МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ

ФАКУЛЬТЕТ МЕХАТРОНІКИ ТА КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

Кваліфікаційна магістерська робота

на тему: Розроблення програмного забезпечення для створення інформаційно-пошукової системи

Виконала: студентка групи МгІТ-1-22
спеціальності 122 Комп'ютерні науки
освітньої програми Комп'ютерні науки
Богдан МНОЖИНСЬКИЙ
Керівник:
к.т.н., доц. Тетяна АСТІСТОВА

Рецензент _____
д.т.н., проф. Володимир Щербань

Київ 2023

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ

Факультет мехатроніки та комп'ютерних технологій

Кафедра комп'ютерні науки

Спеціальність 122 Комп'ютерні науки

Освітня програма Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерні науки

_____Володимир ЩЕРБАНЬ

«_____»_____2023_____року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТА

Множинському Богдану Георгійовичу

1.Тема роботи: Розроблення програмного забезпечення для створення інформаційно-пошукової системи,

науковий керівник роботи: Астістова Тетяна Іванівна, к.т.н., доц.,

затверджені наказом закладу вищої освіти від 12. 09. 2023 року , № 210-уч.

2. Строк подання студентом роботи 12.11. 2023р.

3. Вихідні дані до роботи:

Розробка кафедри комп'ютерних наук

4. Зміст дипломної роботи (перелік питань, які потрібно розробити)

РОЗДІЛ 1. Теоретична частина; РОЗДІЛ 2. Методи та алгоритми пошукових систем; РОЗДІЛ 3. Програмна реалізація модулів системи. Додатки - програмні коди модулів системи

5.Консультанти розділів кваліфікаційної магістерської роботи

Розділ	Ім’я, прізвище та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Вступ	Тетяна АСТІСТОВА, к.т.н., доц.		
Розділ 1	Тетяна АСТІСТОВА, к.т.н., доц.		
Розділ 2	Тетяна АСТІСТОВА, к.т.н., доц.		
Розділ 3	Тетяна АСТІСТОВА, к.т.н., доц.		
Висновки	Тетяна АСТІСТОВА, к.т.н., доц.		

1. Дата видачі завдання: 08. 2022

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної магістерської роботи	Терміни виконання етапів	Примітка про виконання
1	Вступ	20.08.2023	
2	Розділ 1. Теоретична частина.	10.09.2023	
3	Розділ 2. Методи та алгоритми пошукових систем.	5.10.2023	
4	Розділ 3. Програмна реалізація модулів системи.	25.10.2023	
5	Висновки.	28.10.2023	
6	Оформлення дипломної магістерської роботи (чистовий варіант)	31.10.2023	
7	Здача кваліфікаційної магістерської роботи на кафедру для рецензування	01.11.2023	
8	Перевірка кваліфікаційної магістерської роботи на наявність ознак плагіату	03.11.2023	
9	Подання кваліфікаційної магістерської роботи на затвердження завідувачу кафедри	07.11.2023	

Студент _____ Богдан МНОЖИНСЬКИЙ

Науковий керівник роботи _____ Тетяна АСТІСТОВА

Директор НМЦУПФ _____ Олена ГРИГОРЕВСЬКА

АНОТАЦІЯ

Множинський Б.Г. Розроблення програмного забезпечення для створення інформаційно-пошукової системи

Кваліфікаційна магістерська робота за спеціальністю 122 - «Комп'ютерні науки». – Київський національний університет технологій та дизайну, Київ, 2023 рік.

В першому розділі роботи було досліджено та проведено аналіз існуючих всесвітньо відомих інформаційно-пошукових систем та топових українських; розглянуто основні терміни, поняття, функції пошукових систем.

В другому розділі було проведено аналіз алгоритмів, методів пошуку та індексації в документах; розроблена ER-діаграми пошукової бази, структури таблиць інформаційної бази, розроблено алгоритми пошуку інформації та оцінки релевантності знайдених документів. Аналіз алгоритмів показав недооцінення алгоритму релевантності знайдених документів.

В третьому розділі показана структура модулів пошукової системи, структура програми, опис бази даних, опис процедур, SQL, як мова структурованих запитів; розроблена блок-схема процедури додавання індексу до бази індексів та блок-схема процедури редагування ключових слів. Розглянуто пошуковий агент під Windows - центральним модулем роботи з даними є модуль даних DataModuleSearcherм , а пошуковий клієнт реалізований за допомогою Borland Delphi 7. Показан алгоритм роботи пошукового агента для Інтернету та фрагменти програм в мові PHP.
Проведено аналіз релевантності та вагових коефіцієнтів html-документу.

Розроблена система може бути корисною як для досліджень у галузі пошуку документів та оцінки релевантності, так і для промислового використання.

Ключові слова: релевантність, інформаційно-пошукова система, ER-діаграма, DataModuleSearcherм, Borland Delphi 7, html , SQL, PHP.

ANNOTATION

Mnozhynskyi B.G. Development of software for creating an information search system.

Qualifying master's thesis in specialty 122 - "Computer science". - Kyiv National University of Technology and Design, Kyiv, 2023.

In the first section of the work, the existing world-famous information and search systems and top Ukrainian ones were researched and analyzed; the main terms, concepts, functions of search engines are considered.

In the second section, an analysis of algorithms, methods of search and indexing in documents was carried out; developed ER-diagrams of the search database, structures of the information database tables, developed algorithms for searching for information and evaluating the relevance of found documents. Analysis of the algorithms showed that the algorithm underestimated the relevance of the found documents.

