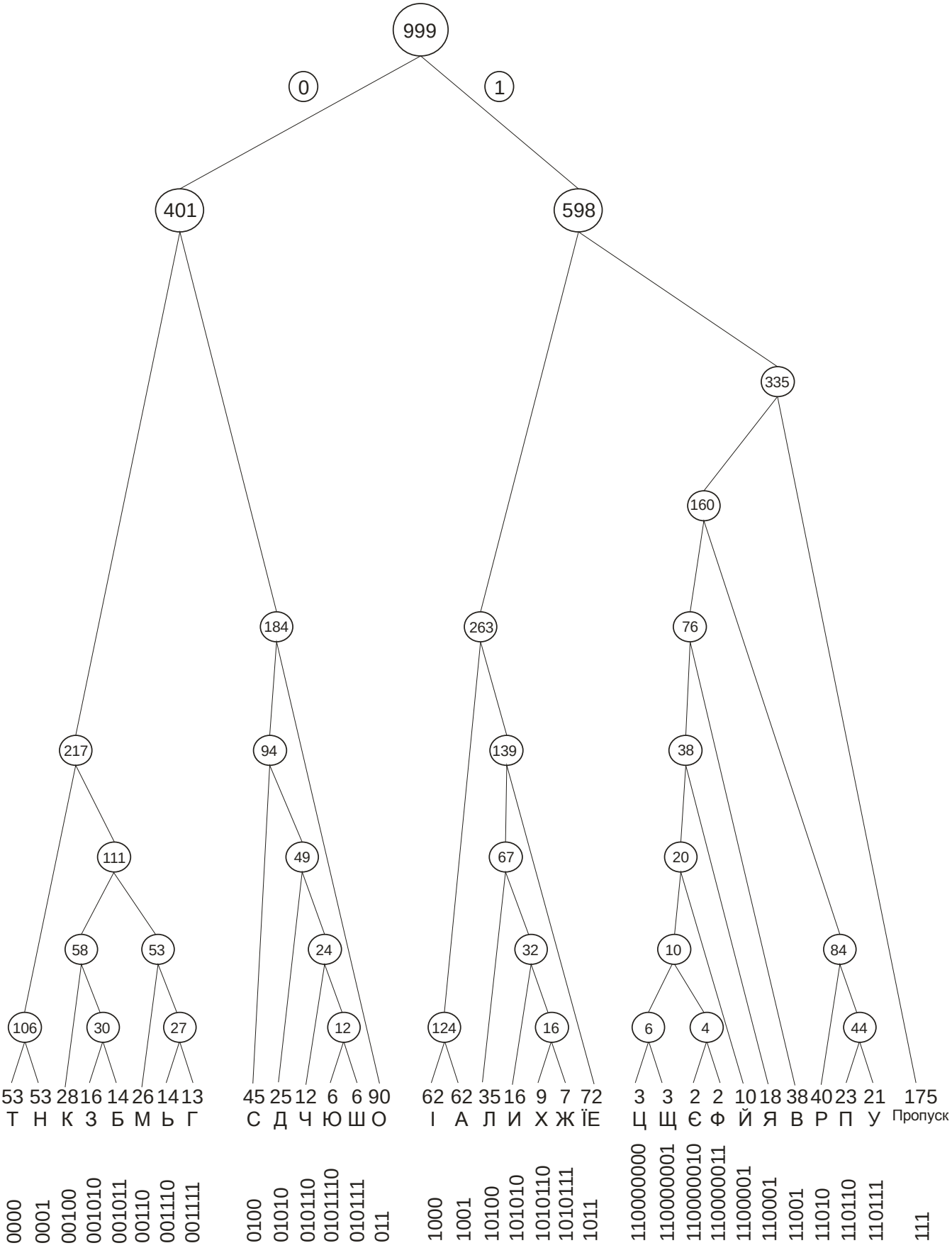


Рисунок 2.4 - Побудова коду на основі рівномірної частотної інформації

Таблиця 2.5 - Префіксний відповідний десятковий код символів

Буква	Частота	Код	Буква	Частота	Код
пропуск	0,175	0	о	0,090	4
і	0,062	6	т	0,053	8
р	0,040	11	в	0,038	12
м	0,026	15	д	0,025	16
я	0,018	19	и	0,016	20
б	0,014	22	г	0,013	24
х	0,009	27	ж	0,007	28
ц	0,004	31	щ	0,002	33
ї	0,072	5	а	0,062	7
н	0,053	9	с	0,045	10
л	0,035	13	к	0,028	14
п	0,023	17	у	0,021	18
з	0,016	21	ь	0,014	23
ю	0,006	29	ш	0,006	30
є	0,003	32	ф	0,002	34

Таблиця 2.6 – Обчислення довжини коду при використанні десяткового префіксний коду

Буква	Частота	$Код_{10}$	$Код_2$	Довжина бітового рядка	(частота) (довжина коду)	Мат. очікування
пропуск	0,175	0	0	1	0,175	0,175
о	0,090	4	100	3	0,27	0,858
ї,е	0,072	5	101	3	0,216	
і	0,062	6	110	3	0,186	
с	0,045	10	1010	4	0,18	
...	...	...	...	...	...	...
ш	0,006	30	11110	5	0,03	
ц	0,004	31	11111	5	0,02	
є	0,003	32	100000	6	0,018	0,042
щ	0,002	33	100001	6	0,012	
ф	0,002	34	100010	6	0,012	
					Результат	3,417

Таким чином в результаті проведеної оптимізації довжина коду становить 6 біт $\left(34_{dec} = 10010_{bit}\right)$ і в найгіршому варіанті буде використана пам'ять яка займе : $\langle фффф \rangle = 34343434_{dec} = 24bit$ проти 40 біт при використанні формата nvarchar.

Проведемо експертну оцінку роботи алгоритму отримання релевантності порівнюваних рядків із застосуванням кодування і без його кодування (табл. 2.7).

Таблиця 2.7- Результати тестування оцінки релевантності із застосуванням кодування і без його використання

Операція	Без кодування (тис. рядків)				З кодування (тис. рядків)			
	10	20	30	50	10	20	30	50
перекодування *N*-грам вхідного масиву (сек)	0	0	0	0	12,5	20,5	28,7	47
Отримання релевантності при $N = 4$ (сек)	39,5	78	116	207	22	47,5	69	117
Отримання релевантності при $N = 1..4$ (сек)	77,5	154	242	378	49	98,8	162	263
Результат	117	232	358	585	83,5	166,8	259,7	427
приріст швидкості (%)					28,6	28,1	27,5	27

В результаті проведеного дослідження обчисливши середнє значення, будемо мати в даному випадку приріст швидкості в середньому на 27,8%.

Таким чином, отримані результати в результаті проведеного дослідження дозволяють побачити оцінку складності запропонованого алгоритму визначення релевантності двох рядків. Простота і при цьому покращань швидкодія запропонованого алгоритму визначення релевантності двох рядків є його перевагою, алгоритм є конкурентним у випадку проведення обчислень великих об’ємів інформації.

2.3 Розробка моделі наближеного пошуку при опрацюванні пошукових рядків

В теорії інформації а також і в комп'ютерної лінгвістики для визначення різниці двох послідовностей рядків використовується відстань Левенштейна–міра яка визначає мінімальне число кроків заміни, вставки або видалення які необхідно виконати для отримання одного рядка з іншого. Алгоритм запропонований В.Й.

Левенштейном надає можливість оцінити, наскільки один рядок відповідає іншому. Алгоритм представлений Левенштейном надає змогу в чисельній формі оцінити подібність двох рядків які приймають участь в порівнянні. Алгоритму Левенштейна визначає в результаті обчислення мінімальне число кроків проведення операцій заміни, вставки або видалення які необхідних для отримання одного рядка з іншого. Наприклад обчислимо відстань між двома рядками тост і текст. Перетворити перше слово «текст» в «тост», за рахунок проведення операцій заміни, вставки або видалення.

«текст» - «тост» - вхідні дані (початковий стан)

крок 1 (використаємо операцію заміни) - «токст» - «тост»

крок 2 (використаємо операцію видалення)- «тост» - «тост»

Число кроків для переведення першого слова вдруге слово становить два (заміни + видалення) Таким чином, відстань Левенштейна між рядками «текст» - «тост» становить 2.

Розглянемо моделі наближеного пошуку відстані Левенштейна між рядками. Приймемо що S_1 і S_2 - два рядки, для обчислення відстані Левенштейна, довжиною N і M відстань Левенштейна $d(S_1, S_2)$ визначимо за наведеною формулою $d(S_1, S_2) = d(M, N)$, при цьому елементи:

$$D(i,j) = \begin{cases} 0 & \text{якщо } i=1, j=1, \\ i & \text{якщо } i>1, j=1, \\ j & \text{якщо } i=1, j>1, \\ \min(& \text{якщо } i>1, j>1, \\ D(i, j-1)+1, & \\ D(i-1, j)+1, & \\ D(i-1, j-1)+m(S_1[i], S_2[j])), & \end{cases} \qquad (2.7)$$

де $m(S_1[i], S_2[j]) = \begin{cases} 0, & \text{якщо } S_1[i] = S_2[j] \\ 1, & \text{якщо } S_1[i] \neq S_2[j] \end{cases}$

Приведену модель наближеного пошуку відстані Левенштейна розглянемо більш детально. В наведеній формулі цикл по i відповідає за проведення операції

видалення (*D*) з першого слова, цикл по *j* відповідає за проведення операції вставки (*I*) в перше слово, а цикл по *i* і *j* - відповідає за проведення операції заміни символу (*R*) або відсутність замін (*M*). Таким чином відстань між двома нульовими словами становить 0. Для отримання пустого слова зі слова яке має довжину *i*, необхідно при цьому виконати *i* операцій видалення (*D*), а також щоб отримати слово довжиною *i* з пустого рядка, необхідно виконати *i* операції вставки. Для виконання операцій вставка/видалення необхідно провести в будь-якому разі одну операцію, для при виконанні операції заміни може скластися, така ситуація коли символи однакові - тоді операція заміни не виконується.

При проведенні обчислення відстані Левенштейна, справедливі наступні нерівності:

$$\begin{cases} d(S_1, S_2) \geq \| |S_1| - |S_2| \| \\ d(S_1, S_2) \leq \max(|S_1|, |S_2|) \\ d(S_1, S_2) = 0 \Leftrightarrow S_1 = S_2 \end{cases}$$

Побудуємо матрицю переведення рядка «текст» в рядок «тост» при проведенні обчислення відстані Левенштейна. Початковий стан матраці наведений в табл. 2.8.

Таблиця 2.8 - Початковий стан матриці

		Т	о	с	т
	0	1	2	3	4
Т	1				
е	2				
к	3				
с	4				
т	5				

Заповнення відповідними значеннями матриці виконується за наступною схемою:

$$a[i, j] = \min\left(a[i+1, j]+1; a[i, j-1]+1; a[i-1, j-1] + if\left(S_1[i] = S_2[j].0.1\right)\right),$$

де, $if\left(S_1[i] = S_2[j].0.1\right)$ - повертає 0, в разі якщо символи, які стоять в однакових позиціях та співпадають, 1 в іншому випадку. Таким чином, для

отримання значення елемента необхідно використати при цьому значення його сусідів по матриці знизу, зверху, а також по діагоналі. Так, наприклад для елемента $a[1,1]$ отримаємо:

$$a[1,1] = \min\left(a[0,1]+1; a[1,0]+1; a[0,0] + if\left(S_1[1] = S_2[1].0.1\right)\right)$$

$$a[1,1] = \min\left(2,2,0 + if\left("m" = "m", 0,1\right)\right) = \min(2,2,0) = 0$$

...

Таким чином в результаті проведених дій отримуємо матрицю, зі значення елементи якої дорівнюють відстані Левенштейна між словами $S_1 = "тост"$ *і* $S_2 = "текст"$ (табл. 2.9)

Таблиця 2.9 - Заповнена матриця відстанями Левенштейна

		т	о	с	т
	0	1	2	3	4
т	1	0	1	2	3
е	2	1	1	2	3
к	3	2	2	2	3
с	4	3	3	2	3
т	5	4	4	3	2

Алгоритм пошуку відстані Левенштейна найшов широке застосування в граматичних додатках, в MS Office використовує даний алгоритм для перевірки правопису, а також в інших продуктах, для пошуку і обробки текстів:

1. В інформаційно - пошукових системах для пошуку об'єктів , записів по найменуванню.

2. У системах управління базами даних при пошуку інформації по неповними заданими даними або неточними в пошуковому запиті.

3. Алгоритм Левенштейна також використовується для виправлення помилок під час набору тексту.

4. Алгоритм Левенштейна використовується для виправлення помилок в системах автоматичного розпізнавання відсканованої текстової інформації.

5. В програмних продуктах, пов'язаних з обробкою текстової інформації в автоматичному режимі.

Алгоритм пошуку відстані Левенштейна виконує фільтрацію текстової інформації при цьому відкидає неприйнятні варіанти. Алгоритму Левенштейна визначення відстані між рядками або текстовими полями притаманні наступні недоліки: при переміщенні місцями рядків, частини рядків можливий варіант отримання відносно великих відстаней; при цьому відстані між різними короткими рядками отримуються невеликими, в той же час відстані між подібними довгими рядками отримуються значними.

При використанні алгоритму послідовного перебору, рядки витягуються з бази даних послідовно і порівнюються з пошуковим зразком. В даному випадку для порівняння рядків запропоновані для використання бітовий алгоритм - agrep. Особливістю даного класу алгоритмів є їх висока ефективність. До недоліків даного класу алгоритмів слід віднести повільність їхньої роботи. Проведені дослідження в області алгоритмів пошуку інформації показали що не всі алгоритми, ефективніші послідовного перебору. Розглянемо наступний приклад. Пошук необхідної інформації здійснюється за шаблоном vivid в наступному рядку vivi&dv&vivid, в наведеному прикладі неспівпадання символів заданому шаблону і розглянутого тексту позначаємо великою буквою (табл. 2.10).

Таблиця 2.10- Приклад пошук в рядку по заданому шаблону

	v	i	v	i	&	d	v	&	v	i	v	i	d
1	v	i	v	i	D								
2		V											
3			v	i	V								
4				V									
5					V								
6						V							
7							v	I					
8									v	i	v	i	d

Алгоритму послідовного перебору вимагає виконання кроків не менше n-m +1 порівнянь і при цьому «працює» достатньо повільно. Алгоритму послідовного перебору достатньо простий при його реалізації на одній з мов програмування

також дозволяє проводити пошук інформації з використанням будь-якої довжини шаблону.

Розглянемо наступний алгоритм точного пошуку Бойєра-Мура. На сьогодні на практиці використовуються, декілька варіантів удосконалення наведеного простого рішення виконання поставленої задачі. Найбільш затребуванні на сьогоднішній день: алгоритм Бойера - Мура і алгоритм пошуку корисної інформації Кнута - Морріса - Пратта. Результати проведеного дослідження показали перевагу алгоритм пошуку якісної інформації Бойера – Мура. Так при використання даного алгоритму у більшості випадків він працює швидше при точного пошуку інформації. На сьогодні існує декілька модифікацій алгоритму Бойера – Мура. При використанні даного алгоритму, на початку його роботи порівняння проводитися між останнім символом використовуваного шаблону і *m*-ним символом рядка пошуку. Таким чином порівнюваємо *d* і *&*; дані символи в даній позиції не співпадають, і шаблон зсувається на наступний символ рядка пошуку. Тут необхідно зауважити - визначається символ рядка пошуку, який став причиною неспівпадання (у даному випадку *&*), а також встановлюється, в якому саме місці шаблону на якій позиції знаходиться даний символ. Залежно від отриманого результату на даному етапі приймається рішення про необхідну величину зсуву. У розглянутому варіанті символ *&* не знаходиться в шаблоні, тому в даному варіанті можна зсувати шаблон на *m* символів вправо. Наступне порівняння, як бачимо з прикладу, проводиться з останнім символом заданого шаблону пошуковим запитом і з десятим символом рядка пошуку, тобто ми виконали зсув рівно на п'ять позицій. Десятий символ в нашому рядку - *i*, в заданому шаблоні він знаходиться на четвертій і другій позиції. Далі проводимо зсув шаблона на одну позицію, так як *i* в тексті також може співпадати з символом який знаходиться на четвертій позиції шаблону.

До переваг даного методу слід віднести наступне, так як розбіжностей при роботі алгоритмів пошуку інформації буває значно більше, чим співпадань, що призводить, в свою чергу, до великих зсувів. В наведеному прикладі при

використанні алгоритму Бойера-Мура для пошуку співпадань необхідно виконати лише вісім порівнянь проти двадцяти при використанні алгоритму послідовного перебору.

Розглянемо наступний алгоритм пошуку інформації ЗСУВ-І. Даний алгоритм ЗСУВ-І опублікували в 1992 році науковці з університету Арізони - Сан Ву і Уді Манбер. Розглянемо роботу даного алгоритму: Позначимо чепез P - шаблон за яким ведеться пошуку, а через T - рядок пошуку необхідної інформації, довжина яких дорівнює n і m відповідно. Необхідно відшукати в даному випадку, не тільки всі входження шаблону P в рядок пошуку T, а також відшукати входження в рядок пошуку T всіх можливих префіксів шаблону P. Позначимо множину префіксів шаблону P наступним чином P^* і запишемо у вигляді формули:

$$P^* = \{ p_1,\ p_1 p_2\ , \dots , p_1 \dots p_n \}, \qquad n = \overline{1, N} \tag{2.8}$$

де n -довжина шаблону пошуку P

У наведеному вище прикладі множина префіксів шаблону пошуку P представляється наступними елементами:

$$P^* = \{V,\ VI,\ VIV,\ VIVI,\ VIVID\}$$

Відповідно до поставленої задачі сформуємо матрицю R, в якій представимо наступну інформацію - для будь-якої позиції рядка пошуку в комірці матриці буде вказано, чи ця позиція, є кінцем одного з префіксів шаблону P елементів з P^*. Отримані результати представлені в таблиці 2.11.

