

## АНАЛІЗ СТАНУ ПРОБЛЕМ ІДЕНТИФІКАЦІЇ ОБ'ЄКТІВ У БАЗАХ ДАНИХ

*У статті проведено аналіз стану проблем ідентифікації об'єктів в базах даних.*

*Сучасні алгоритми словникового пошуку ефективніші алгоритмів послідовного перебору, але при роботі з рядками великої довжини послідовний перебір істотно виграє в швидкості. З найбільш ефективних алгоритмів, розглянутих в роботі, слід зазначити алгоритми N-грам, trie-дерева, а також сигнатурні алгоритми, які забезпечують хороше співвідношення між розміром індексу і швидкістю пошуку.*

*Наведена порівняльна характеристика алгоритмів пошуку з точки зору ефективності, швидкості і необхідного дискового простору*

*Ключові слова: ідентифікації об'єктів, бази даних, алгоритм пошуку, релевантність.*

**Вступ.** Неухильне зростання обсягів інформації спричиняє необхідність широкого використання передових інформаційних технологій для ефективного управління потоками даних, при цьому найбільшу значимість набувають завдання створення прогресивних інструментів оцінки та контролю зростаючих потоків інформації, оптимізацій процедур

обробки, агрегації, узагальнення, пошуку та аналізу даних. Зростає попит на створення як корпоративних автоматизованих інформаційних систем (АІС), так і окремих спеціалізованих рішень. Автоматизовані інформаційні системи розробляються на основі інформаційно-аналітичних баз даних, які використовуються в якості ключового елементу системи і забезпечують зберігання і обробку всієї сукупності даних, що надходять від підрозділів і філій. З точки зору технологій АІС представляє набір апаратних засобів, технологій, методів і алгоритмів, спрямованих на підтримку життєвого циклу інформації і включає три основні процеси: обробку даних, управління інформацією та управління знаннями.

Разом з тим, існують фактори, які стримують розвиток АІС. Для зазначених процесів характерні проблеми управління якістю даних, у тому числі пов'язані з наявністю як в запитах, так і безпосередньо в базах даних орфографічних і фонетичних помилок, помилок введення інформації, а також відсутність єдиних стандартів транскрипції з іноземних мов. В даний час універсальної методики їх рішення не існує, оскільки кожна проблема має власну специфіку. Внаслідок цього завдання текстового пошуку в базах даних не може бути повною мірою вирішене тільки методами перевірки на точну відповідність. Постає актуальною задача розробки спеціальних методів і технологій пошуку інформації з використанням нетривіальних рішень, у тому числі з використанням операцій не суворої відповідності.

**Проблеми ідентифікації об'єктів у базах даних.** На даний момент СУБД широко використовуються в організації сучасних інструментальних, промислових, аналітичних та інформаційних систем. Однак такий бурхливий розвиток інформаційних технологій баз даних поставив також ряд нових проблем і визначив напрямки подальших досліджень у цій області. Програмне забезпечення на даний момент розвивається в умовах швидкого зростання обчислювальних потужностей, апаратних можливостей, швидкості доступу до пам'яті, обсягу пам'яті, пропускну здатності та надійності каналів передачі даних. Все більшого значення набувають засоби, які забезпечують взаємодію в розподіленій системі функціонування інформаційних систем.

Ефективність управління сучасним бізнесом заснована на можливості отримання управлінським персоналом всебічної інформації з усіх напрямків діяльності. При цьому важливим є встановлення контролю над зростаючими потоками інформації, прискорення процесу їх обробки, пошук та аналіз даних. Існуючі в даний час і розроблювані нові автоматизовані системи характеризуються великою різноманітністю підтримуваних інформаційних ресурсів, способів організації даних, функціональними можливостями користувальницьких інтерфейсів та іншими технологічними характеристиками. Для розробки інформаційних систем даної категорії необхідний 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 і 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. В 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 підвищення ризику втрати іміджу. 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, неякісного надання послуг («брак» в інформації). Даний вид втрат присутній майже в будь-якій організації. В середньому організація втрачає 25-40% часу співробітників, від даного виду втрат (втрати оплачуваного часу співробітників на непродуктивну діяльність). Даний вид втрат може досягати, наприклад, у менеджерів середньої категорії 50% робочого часу, у менеджерів нижчої категорії до 80%. За цими причинами, в середньому організація втрачає близько 35% робочого часу задіяних співробітників. Це може призвести до подорожчання однієї операції до 100%; втрати часу, коштів, клієнтів по причині відсутності або дублювання інформації. В цілому втрати становлять близько 15% часу співробітників.