Таблиця 2.11 - Приклад пошук в рядку

	v	i	v	i	&	d	v	&	v	i	v	i	d
V	1	0	1	0	0	0	1	0	1	0	1	0	0
VI	0	1	0	1	0	0	0	0	0	1	0	1	0
VIV	0	0	1	0	0	0	0	0	0	0	1	0	0
VIVI	0	0	0	1	0	0	0	0	0	0	0	1	0
VIVID	0	0	0	0	0	0	0	0	0	0	0	0	1

В матриці R представленої в табл. 2.11, відображена наступна інформація $R[i, j] = 1$, у випадку якщо перші k символів шаблону пошуку P точно співпадають з k символами рядка пошуку T, а також з попередніми символами t_j

тобто, якщо виконується наступна умова $p_1 \dots p_n = t_{j-k+1} \dots t_j$. В даній ситуації слід приділити увагу на останній рядок, оскільки в даному випадку він відповідає входженню пошукового шаблону в текст пошуку, але також і інші рядки теж не менш важливі. Для формування матриці при використанні алгоритм пошуку інформації ЗСУВ-І слід звернути увагу на наступне: *j* + 1-й стовбець матриці залежить в даній ситуації тільки від *j*-го стовбця пошукового шаблону і також символу t_{j+1}. Наприклад, співпадання в *j*+*l* для viv буде виконуватися у випадку тільки тоді, коли співпадання символів виконувалося для vi в *j* і $t_{j+1} = v$. Після узагальнення розглянутої інформації отримаємо:

$$R[i,j] = \begin{cases} 1, & якщо\ R[i-1, j-1] = 1\ і\ p_i = t_j \\ 0, & якщо\ R[i-1, j-1] \neq 1 \end{cases} \qquad (2.9)$$

Будемо, в даному випадку, вважати наступне:

$$R[0, j] = 1,\ R[i, 0] = 0,\ \left(j = \overline{1 \dots m},\ i = \overline{1 \dots n}\right)$$

У випадку реалізації алгоритму пошуку інформації ЗСУВ-І , виникає необхідність програмування отриманої рекурсії , тому для обчислення кожного елемента матриці потрібно використання умовного оператора IF. При використанні пошукового шаблону, довжина якого не перевищує 32 біти, в такому стовпчики матриці є потреба представити у вигляді 32-бітових машинних слів. При виконанні цієї умови представляється можливим обчислювати весь стовпець матриці відразу. Якщо звернути увагу на перші два стовбці матриці, одиниці в наступному стовпці можуть з'явитися при виконанні наступних умов одночасно: одиниця може з'явитися в комірці матриці, яка відповідає - «I» в рядку і для останнього елемента префікса пошукового шаблону «*VI*» $p_2 = t_2 = i$, а також при виконанні другої умови, якщо в першому, стовбці матриці в комірці вище стоїть одиниця. Виконання першої умови надає рівність останніх символів рядка пошуку і шаблону, а виконання другої умови - співпадання попередніх символів рядка і шаблону. Для перевірити другої умови достатньо зсунути перший

стовпець. матриці Для перевірки першої умови, необхідно попередньо для всіх символів рядка пошуку обчислити вектори які мають довжину *n*. Так, наприклад вектор для символу $t_4 = i$ в другій і четвертій позиціях має одиниці і нулі у всіх інших комірках матриці отримуємо вектор t_4- 01010. Подібним чином обчислюємо вектор для $t_3 = v - 10100,$ а також для $t_6 = d - 00000.$ Для перевірки чи виконується друга умова необхідно виконати порівняння отриманого в результаті зсуву першого стовбця вниз (в нашому варіанті це вираз 10000 після обчислення приймає вид- 11000) і таким чином визначається характеристичний вектор для $t_2 = i$. В комірках матриці в позиціях, в яких одиниці співпадають, надаємо їм також одиничне значення, а в решту комірок матриці заносяться нулі. Таким чином обчислене значення стає наступним стовпцем матриці. Єдине, на що слід звернути увагу, в даному випадку на першу позицію, для якої виконується друга умова, оскільки нами введена наступна умова: $\forall\ j\left(1 \le j \le m\right)$ тому приймемо $R\left[0, j\right] = 1$. Таким чином при проведенні обчислення перша позиція при виконанні операції зсуву завжди будемо заповнювати одиницями. Тому отримаємо наступні результати: для третього стовпця - 10100 після виконання операції зсуву та доповнення матимемо - 11010. Після виконання операції порівняння (побітової операції AND), з вектором для $t_4 = i\,(01010)$ отримаємо наступний результат - 01010. Таким чином для реалізації алгоритму пошуку, була на першому етапі побудована матриця, яка надає всі співпадання префіксами *P** пошукового шаблону *P*. На другому етапі необхідно обчислити отриману рекурсію, результатом обчислення є заповненні комірки матриці відповідними елементами. Запропонований новий підхід обчислення стовбців матриці з використанням зсуву попереднього стовбця матриці і операції побітового множення «AND». При виконанні алгоритму пошуку, необхідно виконати операцію порівняння не більше *n* раз, порівняння виконується на рівні бітових операцій. Розглянутий алгоритм пошуку в разі необхідності досить легко доповнити для обробки, в разі необхідності, більш складних шаблонів.

Розглянемо наступний приклад: необхідно знайти слова, довжина яких не перевищує п'яти, замість використання пошукового шаблону «vivid», на четвертій і другій позиціях в цих словах повинна знаходитися *i*. Для вирішення поставленої задачі необхідно змінити попередню обробку і при цьому також відкоригувати матрицю характеристичних векторів. Таким чином на першій, п'ятій і на третій позиціях у всіх векторах необхідно виставити одиниці. На позиціях в комірках матриці де виставленні одиниці може бути будь-який символ рядка пошуку. При необхідності виключити при роботі алгоритму, з розгляду цифри, то їх вектори характеристичні необхідно заповнити нулями (00000). В результаті проведеного дослідження наведеного алгоритму можна зробити наступне заключення, даний алгоритм легко адаптувати на випадок застосування його в алгоритмах наближеного пошуку.

Розглянемо ситуацію в разі необхідності виконати наближений пошук. Наприклад необхідно знайти входження шаблону «vivid» з одним зміненим символом рядка пошуку. Для вирішення поставленої задачі заповнимо дві матриці, як представлено в табл. 2.12 і в табл. 2.13.

Таблиця 2.12 - Матриця R_1

	v	i	v	i	&	d	v	&	v	i	v	i	d
V	1	0	1	0	0	0	1	0	1	0	1	0	0
VI	0	1	0	1	0	0	0	0	0	1	0	1	0
VIV	0	0	1	0	0	0	0	0	0	0	1	0	0
VIVI	0	0	0	1	0	0	0	0	0	0	0	1	0
VIVID	0	0	0	0	0	0	0	0	0	0	0	0	1

Таблиця 2.13 - Матриця R_2

	v	i	v	i	&	d	v	&	v	i	v	i	d
V	1	1	1	1	1	1	1	1	1	1	1	1	1
VI	0	1	0	1	0	0	0	1	0	1	0	1	0
VIV	0	0	1	0	1	0	0	0	1	0	1	0	1
VIVI	0	0	0	1	0	0	0	0	0	1	0	1	0
VIVID	0	0	0	0	1	0	0	0	0	0	0	0	1

Як бачимо з вмісту наведених таблиць – табл. R_1 співпадає зі значеннями з табл. 2.11 а також побудована аналогічним підходом. Таблиця R_2 подібна до табл.

2.12, лише з різницею, в даній таблиці позначенні як точні співпадіння, так і співпадання при зміненому одному символі рядка пошуку. П'ятий стовпець матриці R_2 відрізняється, як представлено в матрицях, від п'ятого стовпця матриці R_1 в першій, п'ятій і третій позиціях рядка пошуку. Таким чином бачимо, що підрядок «vivi&» співпадає з пошуковим шаблоном «vivid» з однією заміненою буквою, підрядок «vi&» також співпадає з пошуковим шаблоном «viv» з однією заміненою буквою і символ & співпадає з *v* (перший рядок матриці R_2 заповнений одиницями). Таким чином співпадання підрядка «vivi&» з пошуковим шаблоном «vivid» при одній заміні показано в четвертому стовпчику матриці R_1 як точне співпадання по пошуковому шаблону vivi. В результаті виконаних дій, якщо присутнє точне співпадання до останнього символу рядка пошуку, тоді в даному випадку, існує співпадання, не більше одного, з однією заміною пошукового шаблону. Один із способів заповнення матриці R_2 додатковими одиницями, тобто заповнення матриці R_2 по матриці R_1, полягає в наступному: зсув попереднього стовпчика матриці R_1 вниз, при цьому операція над бітами AND не виконується. Розглянемо наступний десятий стовпець в матриці R_2 - 11010, як бачимо в другому рядку матриці знаходиться одиниця (що означає точне співпадання по підрядку vi), яке, при цьому, забезпечується виконанням операції зсуву. Четвертий рядок в даному стовбці матриці відповідає за співпадання підрядка v&vi з префіксом пошукового шаблону «vivi», але тут, як бачимо, заміна відбулася раніше. При розгляді дев'ятого стовбця матриці R_2 також маємо співпадання з однією заміною підрядка «v&v» і при перевірці останнього символу на співпадання *-i*.

Всі можливі варіанти, при розгляді даного підходу щодо наближеного пошуку враховуються, завдяки використанню додаткових двох арифметичних операцій. В разі, якщо має місце точного співпадання поточного символу рядка пошуку з символом префікса пошукового шаблону або в результаті заміни, то рішення по виниклій ситуації приймається по результатам зсуву попереднього

стовпчика матриці R_1. У випадку, якщо заміна символу рядка пошуку з символом префікса пошукового шаблону відбулася раніше, то в даному випадку проводиться заповнення одиницею попереднього стовпчика матриці R_2 а проведення бітової операції AND над даним стовпчиком і його характеристичним вектором (табл. 2.14).

Таблиця 2.14 – Реалізація алгоритму наближеного пошуку

	v	i	v	i	&	d	v	&	v	i	v	i	d
V													
VI		1	0	1	0	0	0	1	0	1	0	1	0
VIV		0	1	0	1	0	0	0	1	0	1	0	1
VIVI		0	0	1	0	0	0	0	0	1	0	1	0
VIVID		0	0	0	1	0	0	0	0	0	0	0	1

Розглянемо виконання операцій вставки і пропуски при виконанні алгоритму. Наперед виконанні операції вставки а також пропуски на даному етапі враховуються стовпцем попереднім матриці R_2 і в разі необхідності можуть бути визначені з використанням операцій зсуву та побітової операції множення «AND», таким же чином як і для попередніх замін відповідних символів. Операція вставки в кінці рядка пошуку обчислюється шляхом проведення операції копіювання стовпчика, попереднього, матриці R_1 без проведення операції зсуву, а виконання операції пропуски обчислюється шляхом проведення операції зсуву нового стовпчика матриці R_1.

Розглянемо третій стовпець матриці R_2, даний стовбець враховує вставки, заміни а також пропуски, і мав би виглядати наступним чином - 11110. В даному випадку четверта одиниця відповідає за співпадання підрядка «vivi» з префіксом шаблону «viv» пропуском останнього символу рядка пошуку *i*, дана ситуація визначається операцією зсуву третього стовпчика матриці R_1. Друга одиниця третього стовбця матриці R_2 отримана наступним чином: зі співпадання підрядка «vi» з префіксом шаблону «viv» після виконання операції додавання v, яка виконується операцією копіювання другого стовбця матриці R_1, або також

одиниця може бути отримана зі співпадання підрядка «vi» з префіксом шаблону «v» з третьої позиції, при цьому необхідно відкинути останній символ *i*.

Для реалізації операції знаходження всіх входження пошукового шаблону шаблону з одним неспівпадаючим, з рядка пошуку, символом необхідно обчислити матрицю R_2 з використанням даних матриці R_1 за наступною послідовністю виконання операцій.

Позначимо через $R[j]$ відповідний стовпець матриці R з номером j і отримаємо $R^T[j]=\left(r(1,j),...,r(k,j),...,r(n,j)\right)$, де *n*-довжина пошукового пошуку. Для обчислення матриці R_2, необхідне виконання виконати наступних умов: обчислити стовбці проміжної додаткової матриці R_2^*, з використанням операції зсуву відповідних стовпців матриці *R*:

$$R_2^*[j]=\left(>> R[j-1]\right) \qquad (2.10)$$

де >>- операція зсуву для якої виконується наступна умова

$R^T[j]=\left(r(1,j),...,r(k,j),...,r(n,j)\right)$, то

$>> R^T[j]=\left(r(0,j),...,r(k-1,j),...,r(n-1,j)\right)$, при цьому для всіх $\forall j\left(1\le j\le m\right)$ будемо вважати $R[0,j]=1$, $\forall i\left(0\le i\le n\right)$ $R[i,0]=0$; виконати операції обчислення над елементами матриці R_2^* з використанням формулу (2.9):

$$R_2^*[i,j]=\begin{cases}1, & якщо\ R_2[i-1,j-1]=1\ i\ p_i=t_j\\ R_2^*[i,j], & якщо\ R_2[i-1,j-1]\neq 1\end{cases} \qquad (2.11)$$

при цьому будемо вважати наступне $R_2[0,j]=1,\ R_2[i,0]=0,\ \left(j=\overline{1..m},\ i=\overline{1..n}\right)$.

В результаті проведеного експерименту виявлення входжень пошукового шаблону в рядок пошуку, з використанням алгоритму наближеного пошуку, на основі використання арифметичних операцій вставки, заміни і пропуску в змозі виконати обчислення і при цьому виявити всі входження шаблону в рядок пошуку. Крім того, в разі необхідності, для того щоб при роботі алгоритму

допускалося в результаті пошуку більше однієї помилки, для вирішення цієї задачі необхідно ввести додаткові матриці на кожну помилку і при цьому необхідно провести аналогічні обчислення для перетворення таблиці в іншу, дозволяє виконувати наближений пошук будь-якого виразу, з врахуванням помилок чи провести точний пошук.

2.4 Висновки

Отримані в ході проведеного обчислення результати наближеного пошуку, а також в разі необхідності виконання точного пошуку отримали наступні висновки:

1. Запропонована модель представлення релевантності подібності рядків на основі розглянутих алгоритмів порівняння підрядків пошукового шаблону та рядка пошуку. Отримані результати задовільняють роботу алгоритмів пошуку, як по ефективності, так і за швидкістю отримання результатів пошуку кінцевими користувачами. Особливість отриманих результатів характеризується достатньою стабільністю в точності, як результат надається можливість створення модифікацій алгоритмів наближеного пошуку інформації з метою розширення області виконуваних задач повноти.

2. Запропонована модель наближеного пошуку при опрацюванні пошукових рядків. Надає можливість в разі необхідності, щоб при роботі алгоритму допускалося в результаті пошуку більше однієї помилки, для вирішення цієї задачі необхідно ввести додаткові матриці на кожну помилку і при цьому необхідно провести аналогічні обчислення для перетворення таблиці в іншу, дозволяє виконувати наближений пошук будь-якого виразу, з врахуванням помилок чи провести точний пошук.

3 РОЗРОБКА АЛГОРИТМІВ НАБЛИЖЕНОГО ПОШУКУ ІНФОРМАЦІЇ В БАЗАХ ДАНИХ

3.1 Розробка алгоритму оптимізації запису інформації в бази даних

Розглянемо принцип роботи алгоритму оптимізації запису інформації в бази даних:

1. Вхідними даними роботи алгоритму - масив інформації, яку необхідно записати в базу даних з умовою оптимізації інформації на предмет дублювання даних.

2. Обчислюємо відстані Левенштейна, визначаємо релевантність кожного рядка масиву з інформацією в базі даних, виконуємо оптимізації текстової інформації при цьому відкидаємо неприйнятні варіанти.

3. Якщо значення відстані Левенштейна (релевантності) вище межі автоматичної границі ідентифікації (Γ_a), то вхідний рядок, поточний вноситься в лог для видалення.

4. Якщо значення відстані Левенштейна (релевантності) нижче межі ручної ідентифікації (Γ_p), то рядок вноситься в лог для запису.

5. Якщо значення релевантності вище ручної ідентифікації (Γ_p), але при цьому нижче автоматичної границі ідентифікації (Γ_a), то такий рядок відправляється в лог прийняття рішень для подальшої обробки експертом.

6. Якщо отримане значення релевантності рядка нижче ручної ідентифікації (Γ_p), в даному випадку рядок новий і йде його запис в БД.

Джерела внесення та зміни затребуваної інформації в базу даних це введення інформації кінцевими користувачами, а також імпорт даних з різних зовнішніх джерел. У випадку внесення та зміни затребуваної інформації в базу даних кінцевими користувачами, необхідно забезпечити, в даному випадку, мінімальний час роботи пошукової системи, на даному етапі система повинна працювати, за

рахунок зменшення результату точності визначеності релевантності, гранично швидко. Тому, в даній ситуації, граничні межі ідентифікації Γ_a, Γ_p можуть бути змінені при необхідності, для забезпечення при цьому необхідної швидкості пошуку релевантної інформації. Так як Алгоритми наближеного пошуку інформації під час роботи не можуть надати гарантії забезпечити задану точність, тому при використанні кінцевими користувача системи, надається можливість ігнорувати підказку системи і на наступному кроці підтвердити запис інформації в БД. При вирішенні задачі оптимізації запису інформації в БД кінцевими користувачами виділимо наступні етапи: провести оптимізацію інформації на предмет дублювання, на рівні введення кінцевими користувачами інформації та її відхилення при необхідності; провести оптимізацію інформації на предмет дублювання, використовуючи обчислення релевантності і аналізу введеної інформації відповідно до граничні межі ідентифікації Γ_a, і автоматичне видалення інформації яка не задовольняє межі ідентифікації Γ_a; аналіз та обробка інформації кінцевим користувачем значення релевантності якої вище ручної ідентифікації (Γ_p), але при цьому нижче автоматичної границі ідентифікації (Γ_a).