Основним чинником, що стимулює розвиток технологій пошуку, є поява великої кількості електронних бібліотек і архівів, які 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, є тексти на природних мовах. Цим обумовлено широке застосування в таких системах технологій текстового пошуку. Дані технології використовуються 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 глобальній 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 із відсутністю єдиних стандартів транскрипції з іноземних мов.

Внаслідок цього, завдання пошуку в базах персональних даних не може бути в повній мірі вирішено тільки методами перевірки на точну відповідність. Постає актуальною задача розробки спеціальних методів і технологій текстового пошуку з використанням нетривіальних рішень, в тому числі на основі операцій несуророї відповідності. Проведений аналіз напрямків розвитку сучасних баз даних показує, що за останні роки склалися і формуються тенденції розвитку інформаційних технологій, які здійснюють істотний вплив на функціональні можливості автоматизованих систем кредитних організацій.

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 та засоби Business Intelligence. 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.

**Обмеження існуючих пошукових алгоритмів.** Відстань між рядками: якщо  $a$  і  $b$  - деякі рядки – послідовності символів алфавіту  $S$  довжин  $m$  і  $n$  відповідно і заданий набір операцій, що перетворюють дані рядки: вставка символу; видалення символу; заміна одного символу на інший. Кожній операції присвоєна вартість. Існує послідовність операцій, результатом застосування якої до рядка  $a$  буде рядок  $b$ . Така послідовність операцій не є єдиною. Вартість послідовності визначається як сумарна вартість операцій, що входять в неї. Тоді послідовність операцій мінімальної вартості визначає метрику на множині рядків.

Відстань Хемінга між двома словами, рівними по довжині, обчислюється по числу позицій, символи в яких не рівні. Це рівнозначно мінімальній кількості операцій заміни (при забороні видалення/вставки) необхідних для перетворення одного рядка в інший. Якщо проводиться порівняння слів, що мають різну довжину, то необхідні також операції

видалення і вставки. Якщо вони мають таку ж вагу, що і заміна, то мінімальна ціна перетворення одного рядка в інший задає метрику - відстань Левенштейна. Ще одна метрика дорівнює мінімальній ціні перетворення при умові, що дозволено використовувати тільки операції видалення/вставки, це еквівалентно присвоюванню заміні ціни 2, а видаленню і вставці ціни 1, тому що операцію заміни можна замінити парою – видалення-вставка. Описана метрика називається відстанню редагування.

Оскільки опечатки мають неоднакову частоту для різних пар букв, необхідно задати функцію вартості перетворення:  $\omega: (S \cup \{\xi\})^2 \rightarrow Z_+$ ,

де  $\xi$  - порожній символ, так що  $\omega(\xi, a)$  - вартість вставки символу  $a$ ,  $\omega(\beta, \xi)$  - вартість видалення  $\beta$ ,  $\omega(\beta, \alpha)$  - вартість заміни  $\beta$  на  $\alpha$ ,  $\omega(\alpha, \alpha) = 0$  для будь-якого  $\alpha$ .

Таким чином, відстань Левенштейна доцільно використовувати для визначення ступеня близькості строкових значень і знаходження «подібних» значень ключових атрибутів.

Метод розширення вибірки. Досить часто, переважно в програмах перевірки орфографії, застосовують метод розширення вибірки. Суть методу полягає в зведенні пошуку за подібністю до точного пошуку. Для цього формується безліч всіх «помилкових» слів, які отримують з пошукового зразка в результаті однієї-двох операцій редагування: вставки, заміни, видалення і транспозиції, після чого побудовані терміни шукаються в словнику (на точну відповідність). Метод відмінно працює зі словниками натуральної мови, розмір яких відносно малий.

Сигнатурні алгоритми. Основою сигнатурних алгоритмів є літерне семпсування. У разі хешування по сигнатурі семпл перетвориться в сигнатурний вектор, який можна розглядати як запис числа в двійковій формі. Таким чином, хеш-функція  $H(a)$  однозначно визначає перетворення  $F(w)$  рядка в ціле число. Функція  $F(w)$  є хеш-функцією і може бути використана для процедури індексації словника.

Практично всі сигнатурні алгоритми, зокрема хешування за сигнатурою, досить чутливі до вибору параметрів і хеш-функції  $H(a)$ . Розмір сигнатури не повинен бути дуже великим або дуже маленьким. В обох випадках пошук не може здійснюватися ефективно. Необхідно також враховувати, що числа, які перебираються в процесі пошуку записів, лінійно залежать від розміру словника. Швидкість пошуку в сигнатурному *хеш* також істотно залежить від вибору хеш-функції  $H(a)$ .