В основі реалізації системи оптимізації запису інформації в бази даних покладений алгоритм порівняння підрядків рядка пошуку з пошуковим шаблоном, даний алгоритм покладений в основу визначення релевантності порівнюваних рядків. На першому етапі роботи системи оптимізації інформації, рядки, які обробляються, об'єднуються в один. Наприклад, слова прізвище, ім'я, та по батькові обробляються як один загальний рядок "ПІБ". На другому етапі результати пошуку першого уточняються, шляхом обчислення релевантності вже для окремих рядків пошуку. Наприклад, якщо виникла наступна ситуація коли два рядки які містять інформацію про клієнта співпадають з коефіцієнтом релевантності більше ніж на 90%, але при цьому роботі алгоритму з прізвищами клієнта отриманий коефіцієнт до 53%, то в даному випадку можна відкинути

отримані результати при першого етапу. Алгоритму оптимізації запису інформації в бази даних представлений на рис. 3.1.

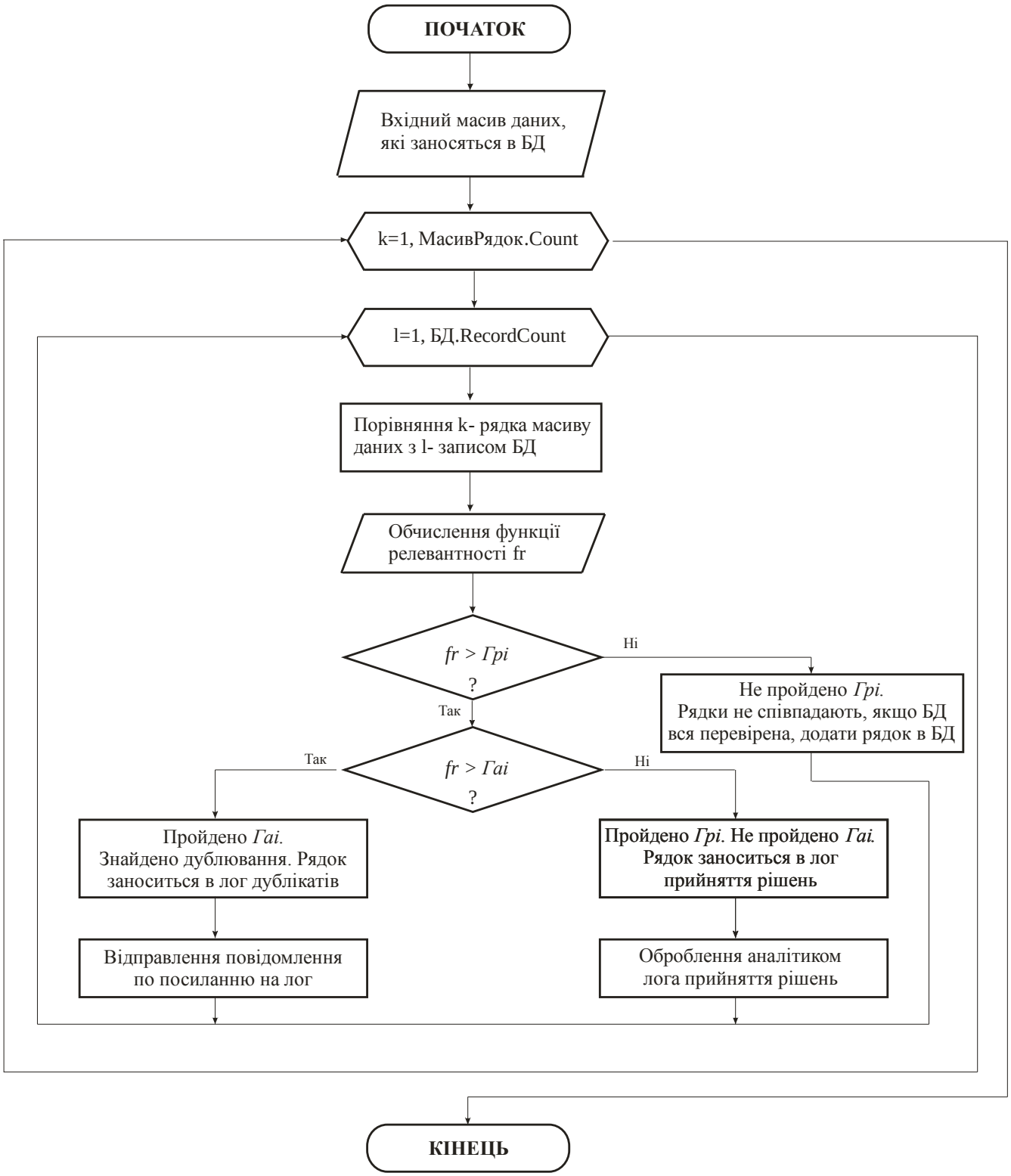


Рисунок 3.1 - Алгоритм оптимізації запису інформації в бази даних

Використання методу *N*-грам при обчисленні релевантності подібності двох рядків надає для обробки досить об'ємну множину підрядків (префіксів), що може бути досить суттєвих недоліком при проведенні відповідних обчислень. Таким чином, при максимальній довжині шаблона, наприклад $N = 4$ та при обчислені множини рядків з середньою довжиною рядка L, буде сформовано наступну множину рядків, кількість яких складе - $M = L\text{-}N + 1$. Наведемо наступний приклад, нехай в базі даних зберігається інформація по 100 000 клієнтів, при цьому середня довжина рядка інформації складає 30 символів, в даному випадку отримаємо наступну множину підрядків $M = K \cdot (L - N + 1) = 100\ 000 \cdot (30 - 4 + 1) = 2\ 700\ 000$ елементів, де K – кількість записів в базі даних. Хоча, кількість елементів унікальних, в даному випадку буде значно менша. В даному випадку буде складно організувати ефективний швидкий пошук необхідної інформації в базі даних. Для ефективного вирішення поставленої задачі, при обчисленні довгих рядків виникає потреба в збільшенні довжини пошукового шаблону.

Алгоритм оптимізації запису інформації в бази даних забезпечує: збереження оптимізованої інформації в базі даних, виключаючи при цьому дублювання інформації, гарантує цілісність інформації, а також забезпечує зниження зашумленість потоку даних, результатом яких є помилки кінцевого користувача при введенні необхідної інформації в базу даних операторського введення; забезпечує налаштування правил як автоматичний режимі, так і забезпечує ручний режим, втручанням адміністратора, при виникненні особливих ситуацій в складних випадках. В разі необхідності надається можливість редагування реквізитів клієнта в плані присвоєння їм нових значень ваг, при цьому змінюючи поріг ідентифікації відповідних правил.

3.2 Розробка алгоритму ідентифікації особистості в базах даних

Однією з задач яку необхідно вирішити при роботі з інформацією яка відноситься до особистості, це вирішення задачі її однозначної ідентифікація. Для вирішення задачі ідентифікації особистості в базах даних використати метод порівняння її основних реквізитів. Використання такого підходу для вирішення поставленої задачі, не завжди є достатнім, так як по ряду причин, реквізити однієї особистості, які взяті з двох баз даних, в даному випадку можуть не співпадати, по ряду причин: інформація про особистість не завжди доступна (помилки в базах даних та інших джерелі даних); не всі категорії особистостей мають відповідно повні реквізити (страхове свідоцтва, ідентифікаційний код, код пенсійного фонду); реквізити документа особистості можуть бути змінені, в результаті втрати\псування\ чи також що можливе за бажанням особистості; реквізити документа особистості також можуть відрізнятися також внаслідок помилки під час запису в базу даних. Для вирішення даної задачі з врахуванням розглянутих поправок на сьогодні не існує готової методики, по такому типу ідентифікації особистості.

На основі проведеного дослідження та виконаних експериментів пропонується рішення яке дозволяє на сьогодні проводити ідентифікацію особистостей в базах даних з максимально допустимою точністю. На основі отриманих результатів проведеного дослідження запропонований підхід, із застосуванням якого надається можливість організувати ефективний інформаційний обмін даними по особистостям. Розглянемо укрупнений алгоритм ідентифікації особистості в базах даних розроблений на основі запропонованої технології. Алгоритм ідентифікації особистості в базах даних складається з трьох наступних основних блоків (рис.3.2): формування інформаційного масиву елементами якого є «подібність» особистостей; використання підходу обробки масиву «подібності» особистостей на основі використання підходу несуворої відповідності; відпрацювання виняткових ситуацій під час робити системи.

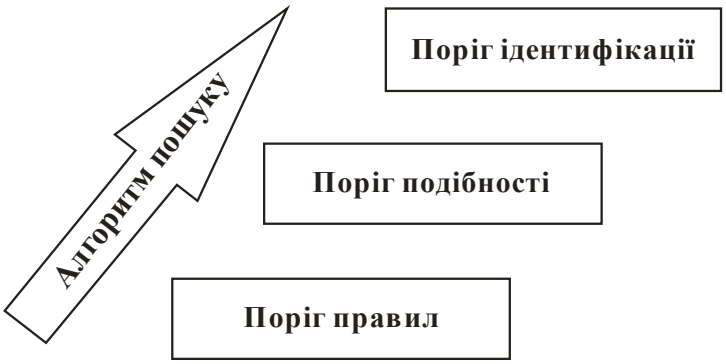


Рисунок 3.2- Укрупнена схема ідентифікації особистості в базах даних

Розглянемо основні визначення які використовуються в основі робити алгоритму ідентифікації особистості в базах даних: вага рядка інформації - умовний коефіцієнт реквізиту, значення реквізиту вага залежить від достовірності, повноти, а також актуальності використовуваного на даному етапі реквізиту. Умовний коефіцієнт реквізиту - вага визначає значимість використовуваного реквізиту при роботі системи для ідентифікації особистості. Наприклад. реквізит «місце народження» відповідає меншою достовірністю і повнотою, ніж реквізит наприклад «прізвище» і, таким чином в даному випадку може мати меншу «вагу». В табл. 3.4 для прикладу наведені реквізити з відповідними «вагами».

Таблиця 3.3 – Призначення ваг реквізитам

Реквізит	Вага
«Прізвище»	5
«Ім'я» і «Дата народження»	4
«По батькові» і «Місце народження»	3

Розглянемо наступне поняття яке буде використовуватися алгоритмом ідентифікації особистості в базах даних - «правило» - поєднання всіх реквізитів особистості, за якими буде здійснюватися алгоритмом пошук необхідної інформації. Алгоритмом ідентифікації особистості який використовує в своїй роботі правила правилами наступний: при ідентифікації особистості використовуються ті реквізити, які описані та вказані в наведених правилах. Наприклад, при використанні для ідентифікації особистості «правила 1» в даному

випадку будуть порівнюються тільки «Ім'я», «Прізвище» а також «Дата народження», при цьому інші реквізити при ідентифікації особистості враховуватися не будуть. Для кожного правила, визначених в базі правил, яка використовується алгоритмом ідентифікації особистості, визначається його сумарна «вага», яка дорівнює сумі «ваг» кожного реквізита даного (табл.. 3.4).

Таблиця 3.4 – Визначення сумарної ваги правил

№ правила	Реквізити	Сумарна вага реквізитів	Поріг ідентифікації по правилу
Правило 1	Прізвище (5) + Ім'я (4) + По батькові (3)	12	11
Правило 2	Прізвище (5) + Ім'я (4) + Дата народження (4)	13	12
Правило 3	Ім'я (4) + По батькові (3)+ Дата народження (4)	11	10
Правило 4	Дата народження (4) + По батькові (3) + Місце народження (3)	10	9

Для ідентифікації особистості з обчисленого масиву «подібності» людей, при пошуку інформації в БД встановлюється так званий поріг ідентифікації. Поріг ідентифікації використовується, в разі необхідності видалити особистість, яка не відповідає значенню порогу ідентифікації. В тому випадку, якщо заданий поріг ідентифікації пройшли більше однієї особистості, то вирішити виниклу проблему без втручання кінцевого користувача не можливо. Наступним кроком роботи алгоритму ідентифікації особистості - конкретне визначення особистості, вирішується порівнянням отриманого значення релевантності реквізитів документа. В табл. 3.5 наведені результати простого порівняння реквізитів, роботи алгоритму ідентифікації особистості, помилки в імені особистості, призвели до того що знайдена особистість не переходить поріг ідентифікації ні по одному з наведених правил. Отримане значення релевантності, використовуваним алгоритмом, як бачимо в табл. 3,5 має досить низький показник по реквізиту «імені» - 46,5%. Даний результат отриманий з (2.3) пов'язаний з тим що (2.3)

використовує як і при порівняння реквізита «адрес» використовується при цьому шаблон *P*– максимальної довжини при $N = \{1...4\}$, середня довжина реквізитів, які використовуються для ідентифікації особистості, як правило менше середньої довжини реквізитів використовуваних по адресам фірм, в результаті складова (2.3) при використанні максимальної довжини шаблона $N = 4$ зменшує релевантність реквізитів, які використовуються для ідентифікації особистості. Таким чином при використані (2.3) доцільно використовувати шаблон *P*– максимальної довжини на діапазоні $N = \{1...3\}$.

Таблиця 3.5 - Результати простого порівняння реквізитів

Масив «подібності» особистостей	Прізвище	Ім'я	Дага народження	По батькові	Місце народження
	Іванов	Петор	03.03.1976	Сергійович	Київ
БД	Іванов	Петр		Сергійович	Київ
Вага	5	4	4	3	3
R	100%	46.5%	0%	100%	100%
R* Вага	5	1.86	0	3	3
Правило 1 9.86/11	5	1.86		3	
Правило 2 6.86/12	5	1.86	0		
Правило 3 4.86/10		1.86	0	3	
Правило 4 6/9			0	3	3

В основу роботи алгоритму ідентифікації особистості в базах даних покладена умова:

$$\sum_{i=1}^{n} p_j(i) \cdot (R_i \cdot w_i + L_i) \geq k_j; \qquad j = \overline{1,m}; \qquad i = \overline{1,n}; \tag{3.1}$$

де $p_j(i)$ - реквізит правила, яке використовується для ідентифікації особистості: $p_j(i) = 1$, в тому випадку якщо *i*-ий реквізит входить в *j*-е правило ідентифікації особистості; $p_j(i) = 0$, якщо *i*-ий реквізит не входить в *j*-е правило ідентифікації особистості; R_i - результат отриманої релевантності від *i*-их

реквізитів; w_i - вага i-го реквізиту, значення ваги i-го реквізиту визначається в залежності від достовірності, повноти а також актуальності реквізиту; L_i - поправочний коефіцієнт, обчислюється в залежності від відстані Левенштейна між реквізитами; k_j - встановлений поріг ідентифікації особистості для заданого правила, поріг ідентифікації використовується, в разі необхідності видалити особистість, яка не відповідає зазначеному порогу ідентифікації. У випадку, якщо заданий поріг ідентифікації пройшли більше однієї особистості, то вирішити виниклу проблему без втручання оператора не представляється можливим; m - кількість правил, використовуваних для ідентифікації особистості; n - кількість реквізитів, які беруть участь в формуванні правил.

Запропонований алгоритм ідентифікації особистості в базах даних дозволяє вирішити наступні задачі: виконати ідентифікації особистості в базах даних, і може бути використана при первинному об'єднанні інформації накопичених відомчих баз даних, також при створенні та супроводі реєстрів населення; здійснювати об'єднання записів бази даних, відсоток подібності (релевантності), яких вище встановленої границі; при цьому надає гарантії забезпечити інформаційну цілісність та несуперечливість інформації, а також при цьому знизити зашумленість даних, обумовлених внесенням помилок, при введенні інформації кінцевим оператором;.

3.3 Розробка алгоритму пошуку інформації в базах даних за реквізитами особистості

Роботу алгоритму пошуку інформації в базах даних за реквізитами особистості можна розбити на три етапи:

1. Пошук інформації в базах даних за реквізитами особистості, довжина яких більше 50 символів.

2. Пошук інформації в базах даних за реквізитами особистості з середньою довжиною менше 50 символів: назви вулиць, назви місць роботи, юридичних осіб, торговельних точок, електронних адрес, телефонів, факсів.

3. Пошук інформації в базах даних за реквізитами особистості а саме за прізвищами або отриманням «на льоту» інформацію про клієнтів кредитним інспектором.

Блок фонетичної подібності під час роботи алгоритму пошуку інформації в базах даних за реквізитами особистості, можна використовувати як в разі ручному введенні необхідної інформації в базу даних, під час внесення реквізитів клієнта, так і при пошуку необхідної інформації за короткими реквізитами. В даному випадку відстань Левенштейна реквізитами клієнта доцільно використати в якості ранжируючої функції при виведенні результатів інформації. Запропонований алгоритм наближеного пошуку інформації в базах даних в даній ситуації може застосуватися до вирішення першої задачі.

Розглянемо більш детально роботу алгоритму пошуку інформації в базах даних за реквізитами особистості. В разі коли інформація про клієнта заноситься в базу даних безпосередньо оператором і при цьому включений блок наближеного пошуку, паралельно операції введення даних реквізитів клієнта, проводиться аналіз інформації, введеної в спеціальне поле інтерфейса, і формується при цьому попередній масив подібної інформації про клієнтів, яка вже є в базі даних. Таким чином в міру заповнення форми інформації по даному клієнту попередній масив подібності інформації про клієнтів аналізується з використанням релевантності та урізається при цьому відкиданням записів з масиву, яким не вдалося пройти поріг ідентифікації.

Робота алгоритму пошуку інформації в базах даних за реквізитами особистості представлена на рис. 3.3.

При роботі алгоритму пошуку інформації в базах даних за реквізитами особистості, пошук може здійснюватися за будь - якими реквізитами і при цьому може бути включений/виключений блок наближеного пошуку, на початку роботи

алгоритму необхідно визначати, які реквізити клієнта будуть брати участь в пошуковому запиту, якщо при цьому прізвище бере участь пошуковому запиту, то за допомогою спеціальної функції ця інформація перетворюється і за її допомогою здійснюється попередній пошук в базі даних, куди адресований пошуковий запит.

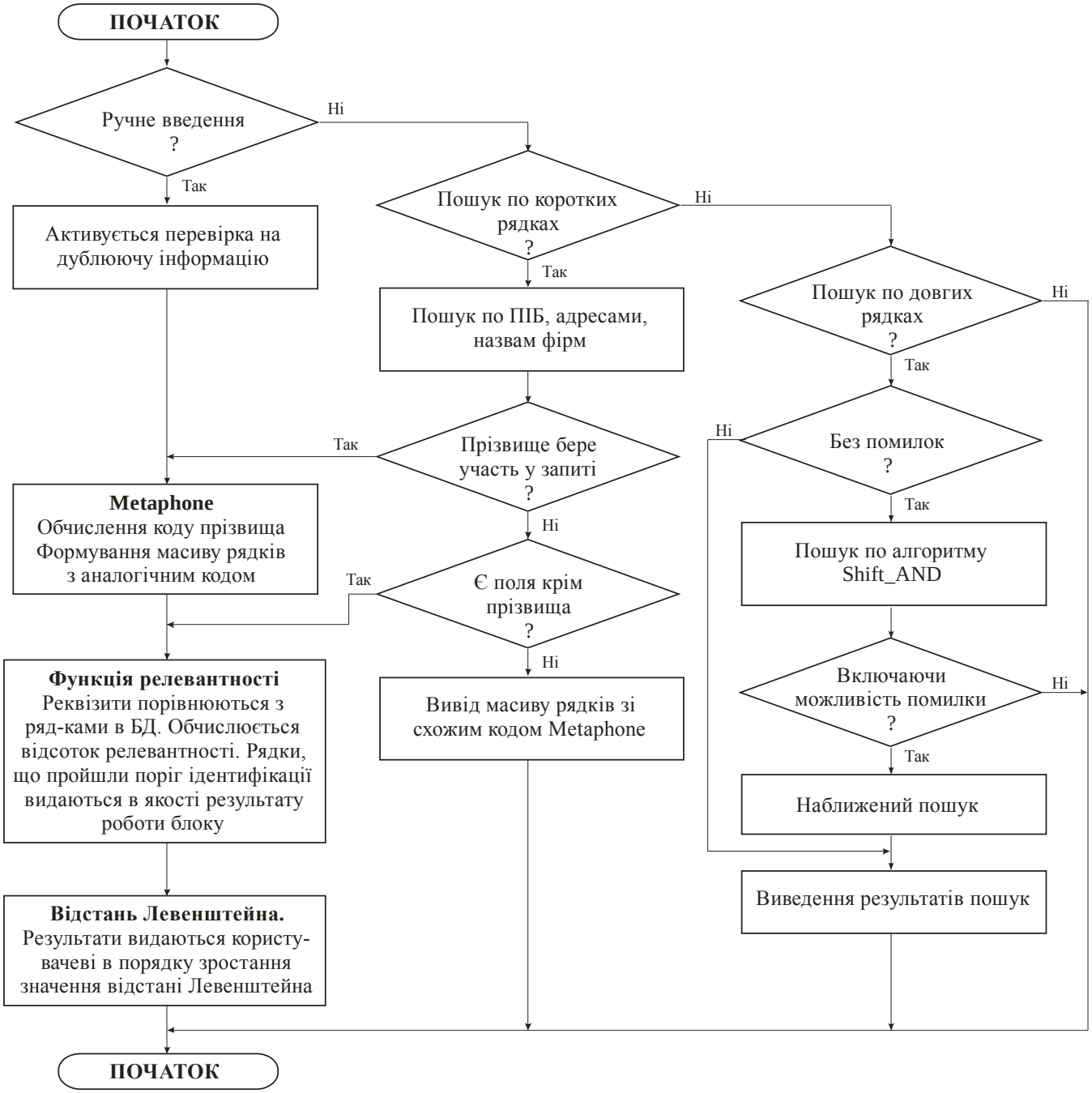


Рисунок 3.3 - Алгоритм пошуку інформації в базах даних за реквізитами особистості

Отримані результати попереднього пошуку формують масив подібності інформації про клієнтів, аналізується з використанням релевантності та урізається, при цьому, відкиданням записів з масиву, яким не вдалося пройти поріг ідентифікації. Отримана інформація в відсортованому в порядку спадання по відстані Левенштейна.

3.4 Висновки

Розроблено алгоритм оптимізації запису інформації в бази даних, використовує значення релевантності подібності рядків, які порівнюються, також дозволяє надавати рішення в даній ситуації в найбільш загальному вигляді. Запропонований алгоритм надає можливість об'єднання рядків, які задовольняють заданому відсотку подібності за вказаним набором реквізитів, вага яких відповідає встановлених межах.

Розроблено алгоритм ідентифікації особистості в базах даних з використанням набору правил ідентифікації, дозволяє оцінити ступінь подібності інформації занесеної в базу даних про клієнтів, система правил і ваг є основою для формування правил ідентифікації особистості, встановлення їхнього порогу ідентифікації, в залежності від призначених ваг реквізитам.

Розроблено алгоритм пошуку інформації в базах даних за реквізитами особистості, в основі якого використовується значення релевантності, отримане в результаті реалізації запропонованої моделі релевантності подібності рядків подібності рядків.

4 ВИКОРИСТАННЯ АЛГОРИТМІВ НАБЛИЖЕНОГО ПОШУКУ ІНФОРМАЦІЇ НА ПІДПРИЄМСТВАХ

4.1 Розробка системи оптимізації інформації при запису в бази даних

Базою для розробки системи «Оптимізації інформації при запису в бази даних» послужили моделі та алгоритмів наближеного пошуку та оптимізації запису інформації в базах даних

Головне призначення системи «Оптимізації інформації при запису в бази даних» провести оптимізацію інформації на предмет дублювання, на рівні введення кінцевими користувачами інформації та її відхилення при необхідності. Запропонована система «Оптимізації інформації при запису в бази даних» інтегрується системи управління базами даних, таких як інтегрується із засобами Oracle 12c, Microsoft SQL Server. система «Оптимізації інформації при запису в бази даних» надається кінцевому користувачу у вигляді дистрибутива, в склад якого входить виконуваний файл, з розширенням .exe та без проблем інсталюється на персональний комп’ютер кінцевого користувача в автоматичному режимі та в необхідній конфігурації. Для успішного функціонування системи «Оптимізації інформації при запису в бази даних» необхідно попередньо встановити на персональний комп’ютер кінцевого користувача відповідне програмного забезпечення:

- ліцензійна версії операційної системи (ОС) Windows 2010 / 2007;
- клієнт системи управління базами даних Microsoft SQL Server 2019.

При цьому також рекомендується також установка останніх оновлень кінцевим користувачем операційної системи Service Pack а також відповідних драйверів, рекомендованих компаніями-виробниками операційних систем та програмно-апаратних засобів. Умови експлуатації системи «Оптимізації інформації при запису в бази даних» в повному обсязі задовольняють вимогам, які в даному випадку пред'являються до персональних комп'ютерів в плані засобів

та мов їх експлуатації. Функціональні обмеження, які можуть бути присутніми при експлуатації системи «Оптимізації інформації при запису в бази даних», в кінцевому рахунку будуть визначатися версією операційної системи і також, що є немаловажним фактором, характеристиками персонального комп’ютера кінцевого користувача.

Рівень підготовка кінцевого користувача запропонованої системи, повинен відповідати рівню досвідченого користувача операційної системи Windows 2010, використовуваних додатків Microsoft Office 2017, систем управління базами даних, в нашому випадку СУБД Microsoft SQL Server 2019. При цьому також необхідно враховувати системні вимоги до робочого місця кінцевого користувача, на якому буде встановлена системи «Оптимізації інформації при запису в бази даних», інтерфейс управління програмно-апаратним комплексом, персональний комп'ютер повинен задовільнять наступним характеристикам: HDD-диск WD Caviar 1000GB WD10EZRZ, оперативна пам'ять, комплект Kingston HyperX DDR4 2x4Gb 2666GHz (HX426C15FBK2/8), процесор AMD Ryzen 3 2200G BOX материнска плата Asus A320M-K.

На стороні сервера, при використані системи «Оптимізації інформації при запису в бази даних» інсталюється серверна частина програмного комплексу, із зазначенням операторів а також використовуваних при цьому проміжних баз даних. Кінцевий користувач, системи «Оптимізації інформації при запису в бази даних» повинен мати необхідні права на вибірку відповідних даних, редагування, додавання, модифікацію та видалення.

Розглянемо архітектуру програмного комплексу запропонованої системи. Під час експлуатації системи «Оптимізації інформації при запису в бази даних» інформація надходить з зовнішніх джерел - комерційних організацій, державних органів та інформаційних партнерів. Отримана інформація від зовнішніх джерел форматується у відповідності до заданих шаблонів і на першому етапі зберігається в проміжній базі даних акумулюються в проміжній базі даних. На наступному етапі вступає в роботу запропонована системи «Оптимізації інформації при

запису в бази даних» задача якої оптимізація інформації яка знаходиться в проміжній базі даних, а також перенесення обробленої інформації в основну базу даних. Архітектура системи «Оптимізація інформації при запису в бази даних» представлена на рис. 4.1.

Програмний комплекс системи «Оптимізація інформації при запису в бази даних» використовує при роботі наступні основні блоки:

1. Блок який відповідає за розподіл інформації на паралельні потоки даних. Основне призначення даного блоку забезпечити достатньо високу швидкодії обробки інформації.

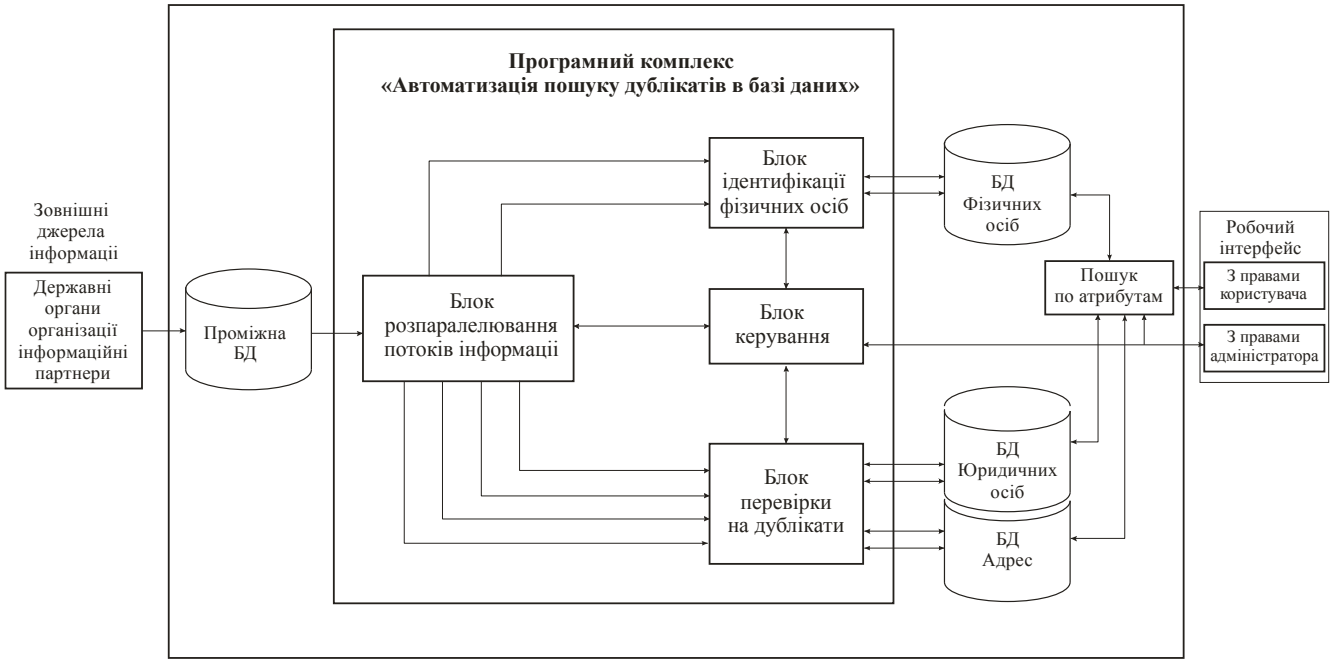


Рисунок 4.1 - Архітектура системи
«Оптимізація інформації при запису в бази даних»

2. Блок оптимізації інформації на предмет дублювання. Блок оптимізації інформації - пакет відповідних процедур та функцій, які використовуються, в разі необхідності, блоком який відповідає за розподіл інформації на паралельні потоки даних, для кожного з потоків даних ізольовано. Блок оптимізації інформації працює відповідно до алгоритму оптимізації запису інформації в бази даних, який розглянутий в третьому розділі. Таким чином, результатом отримані, як наслідок роботи блоку – переносяться у відповідну базу даних, або відхиляються в разі

визначення їх як дублікати, і переносяться в спеціальний лог повторюємо інформації, а також якщо для прийняття конкретного рішення недостатньо підстав поповнення бази даних чи видалення, тоді отримана інформація переноситься в лог прийняття кінцевого рішення і знаходиться там до обробки адміністратором.

3. Блок ідентифікації особистості в базі даних. Робота блоку ідентифікації особистості в базі даних в загальному відповідає роботі блоку оптимізації інформації на предмет дублювання, відмінність роботи блоків полягає, у використанні алгоритму ідентифікації особистості, а також у використовуваних базах даних при перенесенні інформації.

4. Блок керування програмним комплексом системи «Оптимізація інформації при запису в бази даних». Керування програмним комплексом здійснюється через використання інтерфейсу кінцевого користувача, через блок керування програмним комплексом, представляється можливим в широкому діапазоні налаштовувати, роботу системи «Оптимізація інформації при запису в бази даних». У блоці, який відповідає за розподіл інформації на паралельні потоки даних, представлена можливість вибору кількості та налаштування потоків даних, а також розмір інформації у потоці. Для блоку ідентифікації особистості в базі даних і блоку оптимізації інформації на предмет дублювання, блок керування керування програмним комплексом надає інформацію про величину порогових коефіцієнтів для алгоритмів ідентифікації. Блок керування програмним комплексом містить відповідні механізми, процедури та функції, для забезпечення надійної роботи системи, відповідають за логіку обробки адміністратором інформація яка надійшла в лог прийняття рішень.

5. Блок пошуку інформації в базах даних за реквізитами особистості. Блок пошуку реалізований у вигляді окремої підсистеми, як основу блоку пошуку інформації за реквізитами становлять алгоритми, які запропоновані в третьому розділу та реалізовані в даному блоку у вигляді процедур та функцій.. Даний блок розміщений кермо від системи і встановлюється в разі потреби в якості надбудова.

Блок пошуку інформації за реквізитами особистості використовується в якості інформаційно- пошукової.

4.2 Проведення дослідження ефективності запропонованих алгоритмів

Програмний комплексом системи «Оптимізація інформації при запису в бази даних» впроваджений в автоматизовану інформаційно-пошукову систему «Відділ кредитних історій» підприємства. Одна із задач вирішення якої полягає на АІПС ВКІ перевірка анкетних даних реквізитів потенційного клієнта на наявність в базах даних системи «Оптимізація інформації при запису в бази даних». При вирішенні даної задачі використовується блок оптимізації інформації на предмет дублювання. Даний блок приймає участь в обробці інформації при кожному етапі перенесення інформації в базу даних. Під час роботи системи «Оптимізація інформації при запису в бази даних», одночасно працює підсистема збору статистичні даних по кожному блоку системи. Отриманні статистичні дані заносяться у зведені матриці (масиви інформації) і зберігаються на дисковому просторі відповідним чином. Для проведення аналізу роботи блоків програмного комплексу, проведена відповідна обробка отриманих статистичних даних в результаті якої побудовані відповідні графіки, що показують загальне відхилення виявлених кількості помилок та / або виявлену при роботі системи кількість інформації яка дублювалася протягом часу а також виявлення природи цих коливань, Проведений кластерний аналіз зібраних статистичних даних дозволив будувати графіки залежностей на вказаному діапазоні часу від кількості виявлених помилок та виконати порівняння з яким-небудь зазначеним відрізком часу у відповідності до джерел даних; торгової точки, кількості вхідних даних; відділення підприємства, представництва, при цьому масштаб задається адміністратором. Для бачення загальної картини змін, які відбувалися на протязі року, побудовані відповідні графіки на основі отриманих статистичних даних, витягнутих з масивів інформації, отримані графіки відображатимуть виявлену

кількості помилок та виявлену при роботі системи кількість інформації яка дублювалася протягом 2019 рік в масивах інформації. Залежність виявлених кількості помилок та залежність виявлену при роботі системи кількості інформації яка дублювалася від кількості оброблених вхідних даних, протягом 2019 рік по масивам інформації представлені на графіках наведених на рис. 4.2 - 4.3.

Проведемо аналіз отриманих графіків за відповідні проміжки часу на предмет залежностей, а також виявлення подібних інтервалів графіків та їх екстремумів. Як бачимо з наведених графіків, можна виділити загальні характерні риси в поведінці отриманих графіків, бачимо падіння показників кількості дублікатів, а також помилок за жовтень місяць. Коливання показників кількості дублікатів і помилок за жовтень місяць пов'язане, в першу чергу, з постійним проведенням експериментів щодо зміни порогів, використовуваних правил, змін ваг реквізитів, які входять в правила, в режимі автоматичної та ручної ідентифікації в діапазонах 83% - 95% і 35% - 47% відповідно, що суттєво вплинуло на показники виявлених кількості дублікатів і кількості.