Метод хешування по сигнатурі володіє наступними перевагами: дозволяє здійснювати з високою швидкістю пошук на точну рівність і пошук, що допускає одну-дві «помилки» в заданому пошуковому запиті; при правильному виборі параметрів обсяг індексу не більше ніж на 10-20% перевищує розмір файлу, що містить список термінів словника; відрізняється простотою реалізації.

Хешуванню по сигнатурі притаманні і досить суттєві недоліки: він повільно працює, якщо списки слів з однаковими сигнатурами розкидані по несуміжних секторах на диску. В даний час це не є великою проблемою, тому що, з одного боку, розміри пам'яті комп'ютера часто дозволяють завантажити словник цілком, а з іншого боку, дефрагментація словника, як правило, здійснюється за кілька хвилин. Іншим істотним недоліком є занадто великий розмір кінцевої вибірки, якщо слова відрізняються більше ніж на два символи.

Алгоритм N-грам. Алгоритм N-грам представляє модель послідовностей (зокрема природної мови) з використанням статистичних властивостей N-грам. Основна ідея застосування методу N-грам полягала в тому, щоб враховуючи послідовність букв, розрахувати ймовірність появи в послідовності кожної букви. N-грам модель прогнозує  $x(i)$  на основі значень  $x(i-1)$ ,  $x(i-2)$ , ...,  $x(i-n)$ . З точки зору теорії ймовірності дана модель може бути представлена формулою:  $P(x(i) | x(i-1), x(i-2), \dots, x(i-n))$ .

N-грам моделі широко використовуються в статистичній обробці природної мови, в тому числі в задачах розпізнавання мови, фонем, а також послідовностей фонем. При цьому послідовності букв моделюються з урахуванням специфіки та лексики кожної мови.

Розглядають два основні завдання, які потрібно вирішити, використовуючи алгоритм N-грам:

- спосіб розбиття на грами, перед розбиттям необхідно визначити порядок грам, найбільш вигідний для застосування на конкретній вибірці, а також ту умову, чи слід враховувати початок і кінець слова. Від цього буде залежати швидкість, точність і обсяг зберігання грам;
- спосіб підрахунку релевантності. Функція релевантності в даному алгоритмі має виняткове значення. Дана функція представляється як відношення кількості співпадань грам до загального числа, буквених поєднань.

Необхідно також відзначити основний недолік методу N-грам - відносно великий обсяг дискового простору, необхідний для зберігання грам. Однак, даний недолік компенсується точністю пошуку.

Алгоритми послідовного перебору. Застосування даних алгоритмів обумовлено їх високою ефективністю. При послідовному переборі рядки зчитуються послідовно і порівнюються безпосередньо з пошуковим зразком. Для порівняння рядків використовуються бітові алгоритми типу агрег. Незважаючи на те, що алгоритм послідовного перебору працює відносно повільно, далеко не всі альтернативні алгоритми, як показали експерименти, набагато ефективніше простого послідовного перебору. Зокрема, для максимально допустимої відстані редагування, рівної двом, більшість алгоритмів на практиці виявляються повільнішими.

Найпростіше рішення задачі перебору полягає в послідовному порівнянні, починаючи з  $t(1)$  і  $p(1)$ , символів  $T$  і  $P$  до тих пір, поки не буде виявлено рівність чи нерівність порівнюваних символів. В останньому випадку слід повернутися до початку порівняння і, змістившись на один символ по тексту (тепер це буде  $t(2)$ ), повторити процедуру.

Наприклад, пошук виконується за шаблоном  $vivid$  в тексті  $vivi \& dv \& vivid$ . Момент неспівпадання символів шаблону і тексту відзначається великою літерою. Результати представлені в таблиці 1.

У даному прикладі перші чотири символи співпадають, а п'ятий – ні. Продовжувати пошук, починаючи з  $\&$ , не можна, оскільки третій і четвертий символи тексту ( $vi$ ) співпадають з початком шаблону і можуть бути початком точного співпадання, тобто потрібно організувати перебір всіх початкових варіантів, як показано на прикладі в таблиці 1.

Даний алгоритм вимагає виконання не менше  $n-m+1$  порівнянь і працює досить повільно. Разом з тим він відносно простий для реалізації і дозволяє проводити пошук з використанням шаблону будь якої довжини.