Наведені графіки на рис. 4.2 і рис. 4.3 показують збільшення кількості помилок і кількості дублікатів до кінця квітня, далі, як показують графіки йде зниження кількості помилок і кількості дублікатів цього рівня майже вдвічі, це пов’язано з прийнятими на підприємстві відповідних заходів щодо покращання трудової дисципліни в колективі. Результатом введених на підприємстві відповідних заходів стала стабілізація показників виявлених кількості помилок і кількості дублікатів на представлених діапазонах 0,6% - 0,8%, проти 1% - 1,2% до введених на підприємстві відповідних заходів, щодо покращання трудової дисципліни. При подальшому аналізу графіків (рис. 4.2 і рис. 4.3), бачимо помітну різницю в темпах зниження абсолютних і процентних показників кількості помилок і кількості дублікатів в наведених графіках за період в діапазоні з травня по серпень 2019року. Отримані результати за період в діапазоні з травня по серпень 2019року пов'язані з жорсткістю прийнятих правил скорингу, і як результат загальне число, виданих організаціям кількості пакетів, впало, що на

фоні зниження абсолютних і процентних показників кількості помилок і кількості дублікатів і не дало суттєво знизитися відсотковим показникам помилок.

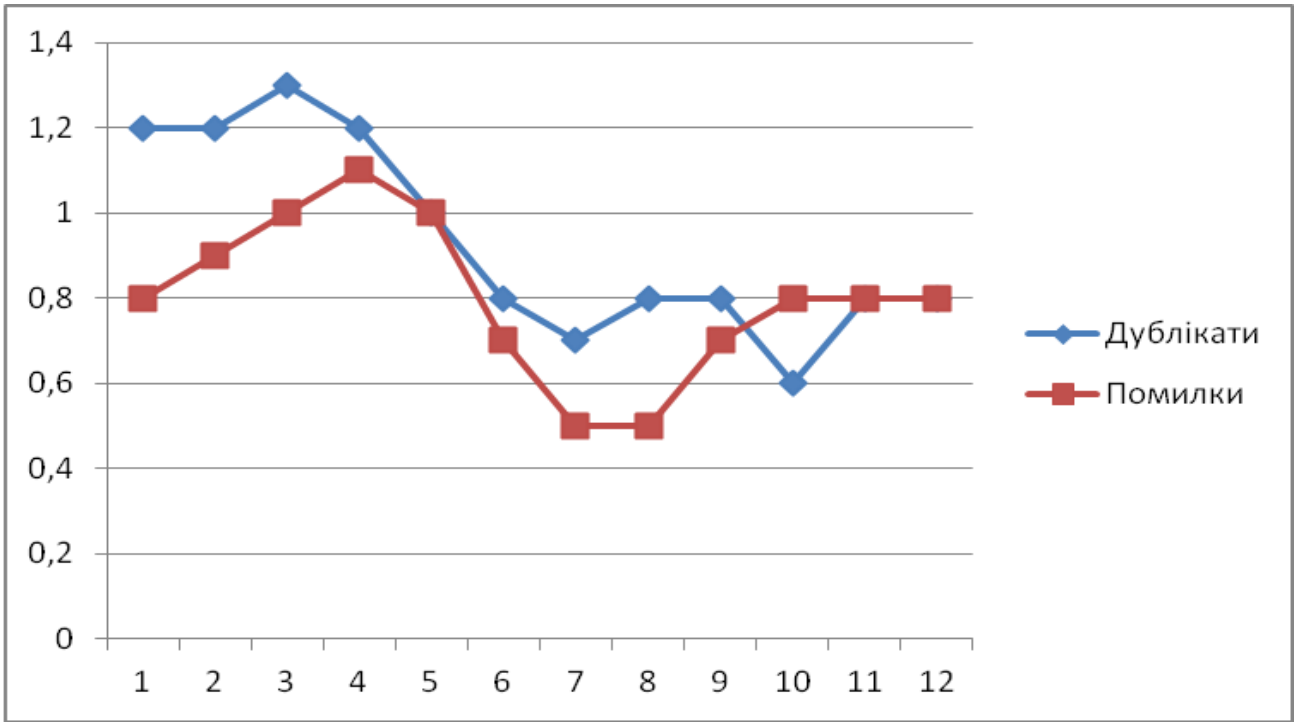


Рисунок 4.2 - Відсоток дублікатів і помилок

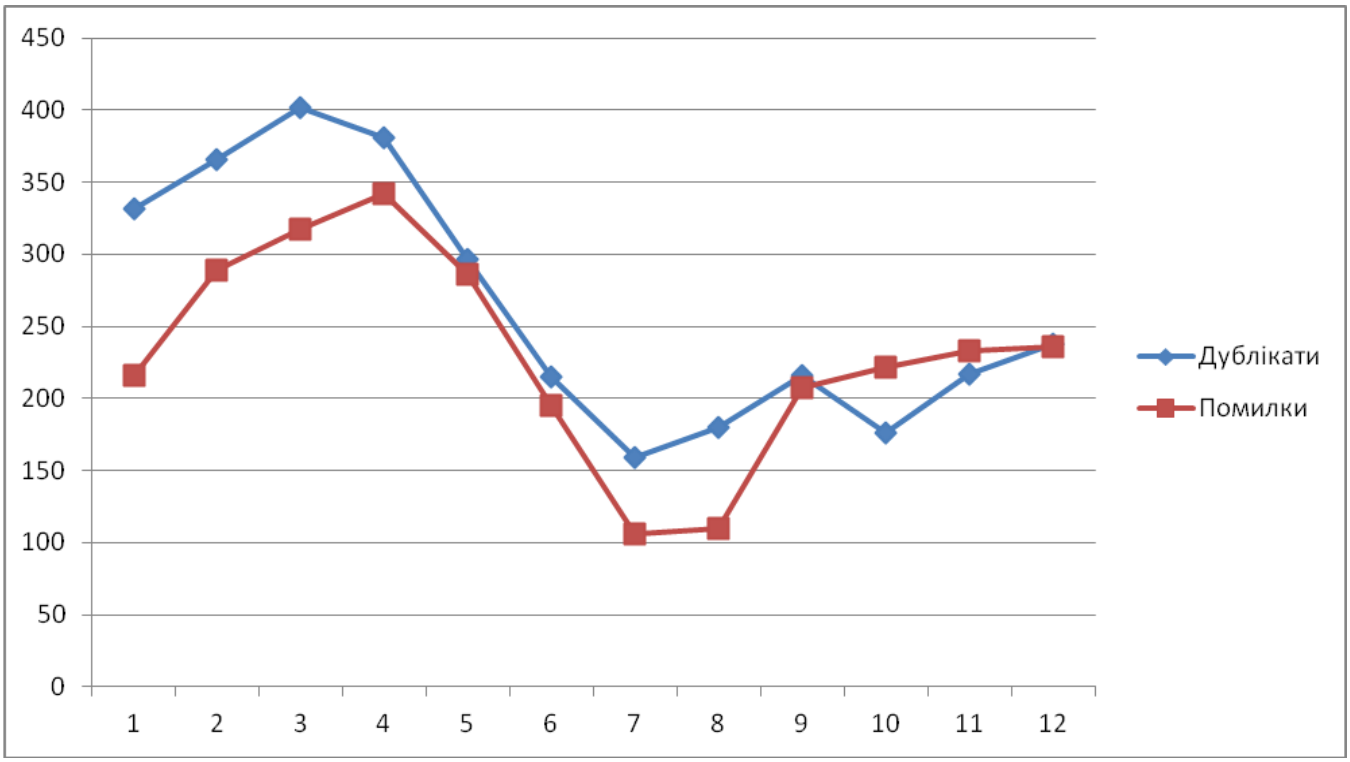


Рисунок 4.3 - Кількість дублікатів і помилок

Для проведення, в даній ситуації, оцінки загальної ефективності від впровадження програмного комплексу системи «оптимізація інформації при запису в бази даних» будемо використовувати показник знаходження кількості помилок під час роботи з використовуваної базою даних. Для визначення ефективності бралися для порівняння показники за 2018 і 2019 роки В результаті проведеного обчислення відібраних даних, отримані дані, що відображають загальну позитивну тенденцію використання системи «оптимізація інформації при запису в бази даних». Так, оптимізація інформації в базі даних, з використанням програмного блоку оптимізації запису інформації в бази даних, призвело до зниження показника зашумленості даних в базі даних уже січні 2019 року, становить 2,47%, а максимального значення показника зашумленості даних отримано в жовтні, і його значення стало 11,18%, таким чином значення показника зашумленості даних знизилось майже в 3,5 рази по відношенню до показника зашумленості даних до відповідного періоду 2018 року. Таким чином, результатом використання програмного блоку оптимізації запису інформації в бази даних, стало зниження зашумленості даних, середнє значення якого за рік склало - 5,3%.

4.3 Висновки

При обробці інформації в БД отримані результати, що свідчать про загальну позитивну тенденцію зниження рівня зашумленості даних. Так, зниження показника зашумленості даних вже в перший місяць склав - 2,47%, а максимума досягло в жовтні - 11,18%, тобто зменшився майже в 3,5 рази. Середній показник зниження загубленості за рік склав 5,3%. Розроблені алгоритми під час виконання дипломної роботи, можуть бути використання в якості типових рішень при проектуванні та застосуванні подібних систем для підприємств середнього та малого бізнесу.

ВИСНОВКИ

Основні результати, отримані в ході наукового дослідження, полягають у наступному:

1. Запропонована модель представлення релевантності подібності рядків на основі алгоритмів порівняння підрядків пошукового шаблону та рядка пошуку. Отримані результати задовільнять роботу алгоритмів пошуку, як по ефективності, так і за швидкістю отримання результатів пошуку кінцевими користувачами. Особливість отриманих результатів характеризується достатньою стабільністю в точності, як результат надається можливість створення модифікацій алгоритмів наближеного пошуку інформації з метою розширення області виконуваних задач.

2. Запропонована модель наближеного пошуку при опрацюванні пошукових рядків. Надає можливість в разі необхідності, щоб при роботі алгоритму допускалося в результаті пошуку більше однієї помилки, для вирішення цієї задачі необхідно ввести додаткові матриці на кожну помилку і при цьому необхідно провести аналогічні обчислення для перетворення таблиці в іншу, дозволяє виконувати наближений пошук будь-якого виразу, з врахуванням помилок чи провести точний пошук.

3. Розроблено алгоритм оптимізації запису інформації в бази даних, використовує значення релевантності подібності рядків, які порівнюються, також дозволяє надавати рішення в даній ситуації в найбільш загальному вигляді. Запропонований алгоритм надає можливість об'єднання рядків, які задовольняють заданому відсотку подібності за вказаним набором реквізитів, вага яких відповідає встановлених межах.

4. Розроблено алгоритм ідентифікації особистості в базах даних з використанням набору правил ідентифікації, дозволяє оцінити ступінь подібності інформації занесеної в базу даних про клієнтів, система правил і ваг є основою для формування правил ідентифікації особистості, встановлення їхнього порогу ідентифікації, в залежності від призначених ваг реквізитам.

5. Розроблено алгоритм пошуку інформації в базах даних за реквізитами особистості, в основі якого використовується значення релевантності, отримане в результаті реалізації запропонованої моделі релевантності подібності рядків .

При обробці інформації в БД отримані результати, що свідчать про загальну позитивну тенденцію зниження рівня зашумленості даних. Так, зниження показника зашумленості даних вже в перший місяць склав - 2,47%. Середній показник зниження зашумленості за рік склав 5,3%. Розроблені алгоритми під час виконання дипломної роботи, можуть бути використання в якості типових рішень при проектуванні та застосуванні подібних систем для підприємств середнього та малого бізнесу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Бирюков А.А. Информационная безопасность защита и нападение. А.А. Бирюков – М.: ДМК Пресс, 2012. – 474 с.

2. Бхуптани, М. Ш. ID-технологии на службе вашего бизнеса. / М. Ш. Бхуптани. – М.: Альпина Паблишер, 2007. – 290 с.

3. Гнеденко, Б. В. Введение в теорию массового обслуживания / Б. В. Гнеденко, И. Н. Коваленко. – 6-е изд. – М.: УРСС: Изд-во ЛКИ, 2013. – 352 с.

4. Гладков, Л.А., Генетические алгоритмы: учеб. пособие для вузов. / Л.А. Гладков, В.В. Курейчик– М.: Физматлит, 2006. —290с.

5. Гольдштейн, Б.С. Сети связи пост-NGN /Б.С. Гольдштейн, А.Е. Кучерявый. –СПб.:БХВ-Петербург, 2014. –160с.: ил.

6. Джхунян, В.Л. Электронная идентификация. Бесконтактные идентификаторы и смарт карты / В.Л. Джхунян, В.Ф. Шаньгин. – М.: «Издательство АСТ»: Издательство «НТ Пресс», 2004. – 470с.

7. Кузьменко Н.Г. Компьютерные сети и сетевые технологии/ Н.Г. Кузьменко– М: Наука и техника, 2013. – 368 с.

8. Кутузов, О. И. Инфокоммуникационные сети. Моделирование и оценка вероятностно-временных характеристик [Текст] : монография / О. И. Кутузов, Т. М. Татарникова - СПб. : ГУАП, 2015. – 381 с.

9. Лекции по теории графов: учебное пособие для студентов / В.А. Емеличев, О.И. Мельников, В.И. Сарванов, Р.И. Тышкевич.–3-е изд.–М.: УРСС: Либроком, 2013. – 382 с.

10. Мартин, Роберт Быстрая разработка программ. Принципы, примеры, практика. /Роберт Мартин, Джеймс Ньюкирк — Изд-во: Диалектика-Вильямс, 2004. — 752 с.

11. Мартин, Роберт С. Гибкая разработка программ на Java и C++. Принципы, паттерны и методики. /Роберт С. Мартин, Джеймс Ньюкирк— Изд-во: Диалектика-Вильямс, 2016. — 704 с.

12. Матвеев М. Д. Администрирование Windows 10 / М. Д. Матвеев, Р. Г. Пронди и др. – Санкт-Петербург: Наука и техника, 2013. – 396 с.

13. Мишин Н. Школа сисадмина. Курс лекций "Сети" Н.Мишин - М.: Вильямс, 2015. — 728 с.

14. Мозолюк В.О. Дослідження проблем ідентифікації об'єктів в базах даних / В.С. Орленко, В.М. Лоза, С.В. Мостовий, В.О. Мозолюк / Тези доповідей XVII міжнародної наукової конференції студентів, аспірантів та молодих учених. / ред. кол. Д. Струнін (голова ВНТ ВІКНУ) – К., - 2020. –С.55

15. Мозолюк В.О. Проблеми ідентифікації об'єктів в базах даних / В.О. Мозолюк, В.М. Джулій // Всеукраїнська науково – практична конференція молодих науковців і студентів «Інтелектуальний потенціал 2020», 9-10.11.2020, - ХНУ.- Частина 2. Комп‘ютерні системи та кібербезпека – 60с..

16. Олифер, В. Г. Компьютерные сети. Принципы, технологии, протоколы /В. Г. Олифер, Н. А.Олифер - СПб.: Питер, 2017. - 992 с.

17. Партыка Т. Л. Информационная безопасность учебное пособие / Т. Л. Партыка, И. И. Попов. – М.: ФОРУМ, 2011. – 432 с.

18. Половко А. М. Основы теории надежности: учебное пособие для студентов вузов/А. М. Половко, С. В. Гуров.–Санкт-Петербург: БХВ-Петербург, 2006. – 702 с.

19. Проскурин В. Г. Защита программ и данных: учебное пособие / В. Г. Проскурин, С. В. Крутов, И. В. Мацкевич. – М.: Академия, 2017. – 198 с.

20. Романец Ю. В. Защита информации в компьютерных системах и сетях. / Ю. В. Романец, П. А. Тимофеев, В. Ф. Шаньгин. – М.: Радио и связь, 20011.–366 с.

21. Салмре, Иво Конечный автомат для пользовательского интерфейса. Программирование мобильных устройств на платформе .NET Compact Framework. / Иво Салмре – Изд-во: Издательский дом "Вильямс", 2016. – 736с.

22. Сердюк В. А. Организация и технологии защиты информации / В. А. Сердюк. – М.: Издательский дом Государственного университета – Высшей школы экономики, 2011. – 571 с.

23. Соболь Б.В., Сети и телекоммуникации. /Б. В. Соболь, Г.А. Манин, Е.Д. Герасименко - Учебное пособие. - М.: Феникс, 2015. — 191 с.

24. Суворов А.Б. Основы технологий массовых телекоммуникаций. / А.Б. Суворов— М.: Феникс, 2014. — 509 с.

25. Таранцев А. А. Инженерные методы теории массового обслуживания / А. А. Таранцев. – Санкт-Петербург: Наука, 2007. – 164 с.

26. Тарасюк М. В. Защищенные информационные технологии / М. В. Тарасюк. – М.: Солон-пресс, 2004. – 191 с.

27. Татт У. Теория графов / У. Татт, пер. с англ. Г. П. Гаврилова. – М. Мир, 2008. – 424 с.

28. Теоретические основы компьютерной безопасности / П.Н.Девянин, О.О. Михальский, Д. И. Правиков, А. Ю. Щербаков. –М.: Радио и связь, 2000.–192 с.

29. Тихоненко О. М. Модели массового обслуживания в информационных системах: учебное пособие для ВУЗов / О. М. Тихоненко. – Минск: Технопринт, 2003. – 327 с.

30. Трубачев А. П. Оценка безопасности информационных технологий / А. П. Трубачев и др. под. общ. ред. Галатенко В. А. – М.: СИП РИА, 2001. – 356 с.

31. Тюрликов, А. М. Методы случайного множественного доступа [Текст] : монография / А. М. Тюрликов - Санкт-Петербург : ГУАП, 2014. - 299 с. : ил.

32. Харари Ф. Теория графов / Фрэнк Херари; пер. с англ. И предисл. В.П, Козырева; под. ред. Г.П. Гаврилова. – 3-е изд., стер. – М.: УРСС, 2016. – 290 с.