Таблиця 1

Приклад роботи алгоритму послідовного перебору

	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

Точний пошук по алгоритму Бойер-Мура. Точний пошук по алгоритму ЗСУВ-І (SHIFT-AND). Алгоритм SHIFT-AND показав хорошу швидкість пошуку і достатньо просто програмується. Крім того, даний алгоритм володіє унікальною особливістю: він може бути легко модифікований для задач наближеного пошуку.

Дерева пошуку. Тріє- дерева представляють собою структуру, пошук в якій заснований на представленні терміна послідовністю символів. На відміну від звичайних збалансованих дерев в Тріє-дереві всі рядки, що мають спільний початок, розташовуються в одному піддереві. Кожне ребро позначено деяким рядком. Термінальним вершинам («листка») відповідають слова списку. У разі невдачі пошук повертає термін словника, що співпадає з шуканим зразком у найбільшій кількості початкових символів. Зазвичай Тріє-дерева використовуються для пошуку по підрядку, але їх можна ефективно використовувати для пошуку за подібністю.

Коренем дерева є вершина, якій відповідає «порожнє» слово довжиною нуль. З неї виходить стільки дуг, скільки символів зустрічається на першій позиції в термінах словника - щонайбільше стільки, скільки символів в алфавіті. Вершини другого рівня відповідають символам другої позиції, і так далі. Кожна кінцева вершина - термін, символи якого записані в єдиному шляху з кореневої вершини в цю кінцеву. Побудову дерева може бути вироблено послідовним додаванням нових термінів. Додавання здійснюється наступним чином: встановлюємо покажчик на кореневу вершину дерева, номер позиції в слові - на нуль; читаємо символ з поточної позиції; якщо з поточної вершини є дуга, що відповідає прочитаному символу, встановлюємо покажчик на відповідну вершину, збільшуємо на одиницю номер позиції в слові і переходимо на крок 2; якщо слово прочитане цілком, воно вже присутнє в словнику. Інакше додаємо до поточної вершині дугу, позначену прочитаним символом, переходимо до створеної вершини, збільшуємо номер позиції в слові на одиницю і повторюємо крок.

Метричні дерева - алгоритми, які використовують для індексування набору відстаней до деяких утворюючих точок. Різні алгоритми метричних дерев, зокрема  $vr$ -дерева використовують розбиття простору на підпростори в залежності від відстані елементів підпростору до утворюючих точок. Для практичних цілей придатні алгоритми, розмір індексу яких лінійно залежить від числа проіндексованих записів. За умови лінійності індексу і наявності цілочисельної функції відстані найшвидший пошук забезпечують БК-дерева. БК-дерева будуються таким чином: випадковим чином вибирається корінь дерева. Вся множина елементів розбивається на піддерева. Елемент  $w$  розміщується в піддереві номер  $k$ , якщо відстань редагування від  $w$  до кореня рівна  $k$ . Для кожного піддерева вибирається випадковим чином новий кореневий елемент, і процес індексації повторюється рекурсивно.

Пошук в БК-дереві ґрунтується на нерівності трикутника, якщо відстань від пошукового зразка до кореня дерева рівна  $p$ , то шуканий елемент може перебувати тільки в піддереві з номерами від  $p$  - до  $p + k$ , де  $k$  - максимально допустима відстань редагування.

Переваги: ефективно зберігаються локалізовані в просторі групи об'єктів; якщо дерево збалансовано, то забезпечує швидкий пошук в гіршому випадку; вставка/видалення однієї точки не вимагає суттєвої перебудови дерева (динамічний індекс).

Таблиця 2

Порівняльна характеристика алгоритмів пошуку

Алгоритми	Ефективність	Швидкість	Розмір дискового простору	Результат
Розширеної вибірки	висока	висока тільки при розмірі словника до 500 тис. записів	низький	Не підходить, в зв'язку з низькою швидкістю на словниках від 5 млн. записів
N-грам	висока	середня, лінійно залежить від довжини	високий	підходить

		рядків, можливо збільшити за допомогою хешування і індексування		
Дерева пошуку	низька	висока	середній	підходить як спосіб індексування
Відстань між рядками	низька	вище середнього	низький	підходить як спосіб сортування, ранжування результатів іншого алгоритму (N-грам)
Хешування по сигнатурі	нижче середнього	вище середнього	середній	підходить
Послідовний перебір	висока тільки при пошуку з малою кількістю опечаток по великому масиву тексту або при порівнянні на повну відповідність	вище середнього	низький	підходить

Недоліки: чутливі до порядку даних, що додаються; дані в листках можуть перекриватися; підходить для точного пошуку, при наближеному повертає занадто великий набір.