33. Чекмарев А. Н. Microsoft Windows 10: справочник администратора. / А. Чекмарев. – Санкт-Петербург: БХВ–Петербург, 2015. – 857 с.

ДОДАТОК А
(обовязковий)
Код (лістинг) програмного забезпечення
системи оптимізації інформації при запису в бази даних

```
$ sudo /usr/local/bin/indexer --config /usr/local/etc/sphinx.conf --
all
Sphinx 0.9.7
Copyright (c) 2001-2007, Andrew Aksyonoff

using config file '/usr/local/etc/sphinx.conf'...
indexing index 'catalog'...
collected 8 docs, 0.0 MB
sorted 0.0 Mhits, 82.8% done
total 8 docs, 149 bytes
total 0.010 sec, 14900.00 bytes/sec, 800.00 docs/sec
$ /usr/local/bin/search --config /usr/local/etc/sphinx.conf ENG
Sphinx 0.9.7
Copyright (c) 2001-2007, Andrew Aksyonoff

index 'catalog': query 'ENG ': returned 2 matches of 2 total in 0.000
sec

displaying matches:
1. document=8, weight=1, assembly=5, model=7
        id=8
        partno=ENG088
        description=Cylinder head
        price=55
2. document=9, weight=1, assembly=5, model=3
        id=9
        partno=ENG976
        description=Large cylinder head
        price=65

words:
1. 'eng': 2 documents, 2 hits

$ /usr/local/bin/search --config /usr/local/etc/sphinx.conf wind
Sphinx 0.9.7
Copyright (c) 2001-2007, Andrew Aksyonoff

index 'catalog': query 'wind ': returned 2 matches of 2 total in
0.000 sec

displaying matches:
1. document=1, weight=1, assembly=3, model=1
        id=1
        partno=WIN408
        description=Portal window
```

```
        price=423
2. document=5, weight=1, assembly=3, model=1
        id=5
        partno=WIN958
        description=Windshield, front
        price=500

words:
1. 'wind': 2 documents, 2 hits

$ /usr/local/bin/search \
--config /usr/local/etc/sphinx.conf --filter  model 3 ENG
Sphinx 0.9.7
Copyright (c) 2001-2007, Andrew Aksyonoff

index 'catalog': query 'ENG ': returned 1 matches of 1 total in 0.000
sec

displaying matches:
1. document=9, weight=1, assembly=5, model=3
        id=9
        partno=ENG976
        description=Large cylinder head
        price=65
<?php
  include('sphinx-0.9.7/api/sphinxapi.php');

  $cl = new SphinxClient();
  $cl->SetServer( "localhost", 3312 );
  $cl->SetMatchMode( SPH_MATCH_ANY  );
  $cl->SetFilter( 'model', array( 3 ) );

  $result = $cl->Query( 'cylinder', 'catalog' );

  if ( $result === false ) {
      echo "Query failed: " . $cl->GetLastError() . ".\n";
  }
  else {
      if ( $cl->GetLastWarning() ) {
          echo "WARNING: " . $cl->GetLastWarning() . "<br>";
      }

      if ( ! empty($result["matches"]) ) {
          foreach ( $result["matches"] as $doc => $docinfo ) {
                echo "$doc\n";
          }

          print_r( $result );
      }
  }
```

```
  exit;
?>

$ sudo mkdir -p /var/log/searchd
$ sudo /usr/local/bin/searchd --config /usr/local/etc/sphinx.conf
$ php search.php
9
Array
(
    [fields] => Array
        (
            [0] => partno
            [1] => description
        )

    [attrs] => Array
        (
            [assembly] => 1
            [model] => 1
        )

    [matches] => Array
        (
            [9] => Array
                (
                    [weight] => 1
                    [attrs] => Array
                        (
                            [assembly] => 5
                            [model] => 3
                        )

                )

        )

    [total] => 1
    [total_found] => 1
    [time] => 0.000
    [words] => Array
        (
            [cylind] => Array
                (
                    [docs] => 2
                    [hits] => 2
                )

        )
)
```

ДОДАТОК Б

(обовязковий)

Перелік наукових праць

2. Doctrine OPvM [Електронний ресурс]. – Режим доступу: URL: http://doctrine-project.org (дата звернення: 04.08.2017).

3. Роберт Мартин Гибкая разработка программ на Java и C++. Принципы, паттерны и методики. /Роберт С. Мартин, Джеймс Ньюкирк, Роберт Косс - Изд-во: Диалектика-Вильямс, 2016. - 704с.

4. Джулій В.М. Методи та алгоритми розробки web-додатків / В.М. Джулій, Ю.О. Гунченко, Д.В. Чешун // Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка. – К.: ВІКНУ, 2017. – Вип. № 56. – С.107-115

Дослідження проблем ідентифікації об'єктів в базах даних

Мозолюк В.О., Джулій В.М.
Хмельницький національний університет

На даний момент СУБД широко використовуються в організації сучасних інструментальних, промислових, аналітичних та інформаційних систем. Однак такий бурхливий розвиток інформаційних технологій баз даних поставило також ряд нових проблем і визначило напрямки подальших досліджень у цій області. Не припиняюча робота дослідників та аналітиків відноситься до питань оптимізації виконання запитів і структур зберігання даних, новітніх способів виконання реляційних операцій, організації пошуку, і багато інших моментів, що визначають результативність роботи СУБД. Програмне забезпечення на даний момент розвивається в умовах швидкого зростання обчислювальних потужностей, апаратних можливостей, швидкості доступу до пам'яті, обсягу пам'яті, пропускної здатності та надійності каналів передачі даних. Все більшого значення набувають засоби, що забезпечують взаємодію в розподіленої системі функціонування інформаційних систем.

Розглянемо більш докладно основні напрямки розвитку сучасних баз даних і СУБД:

1. Стандартизація мови SQL. У сучасних СУБД на даний момент основною мовою написання запитів і доступу до баз даних є мова SQL (Structured Query Language). Міжнародний стандарт даної мови розроблений в 1989 році, і більшість виробників СУБД призвели свої системи у відповідність даному стандарту. Потрібна постійна актуалізація мови SQL до мінливих вимог сучасних програмних продуктів та апаратних засобів.

2. Використання мультипроцесорних організацій. Промислові комерційні СУБД реалізуються на основі архітектури "клієнт-сервер". При даній організації всі операції над базами даних виконуються на сервері, що володіє достатньою продуктивністю і набором обчислювальних ресурсів. Після появи мультипроцесорних симетричних апаратних архітектур в

багатьох СУБД була переглянута організація серверних платформ і реалізована можливість розпаралелювання обчислень.

3. Інтеграція та інтероперабельність. Залишається актуальним рішення проблеми використання баз даних попередніх поколінь і версій. Прагнення до спрощення технологічних процесів і необхідність інтеграції інформаційних ресурсів призвели до розробки СУБД здатних підтримувати поряд зі структурованими даними також і текстові документи і виконувати їх пошук по запитах користувачів. Розвинені засоби текстового пошуку присутні в даний час в DB2 (IBM), Oracle, Microsoft SQL Server та ін.

Ефективність управління сучасним бізнесом заснована на можливості отримання управлінським персоналом всебічної інформації з усіх напрямків діяльності. При цьому важливо встановлення контролю над зростаючими потоками інформації, прискорення процесу їх обробки, пошуку та аналізу даних. Існуючі в даний час і розроблювальні нові автоматизовані системи характеризуються великою різноманітністю підтримуваних інформаційних ресурсів, способів організації даних, функціональними можливостями користувацьких інтерфейсів та інших їх технологічних характеристик. В розробках інформаційних систем даної категорії затребуваний практично весь спектр ключових технологій управління інформацією. В даний час багато організацій накопичили значні обсяги інформації. Внаслідок цього стає актуальною проблема розробки корпоративної системи управління знаннями. Однак серйозною перешкодою на даному шляху є розрізненість інформації в корпоративній системі, нерідко дані по одному напрямку діяльності зберігаються в різних додатках і форматах. Крім того, обсяги оброблюваної електронної інформації наростають по експоненті – цьому сприяє активне впровадження мультимедіа, широке поширення корпоративних і глобальних мереж, відхід більшості компаній від паперового документообігу та перехід на автоматизовані системи управління. В подібній ситуації значно зросла необхідність у створенні та впровадженні ефективних систем пошуку та аналізу даних. Традиційними є системи пошуку, що розвиваються в тісному взаємозв'язку з СУБД і в основному орієнтовані на роботу зі структурованими текстовими даними. Однак інтегровані в СУБД системи пошуку слабо адаптовані для обробки мультимедійної і неструктурованої інформації. За статистикою, частка структурованих даних в сучасних базах даних становить не більше 35-50%, решта ж припадають на частку різних довідників, сканованих документів і іншої розрізненої інформації. У цьому випадку виникає проблема пошуку і вибірки необхідної інформації з великого неструктурованого масиву.

Для багатьох організацій інформація є основним активом. Спотворення або пошкодження важливої інформації може призвести до суттєвих фінансових втрат і репутаційним ризикам. Аналізуючи дані, отримані з відкритих джерел і наукових публікацій, можна виділити основні

види втрат, що виникають внаслідок помилок і спотворень інформації в базах даних: втрати внаслідок невірного, поганого надання послуг («брак» в інформації). Даний вид втрат присутній майже в будь-якій організації. В середньому організація втрачає 25-40% часу співробітників, від втрат даного виду; втрати оплачуваного часу співробітників на непродуктивну діяльність. В тому чи іншому виді даний вид втрат зустрічається в будь якій організації, може досягати, наприклад, у менеджерів середньої ланки більше 50% робочого часу, у менеджерів низової категорії до 80%; втрати внаслідок використання «не оптимальних технологічних ланцюжків. Даний вид втрат присутній майже в будь-якій організації. За цими причинами в середньому організація втрачає близько 35% робочого часу задіяних співробітників і це може призвести до подорожчання однієї операції до 100%; втрати часу, грошових коштів, клієнтів по причині відсутності або дублюванні інформації. Даний вид втрат присутній майже в будь-якій організації. Втрати становлять близько 15% часу співробітників, що спричиняє збільшення вартості виконуваної операції.

Основним чинником, що стимулює розвиток технологій пошуку, є поява великої кількості електронних бібліотек і архівів, що містять значні обсяги актуальних знань. Продуктивність і ефективність будь-якої системи зберігання інформації безпосередньо залежить від ефективності та продуктивності пошукових систем. Саме пошукова система визначає, чи перетворяться в знання численні розрізнені дані, що надходять по різних каналах зв'язку і накопичуються в різноманітних базах даних та електронних архівах. Найбільш поширеним видом інформаційних ресурсів для організацій, що працюють з персональними даними (бюро кредитних історій, банки, страхові організації, будь-які організації з досить крупним штатом співробітників) є тексти на природних мовах. Цим обумовлено широке застосування в таких системах технологій текстового пошуку. Дані технології використовуються при цьому не тільки в системах, побудованих за принципом традиційних текстових систем, але і для пошуку в колекціях, організованих у вигляді веб-сайтів, а також для пошуку в глобальної мережі Інтернет.

При організації пошуку в базах персональних даних клієнтів виникають характерні проблеми, пов'язані з наявністю в запитах орфографічних і фонетичних помилок, помилок введення інформації, а також відсутністю єдиних стандартів транскрипції з іноземних мов. Внаслідок цього задача пошуку в базах персональних даних не може бути повною мірою вирішена тільки методами перевірки на точну відповідність. Стає актуальною задача розробки спеціальних методів і технологій текстового пошуку з використанням нетривіальних рішень, в тому числі на основі операцій несуворої відповідності. Однак універсальної методики пошуку в умовах зашумленості даних не існує, оскільки кожна проблема має власну

оригінальну специфіку. Для рішення виниклих проблем потрібно використовувати алгоритми здатні відшукати всі лексикографічно близькі до шаблону пошуку слова, що відрізняються замінами, пропусками і вставками символів. Таким чином, автоматично стає допустимою помилка, як у вхідних даних, так і в термінах запиту. В даний час можливості виконання пошуку за подібністю не використовуються в СУБД. Таким чином, виникає задача розробки алгоритмів виконання спеціальних реляційних операцій, що виникають в задачі ототожнення записів. Проведений аналіз напрямків розвитку сучасних баз даних показує, що склалися і формуються за останні роки тенденції розвитку інформаційних технологій істотно впливають, у тому числі і на функціональні можливості автоматизованих систем. Задача встановлення відповідності між окремими об'єктами - побудова процедур ототожнення в даний час не має задовільного рішення. Існуючі роботи, присвячені інтеграції БД, дозволяють здійснити тільки інтеграцію схем БД, але не пропонують способів побудови процедур ототожнення. Побудова процедур ототожнення ускладнюється відсутністю серед загальних атрибутів відповідних один одному таблиць різних БД первинних ключів і наявністю помилок операторського введення. Існуючі СУБД не пропонують можливості для використання пошуку за подібністю, що усуває викликані помилки операторського введення.

З урахуванням специфіки роботи з персональними даними пропонується вирішення наступних прикладних задач: повна ідентифікація клієнта при наявності спотворень інформації в базі даних або в пошукових запитах; усунення дублікатів записів при надходженні до БД з множинних джерел зі слабоструктурованою інформацією; пошук і коректування помилок в персональних даних клієнтів (фізичних і юридичних осіб).

В області технологій можна виділити появу принципово нових програмних засобів аналізу фрагментарної, слабоструктурованої, пошкодженої, нечіткої, неповної інформації. До таких засобів можна віднести технологію інформаційного моніторингу комплексних процесів і засоби Business Intelligence (бізнес аналітики). Використання принципово нового інструментарію на основі алгоритмів нечіткого пошуку в міжнародних компаніях і державних організаціях, великих корпораціях і фінансових установах показало їх ефективність і величезний потенціал для вирішення прикладних задач.

Перелік посилань

1. Васильєв В.І. Інтелектуальні системи захисту інформації: навч. посібник / В. І. Васильєв. - 2-е изд., Испр. - М .: Машинобудування, 2012. - 171 с.

2. Гордейчик С.В. Безпека бездротових мереж. / С.В. Гордейчик, В.В. Дубровін - М .: Гаряча лінія - Телеком, 2008. - 288 с.

к.т.н. Лоза В.М. (ВІКНУ)
к.т.н., доц. Орленко В.С. (ХмНУ)
Мостовий С.В. (ХмНУ)
Мозолюк В.О. (ХмНУ)

Проблеми ідентифікації об'єктів в базах даних

Основним чинником, що стимулює розвиток технологій пошуку, є поява великої кількості електронних бібліотек і архівів, що містять значні обсяги актуальних знань. Продуктивність і ефективність будь-якої системи зберігання інформації безпосередньо залежить від ефективності та продуктивності пошукових систем.

При організації пошуку в базах персональних даних клієнтів виникають характерні проблеми, пов'язані з наявністю в запитах орфографічних і фонетичних помилок, помилок введення інформації, а також відсутністю єдиних стандартів транскрипції з іноземних мов. Внаслідок цього задача пошуку в базах персональних даних не може бути повною мірою вирішена тільки методами перевірки на точну відповідність. Стає актуальною задача розробки спеціальних методів і технологій текстового пошуку з використанням нетривіальних рішень, в тому числі на основі операцій несуворої відповідності. Однак універсальної методики пошуку в умовах зашумленості даних не існує, оскільки кожна проблема має власну оригінальну специфіку.

В даний час можливості виконання пошуку за подібністю не використовуються в СУБД. Таким чином, виникає задача розробки алгоритмів виконання спеціальних реляційних операцій, що виникають в задачі ототожнення записів. Проведений аналіз напрямків розвитку сучасних баз даних показує, що склалися і формуються за останні роки тенденції розвитку інформаційних технологій істотно впливають, у тому числі і на функціональні можливості автоматизованих систем. Задача встановлення відповідності між окремими об'єктами - побудова процедур ототожнення в даний час не має задовільного рішення. Побудова процедур ототожнення ускладнюється відсутністю серед загальних атрибутів відповідних один одному таблиць різних БД первинних ключів і наявністю помилок операторського введення. З урахуванням специфіки роботи з персональними даними пропонується вирішення наступних прикладних задач: повна ідентифікація клієнта при наявності спотворень інформації в базі даних або в пошукових запитах; усунення дублікатів записів при надходженні до БД з множинних джерел зі слабоструктурованою інформацією; пошук і коректування помилок в персональних даних клієнтів (фізичних і юридичних осіб).

Список використаних джерел:

1. Гагарина Л. Г. Алгоритмы и структуры данных. / Л. Г. Гагарина, В.Д. Колдаев //— М.:Инфра-М, 2009. - 304 с.

ДОДАТОК В

Презентація

Тема Метод ідентифікації особистості на основі операцій несуворої відповідності та вагових коефіцієнтів

Метою магістерської роботи є оптимізації запису інформації в бази даних, на основі використання значення релевантності подібності рядків, в результаті реалізації запропонованої моделі релевантності подібності рядків. Підвищення ефективності дозволить значно розширити область використання прикладного забезпечення у системах управління і переробки інформації.

Наукова задача - оцінка рівня підвищення інформаційного забезпечення підрозділів підприємства за рахунок зниження зашумленості даних загального інформаційного простору

Об'єкт дослідження: технологія процесу пошуку даних в базах даних інформаційно – пошуковими системами.

Предмет дослідження: є застосування моделей, методів підвищення рівня якісного та ефективного інформаційного забезпечення підрозділів підприємства за рахунок проведення оптимізації інформації в базах даних.

Завдачі досліджень у роботі формулюються наступним чином:

1. Провести дослідження існуючих моделей, методів, а також алгоритмів використовуваних в інформаційно-пошукових системах.
2. Розробити модель представлення релевантності подібності рядків та модель наближеного пошуку інформації при опрацюванні пошукових рядків.
3. Розробити алгоритм оптимізації запису інформації в бази даних та алгоритм пошуку інформації в базах даних за реквізитами особистості.
4. Провести дослідження ефективності запропонованих алгоритмів.