Алгоритми фонетичної схожості. Для англійської мови давно існують і активно використовуються алгоритми SoundEx і MetaPhone, які створюють для прізвища ключ, причому схожим прізвищам або невірним написанням прізвища буде відповідати один ключ. Якщо співробітник банку невірно набере прізвище клієнта, він все одно отримає доступ до його запису в базі даних, так як пошук ведеться за ключам. Зазвичай така система використовується для пошуку можливих правильних альтернатив для слова, написаного з помилкою.

**Висновок.** Сучасні алгоритми словникового пошуку ефективніші алгоритмів послідовного перебору, але при роботі з рядками великої довжини послідовний перебір істотно виграє в швидкості. З найбільш ефективних алгоритмів, розглянутих в роботі, слід зазначити алгоритми N-грам, trie-дерев, а також сигнатурні алгоритми, які забезпечують оптимальне співвідношення між розміром індексу і швидкістю пошуку. Порівняльна характеристика алгоритмів пошуку з точки зору ефективності, швидкості і займаного дискового простору представлено в таблиці 2. Вимоги до алгоритмів розташовані в порядку убудання їх важливості - Ефективність => Швидкість => Розмір дискового простору.

#### ЛІТЕРАТУРА:

1. Ахо А. Структуры данных и алгоритмы / А. Ахо, Д. Хопкрофт, Д.Ульман. – М.: Вильямс, 2009. – 400 с.
2. Вирт Н. Алгоритмы и структуры данных / Н. Вирт. – М.: ДМК Пресс, 2010. – 272 с.
3. Гагарина Л.Г. Разработка и эксплуатация автоматизированных информационных систем: учеб. пособие / Л.Г. Гагарина, Д.В.Киселев, Е.Л.Федотова. – М.: ИД «Форум»: Инфа-М, 2007. – 384 с.
4. Гагарина Л.Г. Алгоритмы и структуры данных / Л. Г. Гагарина, В.Д. Колдаев. – М.:Инфра-М, 2009. – 304 с.

5. Гайдамакин Н.А. Автоматизированные информационные системы, базы и банки данных / Н.А. Гайдамакин. – Москва «Гелиос АРВ», 2002. – 368 с.
6. Кнут Д.Э. Искусство программирования / Д.Э. Кнут. – Том 4. – Вып. 2. Генерация всех коротежей и перестановок. – М: Вильямс, 2008. – 160 с.
7. Макленнен Д. Microsoft SQL Server 2008 / Макленнен Д., Танг Ч., Криват Б.// Data Mining - интеллектуальный анализ данных. – СПб.: БХВ-Петербург, 2010. – 700с.

**Рецензент:** к.т.н., доц. Муляр И.В., доцент кафедры компьютерных систем та мереж Хмельницького національного університету

**Галянт С.В., к.т.н., доц. Красильников С.Р., к.т.н. Джулий А.В.  
АНАЛИЗ СОСТОЯНИЯ ПРОБЛЕМ ИДЕНТИФИКАЦИИ  
ОБЪЕКТОВ В БАЗАХ ДАННЫХ**

*В статье проведен анализ проблем идентификации объектов в базах данных.*

*Современные алгоритмы словарного поиска эффективнее алгоритмов последовательного перебора, но при работе со строками большой длины последовательный перебор существенно выигрывает в скорости. Из наиболее эффективных алгоритмов, рассмотренных в работе, следует отметить алгоритмы N-грамм, trie-деревья, а также сигнатурные алгоритмы, которые обеспечивают хорошее соотношение между размером индекса и скоростью поиска. Приведена сравнительная характеристика алгоритмов поиска с точки зрения эффективности, скорости и необходимого дискового пространства*

*Ключевые слова:* идентификации объектов, базы данных, алгоритм поиска, релевантность.

**Galyant S.V., PhD. Krasilnikov S.R., PhD. Dzhuliy A.V.  
ANALYSIS OF THE STATE IDENTIFICATION OF PROBLEMS  
OBJECTS IN DATABASES**

*The Article conducted analysis problems Authentication objects in the databases of data.*

*Modern algorithms vocabulary Search alhorytmov consistently more efficient brute force, but in terms of the work co Bolshoi dliny posledovatelnyy busting vuyhryvaet significantly in speed. Because most effektivnyh alhorytmov, rassmotrennyh in the work, It should be noted algorithms N-grams, trie-derevyu, as well as syhnaturnye algorithms, kotorye obespechuyayut GOOD sootnoshenye Between Size Index and skorostyu search. Present Comparative characteristics alhorytmov Search with point of view of the effectiveness, velocity and neobhodymoho disk space*

*Keywords:* Authentication objects, базы Data, Search algorithm, relevance.