Наукова новизна роботи полягає в:

1. Модель представлення релевантності подібності рядків, дозволяє надати, для рядків пошуку, кількісну оцінку їх подібності.

2. Удосконалений метод оптимізації запису інформації в бази даних, забезпечує розпізнавання та виключення дублювання даних при використанні інформаційно – пошукових систем, на основі автоматичного вибору схеми ручної або автоматичної ідентифікації, що дозволяє зберегти інформаційну цілісність, а також знизити зашумленість даних, зумовлену наявністю помилок операторського введення.

Методика проведення досліджень. Для вирішення задач поставлених в дипломній роботі використовуються методи системного аналізу, імовірнісних графів, випадкових процесів і математичної статистики, теорії ймовірності, теорії прийняття рішень, методів комп'ютерного аналізу, математичного моделювання, методів модульного і структурного програмування.

Практична цінність Запропоновані алгоритми під час виконання дипломної роботи, можуть бути використання в якості типових рішень при проектуванні та застосуванні подібних систем для підприємств середнього та малого бізнесу. Підвищити рівень інформаційного забезпечення підрозділів підприємства за рахунок зниження зашумленості даних загального інформаційного простору.

Апробація роботи. Наукові результати і основні положення магістерської роботи доповідались і обговорювались на всеукраїнських та міжнародних науково-технічних конференціях,

Публікації. По темі дипломної роботи опубліковано 1 - теза доповідей на всеукраїнських конференціях та 1 стаття.

Порівняння алгоритмів пошуку за ефективністю та швидкістю

Алгоритми	Ефективність	Швидкість	Розмір дискового простору	Підсумок
розширеної вибірки	висока	висока тільки при розмірі словника до500 тис. записів	низький	не підходить, в зв'язку з низькою швидкістю на словниках від 5 млн. записів
n-грамм	висока	середня, лінійно залежить від довжини рядків можливо збільшити з допомогою хешування і індексування	високий	підходить
дерева пошуку	низька	висока	середній	підходить як спосіб індексування
відстань між рядками	низька	вище середнього	низький	підходить як спосіб сортування, ранжування результатів іншого алгоритму
хешування по сигнатурі	нижче середнього	вище середнього	середній	підходить
послідовний перебір	висока тільки при пошуку з малою кількістю помилок по великому масиву тексту або при порівнянні на повну відповідність	вище середнього	низький	підходить

Модель представлення релевантності подібності рядків

(Перший науковий результат)

1. Формуємо набори всіх можливих підрядків довжиною до N:

$$G_j(i)=\left\{ g_{j1}(i),\dots,g_{jk}(i),\dots,g_{jn}(i)\right\};\qquad j=1,2;\quad i=\overline{1,N};\quad n=l_j-i+1;$$

де *i*- довжина підрядка; *j* - номер вхідного рядка; *n* - кількість підрядків довжиною *i* в *j*-му слові.

2. Кожному набору $G_j(i)$ поставимо у відповідність множину $G_j^*(i)$, елементи яких не повторюються із набору $G_j(i)$, тобто повторюваним елементам набору $G_j(i)$ в множині $G_j^*(i)$ буде відповідати один елемент:

$$G_j^*(i)=\left\{ g_{j1}^*(i),\dots,g_{jk}^*(i),\dots,g_{jm}^*(i)\right\};\qquad j=1,2;\quad i=\overline{1,N};\quad m\le l_j-i+1;$$

де *m* - кількість неповторюваних підрядків довжиною *i* в *j*-му слові.

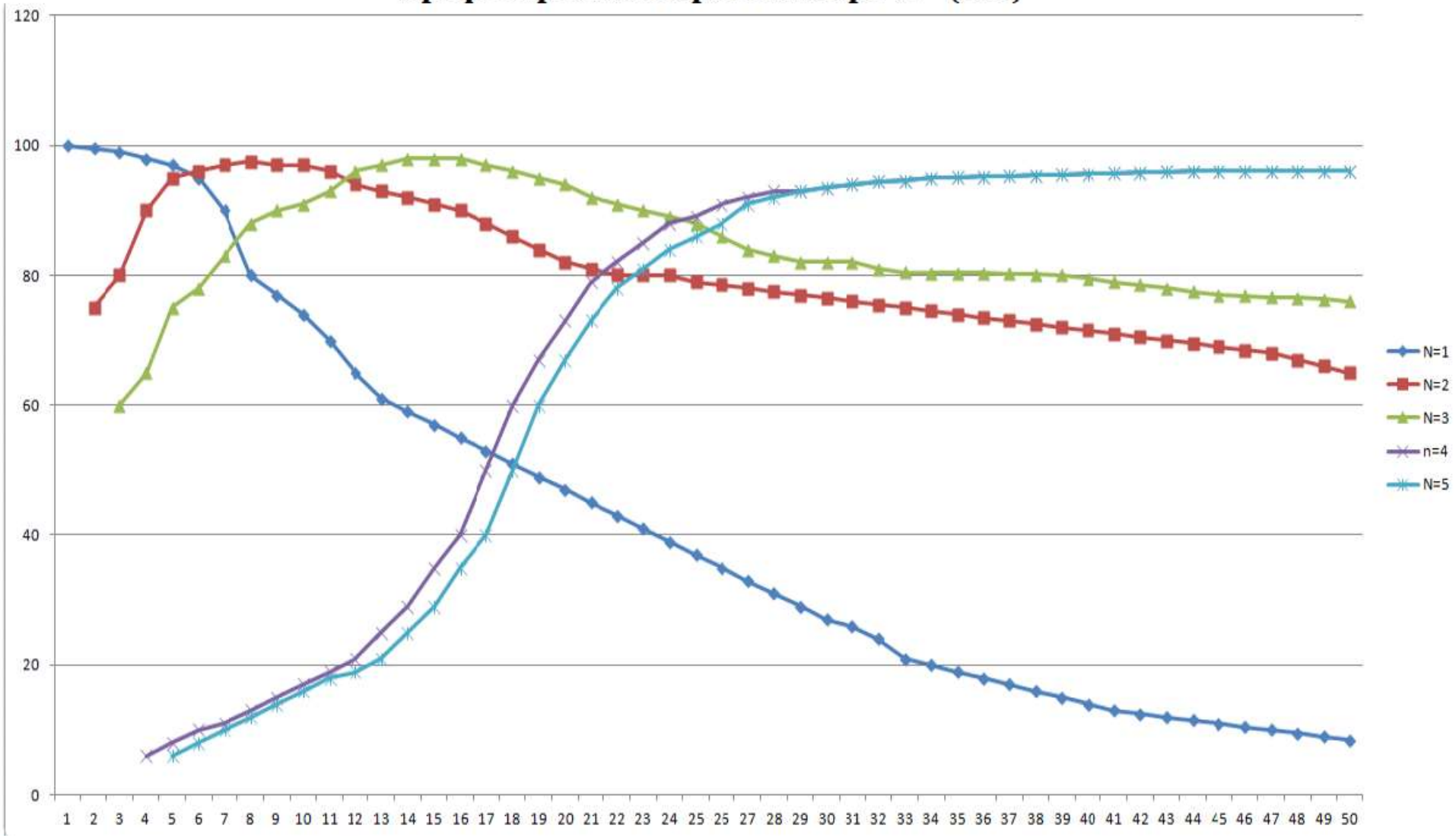
3. Значення функції релевантності $FR=(l_1,l_2,N)$ обчислюється за наступною формулою:

$$FR=(l_1,l_2,N)=\frac{\sum_{i=1}^{N} fr(i)}{N},\qquad fr(i)=\frac{\left|{}^*G_1(i)\right|+\left|{}^*G_2(i)\right|}{\left|G_1(i)\right|+\left|G_2(i)\right|}\qquad {}^*G_j(i)=G_j(i)\cap G_j^*(i),$$

де $g_j(i)\in {}^*G_j(i)\Rightarrow \exists\, g_k^*(i): g_j(i)=g_k^*(i)$, тобто набір ${}^*G_j(i)$ складається з елементів набору $G_j(i)$, для яких є рівні у множині $G_j^*(i)$. $G_j(i)$ - набір підрядків довжиною *i* рядка l_j; $\left|G_j(i)\right|$ - кількість елементів у наборі підрядків $G_j(i)$; $G_j^*(i)$ - множина, в якій не повторюються підрядки набору $G_j(i)$; $\left|G_j^*(i)\right|$ - кількість елементів у наборі підрядків $G_j^*(i)$; $\left|{}^*G_j(i)\right|$ - кількість елементів у наборі підрядків ${}^*G_j(i)$; N - максимальна довжина підрядка.

Дозволяє надати, для рядків пошуку, кількісну оцінку їх подібності , відрізняється від існуючих застосуванням алгоритму несуворої відповідності

Графік прийняття рішення при N= {1..5}



Удосконалений метод оптимізації запису інформації в бази даних

(Другий науковий результат)

1. Вхідні дані: масив інформації, яку необхідно додати в базу даних, з умовою виключення дублювання даних.

2. Проводиться обчислення значення релевантності кожного рядка вхідного масиву з кожним рядком бази даних.

3. Якщо значення релевантності:

- вище межі автоматичної ідентифікації (Γ_a), після якої кількість розпізнаних дублікатів стає практично рівним 100%, то відповідний вхідний рядок оголошуються дублікатом.

- нижче межі ручної ідентифікації (Γ_p), то рядки для яких обчислюється релевантність оголошуються різними і аналіз триває.

- вище Γ_p, але нижче Γ_a, то такі рядки відправляються в лог прийняття рішень для обробки аналітиком.

4. Якщо у якого-небудь рядка вхідного масиву всі значення релевантності нижче Γ_p, то даний рядок оголошується новим і додається в базу даних.

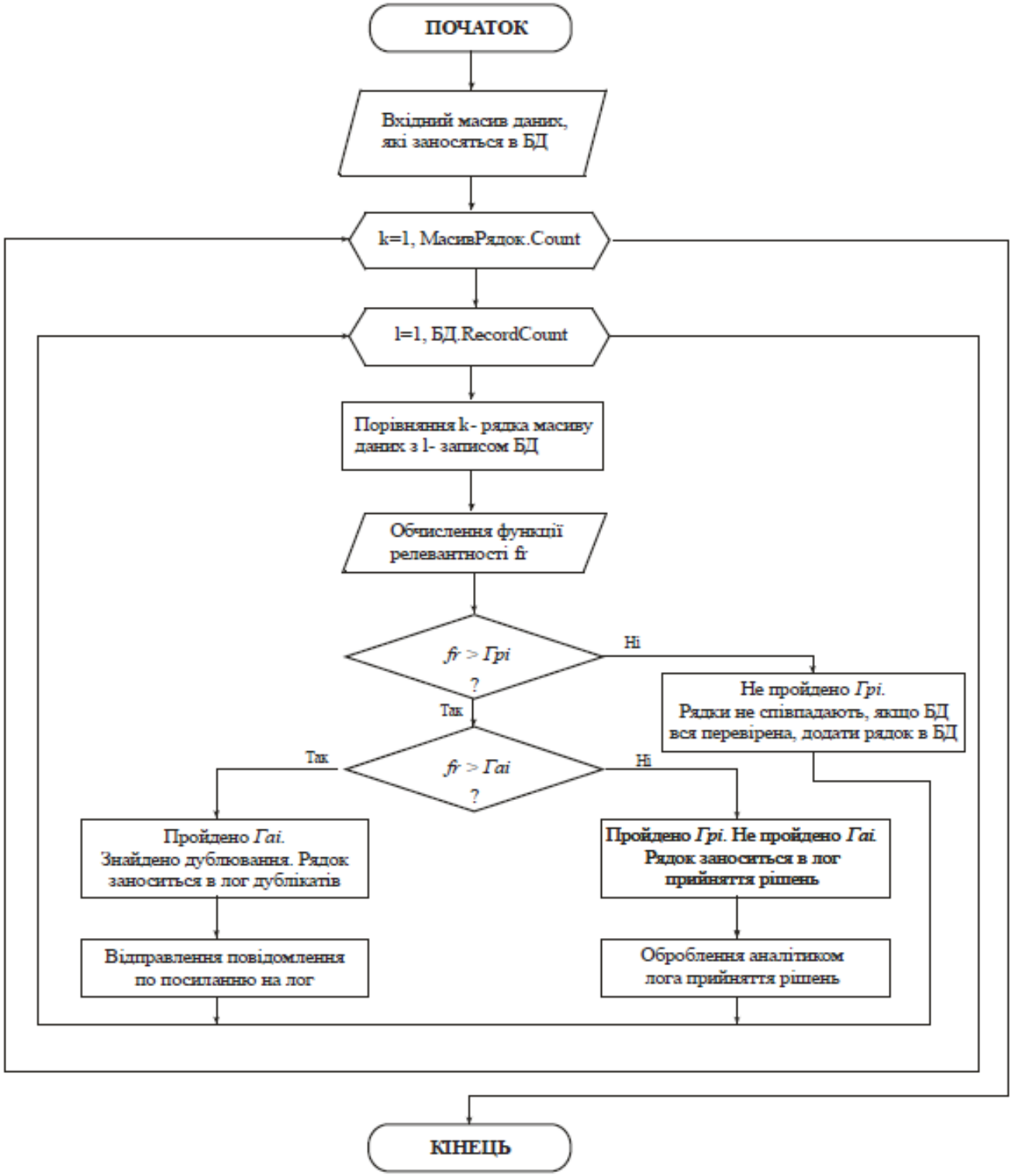
У вирішенні задачі виявлення дублюючої інформації в базі даних можна виділити три етапи:

1. Виявлення дублюючої інформації на рівні введення інформації користувачами та її відхилення;

2. Виявлення дублюючої інформації шляхом порівняння і аналізу уже введених даних відповідно до заданого Γ_a і автоматичне видалення дублюючої інформації;

3. Аналіз та обробка користувачем результатів, які не можуть бути оброблені автоматично (показник відповідності нижче Γ_a але вище Γ_p.

Алгоритм оптимізації запису інформації в бази даних



Метод формування пошукового індексу за реквізитами особистості

Вага - умовний коефіцієнт реквізиту. Залежить від повноти, достовірності, і актуальності реквізиту: «Прізвище» -5; «Ім'я» і «Дата народження» -4; «По батькові» і «Місце народження» -3.

Правило - поєднання реквізитів людини, за якими здійснюється пошук.

№ правила	Реквізити	Сумарна вага реквізитів	Поріг ідентифікації по правилу
Правило 1	Прізвище (5) + Ім'я (4)+ По батькові (3)	12	11
Правило 2	Прізвище (5) + Ім'я (4) + Дата народження (4)	13	12
Правило 3	Ім'я (4) + По батькові (3)+ Дата народження (4)	11	10
Правило 4	Дата народження (4) + По батькові (3) + Місце народження (3)	10	9

$$\sum_{i=1}^{n} p_j(i)\cdot\left(R_i\cdot w_i + L_i\right) \geq k_j; \qquad j=\overline{1,m}; \qquad i=\overline{1,n};$$

де $p_j(i)$ - елемент правила ідентифікації. Правило – поєднання реквізитів фізичної особи, за якими відбувається порівняння, $p_j(i)=1$, якщо i-й реквізит входить в j-е правило, $p_j(i)=0$, якщо не входить.

R_i - результат роботи системи визначення релевантності від i-их реквізитів;

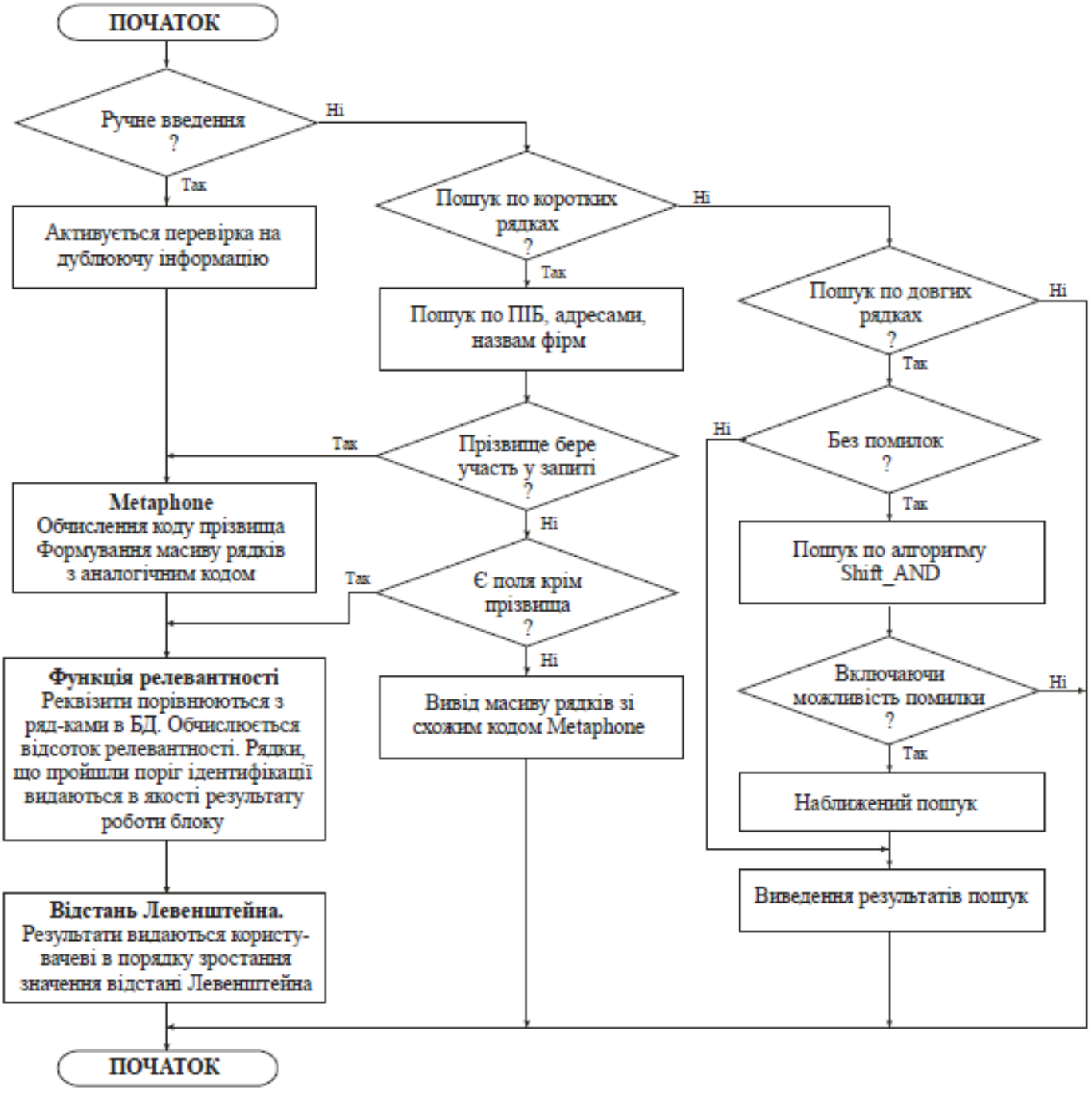
w_i - вага реквізиту. Залежить від повноти, достовірності та актуальності реквізиту. Визначає значимість реквізиту для ідентифікації фізичної особи;

L_i - підвищувальний коефіцієнт розрахований на підставі відстані Левенштейна між i -ми реквізитами;

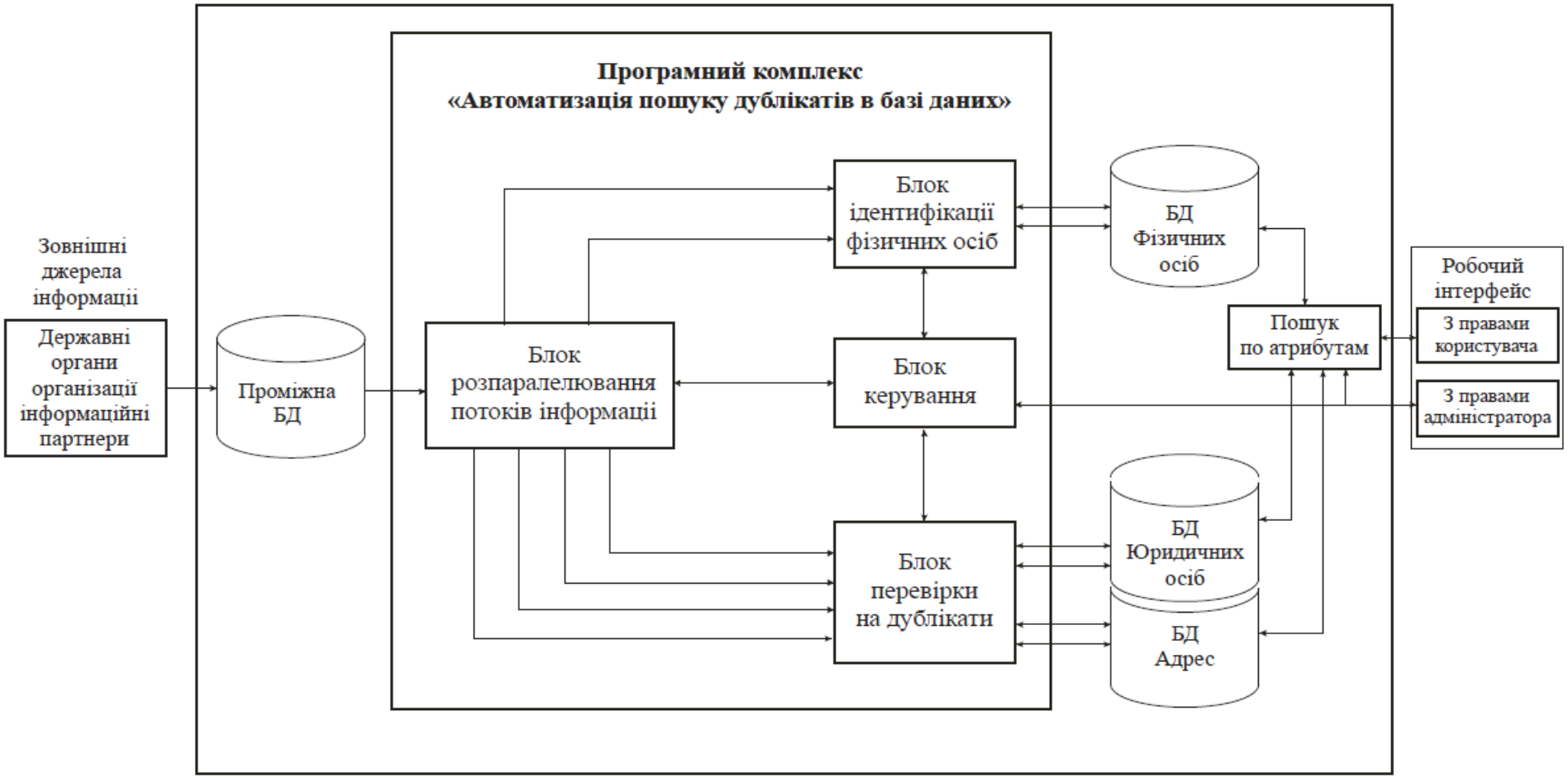
k_j - поріг ідентифікації правила. Необхідний для виключення записів, які не пройшли ідентифікацію за правилами. Якщо поріг пройшли декілька записів, то автоматизовано ідентифікувати громадянина неможливо. Така ситуація відпрацьовується оператором;

m - кількість правил; n - кількість реквізитів, що беруть участь в порівнянні.

Алгоритм пошуку інформації в базах даних за реквізитами особистості



Архітектура системи
«Оптимізація інформації при запису в бази даних»



ВИСНОВКИ

Основні результати, отримані в ході наукового дослідження, полягають у наступному:

1. Запропонована модель представлення релевантності подібності рядків на основі алгоритмів порівняння підрядків пошукового шаблону та рядка пошуку. Отримані результати задовільняють роботу алгоритмів пошуку, як по ефективності, так і за швидкістю отримання результатів пошуку кінцевими користувачами. Особливість отриманих результатів характеризується достатньою стабільністю в точності, як результат надається можливість створення модифікацій алгоритмів наближеного пошуку інформації з метою розширення області виконуваних задач.

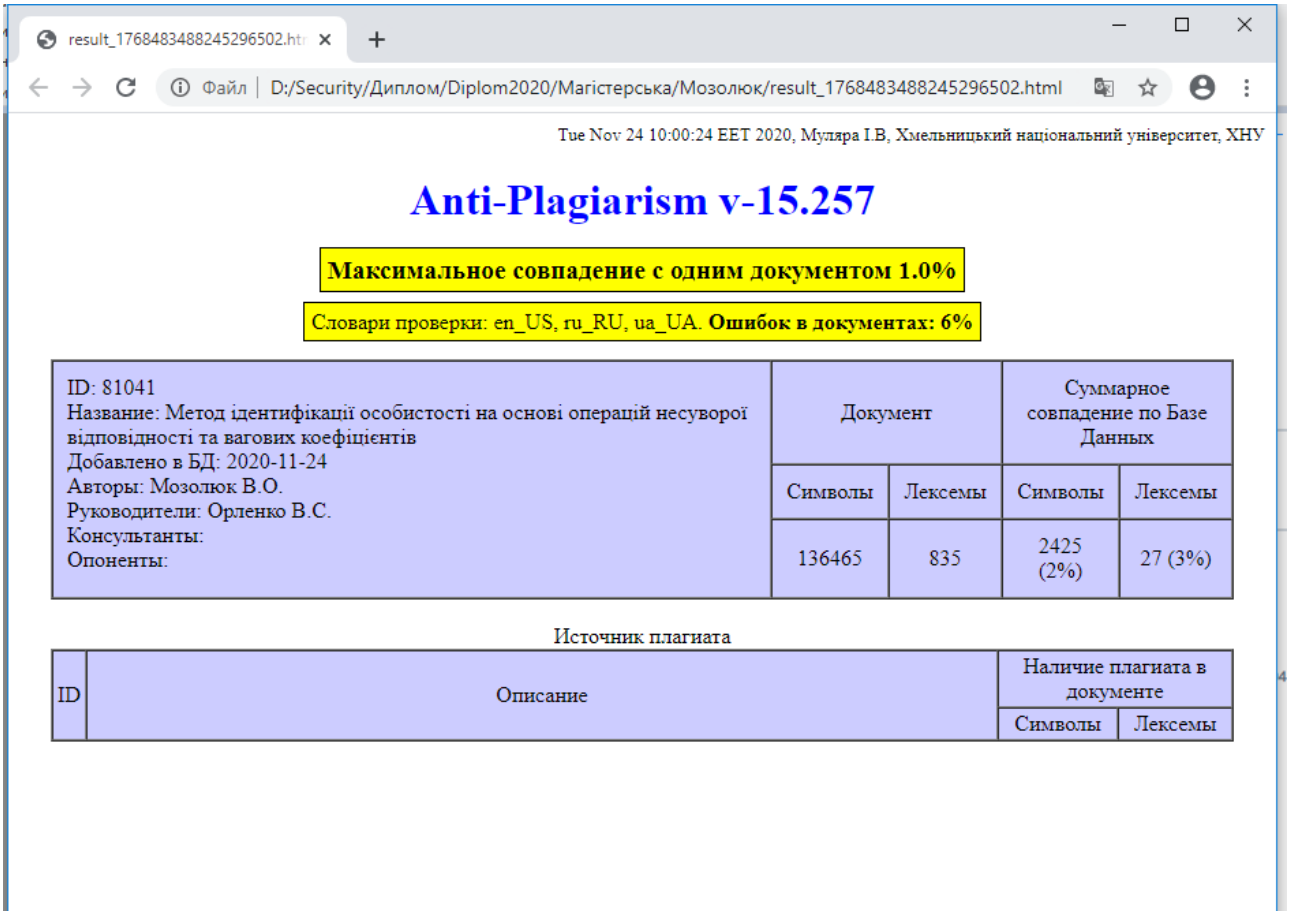
2. Запропонована модель наближеного пошуку при опрацюванні пошукових рядків. Надає можливість в разі необхідності, щоб при роботі алгоритму допускалося в результаті пошуку більше однієї помилки, для вирішення цієї задачі необхідно ввести додаткові матриці на кожну помилку і при цьому необхідно провести аналогічні обчислення для перетворення таблиці в іншу, дозволяє виконувати наближений пошук будь-якого виразу, з врахуванням помилок чи провести точний пошук.

3. Розроблено алгоритм оптимізації запису інформації в бази даних, використовує значення релевантності подібності рядків, які порівнюються, також дозволяє надавати рішення в даній ситуації в найбільш загальному вигляді. Запрпонований алгоритм надає можливість об'єднання рядків, які задовільнають заданому відсотку подібності за вказаним набором реквізитів, вага яких відповідає встановлених межах.

4. Розроблено алгоритм ідентифікації особистості в базах даних з використанням набору правил ідентифікації, дозволяє оцінити ступінь подібності інформації занесеної в базу даних про клієнтів, система правил і ваг є основою для формування правил ідентифікації особистості, встановлення їхнього порогу ідентифікації, в залежності від призначених ваг реквізитам.

5. Розроблено алгоритм пошуку інформації в базах даних за реквізитами особистості, в основі якого використовується значення релевантності, отримане в результаті реалізації запропонованої моделі релевантності подібності рядків .

При обробці інформації в БД отримані результати, що свідчать про загальну позитивну тенденцію зниження рівня зашумленості даних. Так, зниження показника зашумленості даних вже в перший місяць склав - 2,47%. Середній показник зниження зашумленості за рік склав 5,3%. Розроблені алгоритми під час виконання дипломної роботи, можуть бути використання в якості типових рішень при проектуванні та застосуванні подібних систем для підприємств середнього та малого бізнесу.



result_176848348824529650 ×

Файл | D:/Security/Диплом/Diplom2020/Магістерська/Мозолюк/result_176848348824529650 2.html

Tue Nov 24 10:00:24 EET 2020, Муляра І.В, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 1.0%

Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 6%**

ID: 81041 Название: Метод ідентифікації особистості на основі операцій несуворої відповідності та вагових коефіцієнтів Добавлено в БД: 2020-11-24 Авторы: Мозолюк В.О. Руководители: Орленко В.С. Консультанты: Оппоненты:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	136465	835	2425 (2%)	27 (3%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

UNICHECK

User name:
Kafedra kiberbezpeky

Check ID:
1005327599

Check date:
02.12.2020 11:47:22 EET

Check type:
Doc vs Internet

Report date:
02.12.2020 11:48:25 EET

User ID:
100005590

File name: **Мозолюк_ю**

Page count: **93** Word count: **19999** Character count: **151462** File size: **5.23 MB** File ID: **1005450529**

2.92% Matches

Highest match: **0.5%** with Internet source (http://elar.khnu.km.ua/jspui/bitstream/123456789/4165/1/vott_2014_4_27.pdf)

2.92% Internet sources | 321 ..... Page 95

No Library search was conducted

0% Quotes

Exclusion of quotes is off

Exclusion of references is off

0% Exclusions

No exclusions

Modifind

Text modifications detected. Find more details in the online report.

Replaced characters | 58

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ ПО КАФЕДРІ КІБЕРБЕЗПЕКИ ТА КОМП'ЮТЕРНИХ СИСТЕМ І МЕРЕЖ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод ідентифікації особистості на основі операцій несуворої відповідності та вагових коефіцієнтів

Автор: Мозолюк Віталій Олександрович

Спеціальність: 123 Комп'ютерна інженерія

Освітня програма:

Науковий керівник: Орленко Вікторія Сергіївна

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних). Робота приймається до захисту.	✓
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

до захисту допускається

4.12.2020

Дата

Підписи

Клим Ю.М.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА ДИПЛОМНУ РОБОТУ

освітнього ступеня «магістр»

Магістр Мозолюк Віталій Олександрович

Тема Метод ідентифікації особистості на основі операцій несуворої відповідності та вагових коефіцієнтів

Спеціальність 123 «Комп'ютерна інженерія»

спеціалізація «Комп'ютерні системи та мережі»

Обсяг дипломної роботи освітнього ступеня «магістр»:

кількість плакатів 11; кількість сторінок записки 95

1. Короткий зміст ДР та прийнятих рішень В рамках магістерської роботи проведено аналіз моделей, методів і алгоритмів пошуку інформації. Основні результати, отримані в ході наукового дослідження, полягають у наступному: запропонована модель представлення релевантності подібності рядків на основі алгоритмів порівняння підрядків пошукового шаблону та рядка пошуку; запропонована модель наближеного пошуку при опрацюванні пошукових рядків, розроблено алгоритм оптимізації запису інформації в бази даних, використовує значення релевантності подібності рядків; розроблено алгоритм пошуку інформації в базах даних за реквізитами особистості, в основі якого використовується значення релевантності. Отримані результати свідчать про загальну позитивну тенденцію зниження рівня зашумленості даних.

2. Висновок про відповідність ДР дипломному завданню Дипломна робота освітнього ступеня «магістр» у повній мірі відповідає поставленому завданню як в теоретичній, так і в практичній частині дипломної роботи

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У вступі обгрунтовується актуальність теми роботи, дається аналіз досліджуваної проблеми і обгрунтовується застосовуваний підхід до її вирішення, формулюються цілі і завдання дослідження, описується наукова новизна і практична значимість отриманих результатів. У першому розділі якісно та в повній мірі проаналізовано сучасний стан існуючих алгоритмів, методів формування інформаційного забезпечення. Наступні розділи присвячені розробці моделі та методу наближеного пошуку особистості за реквізитами предметної області, що дозволяє структурувати контент інформаційно-довідкових та інших систем.

4. Позитивні сторони проекту Дипломна робота містить ряд інноваційних рішень, що характеризуються науковою новизною: модель представлення релевантності подібності рядків, дозволяє надати, для рядків пошуку, кількісну оцінку їх подібності. Удосконалений метод оптимізації запису інформації в бази даних, забезпечує розпізнавання та виключення дублювання даних при використанні інформаційно – пошукових систем, на основі автоматичного вибору схеми ручної або автоматичної ідентифікації, що дозволяє зберегти інформаційну цілісність, а також знизити зашумленість даних, зумовлену наявністю помилок операторського введення

5. Негативні сторони проекту Метод оптимізації запису інформації в бази даних, забезпечує розпізнавання та виключення дублювання даних при використанні інформаційно – пошукових систем. Перед внесенням інформації в БД йде її перевірка на дублювання. При цьому значно падає швидкість роботи пошукових систем. Яким чином дана ситуація вирішується в дипломній роботі?

6. Оцінка графічного оформлення та пояснювальної записки роботи Графічне оформлення виконане відповідно до теми дипломної роботи з дотриманням стандартів. В загальному графічне оформлення виконане на достатньому рівні. Пояснювальна записка відповідає нормам для її оформлення.

7. Відгук про роботу в цілому В загальному дипломна робота заслуговує позитивної оцінки. Весь матеріал дипломної роботи структурований, чіткий та послідовний. Усі розділи роботи послідовні та логічні, що дозволяє чітко розуміти викладений матеріал в рамках тематики дипломної роботи. Графічний матеріал дозволяє наочно побачити доцільність та ефективність рішень, які були прийняті за основу для досягнення поставленої задачі.

8.Інші зауваження

9. Оцінка дипломної роботи Розглянувши позитивні та негативні сторони представленої дипломної роботи, можна зробити висновок, що вона заслуговує оцінку «добре».

РЕЦЕНЗЕНТ (прізвище, ім'я, по батькові, посада, місце роботи)

Лисенко Сергій Миколайович, доцент, кафедра комп'ютерної інженерії та системного програмування, Хмельницького національного університету

«02» грудня 2020. _____ (підпис)

ХНУ Зам. 54, тир. 6000, 2007