The third section shows the structure of search engine modules, program structure, database description, procedure description, SQL as a structured query language; a block diagram of the procedure for adding an index to the index database and a block diagram of the procedure for editing keywords have been developed. The search agent under Windows is considered - the central module for working with data is the DataModuleSearcherm data module, and the search client is implemented using Borland Delphi 7. The algorithm of the search agent for the Internet and fragments of programs in the PHP language are shown.

An analysis of the relevance and weighting coefficients of the html document was carried out

The developed system can be useful both for research in the field of document retrieval and relevance assessment, and for industrial use.

Keywords: *relevance, information search system, ER diagram, DataModuleSearcherm, Borland Delphi 7, html, SQL, PHP*.

ЗМІСТ

ВСТУП

Актуальність теми. Сьогоднішній день є початком епохи електронного проникнення глобальної комп'ютерної мережі Internet у всі сфери людської життя. Більшість сучасних людей користуються Інтернетом, як найдоступнішим джерелом інформації. Технологія повністю перевернула уявлення про роботу з інформацією та з комп'ютером взагалі

За оцінкою експертів, обсяг інформації, що передається каналами Інтернет, подвоюється кожні півроку. Щодня в мережі з'являються мільйони нових документів, і, природно, що без систем пошуку вони в переважній більшості залишилися б не затребуваними, взагалі не були б ким знайдені, і все те величезна кількість інформації виявилося б нікому не потрібним.

Виникла необхідність створення таких засобів, які б легко орієнтуватися в інформаційних ресурсах глобальних мереж, швидко і надійно знаходити потрібні відомості. В Інтернеті з'явилися спеціальні пошукові засоби.

Основні функції пошукових систем:

- збір інформації з Інтернету та її обробка. Інформація міститься в сховище даних у зручному для пошуку інформації ;
- видавати інформацію користувачу за його запитом. Максимально корисна інформація знаходиться вгорі, і далі по списку вниз.

Актуальність задачі, пов'язаної із розробкою програмного забезпечення для створення інформаційно-пошукової системи системи визначається низкою факторів:

- у зв'язку з зростаючим обсягом інформації потрібні нові пошукові засоби;
- необхідно підвищити якість (релевантність) інформації. Проблема в тому, що її не можна чітко визначити.

Мета роботи *:* Розробка програмного забезпечення для створення інформаційно-пошукової системи.

Об’єкт дослідження. Огляд сучасних пошукових систем та їхньої структури. Дослідження різних варіантів здійснення пошуку інформації та алгоритмів пошуку, визначення релевантності документів.

Завдання дослідження. Провести аналіз процесу організації пошуку у Всесвітній Глобальній мережі та розробити структуру пошукової системи; розробка алгоритмів пошуку та визначення релевантності документів; дослідження розроблених алгоритмів. Розробити програми основних модулів інформаційно-пошукової системи, що демонструє механізм пошуку інформації в Інтернеті .

Предмет дослідження кваліфікаційної магістерської роботи є розгляд новостворених алгоритмів пошуку та визначення релевантності документів, аналіз різних параметрів документів, що враховуються під час здійснення пошуку, розробка бази для інформаційної системи .

Методи дослідження. Методами дослідження виступає аналіз алгоритмів, методів пошуку та індексації в документах та блок-схеми процедур пошукових агентів.

Наукова новизна та практичне значення. Практична цінність кваліфікаційній магістерській роботи полягає в висновках, які були отримані при проведенню аналізу розроблених алгоритмів, визначено ваги різних тегів, що враховуються під час пошуку. Оцінено якість пошуку за допомогою розроблених алгоритмів в тому, що її основні положення, висновки та результати дають можливість організувати процес пошуку інформації у Світовій мережі **.** Забезпечення якіснішого (релевантного) знаходження інформації є основною задачею розробки**.**

Результати роботи були опубліковані в наступних виступах та статтях:

1. Астістова Т. І., Розробка інформаційно-пошукової системи для автоматизованого збору даних / Т. І. Астістова, Б. Г. Множинський // Інформаційні технології в науці, виробництві та підприємництві : збірник наукових праць молодих вчених, аспірантів, магістрів кафедри комп’ютерних наук та технологій / за заг. наук. ред. В. Ю. Щербаня. – Київ : ТОВ "Фастбінд Україна", 2023. – С. 108-111.
https://er.knutd.edu.ua/handle/123456789/24122

РОЗДІЛ 1. ТЕОРЕТИЧНИЙ РОЗДІЛ

1.1. Постановка задачі

Кожен користувач Інтернету може знайти безліч різноманітної та цікавої інформації, а також використовувати всі найбагатші можливості мережі. Ресурси Інтернету вже давно перестали бути просто розвагою, перетворившись на незамінний інструмент для повсякденної роботи людей багатьох професій. Швидке зростання інформації в мережі зробили його океаном різноманітних даних, важливість яких зростає пропорційно до їх обсягу.

Ще 15 років тому була така думка: «В Інтернеті є все, але знайти там нічого неможливо». Однак з появою та швидким розвитком пошукових каталогів, пошукових машин, і всіляких пошукових програм ситуація змінилася, і тепер в Мережі терміново знадобилася інформація іноді можна знайти швидше, ніж у книзі, що лежить на столі.

Найбільш популярним та застосовуваним способом пошуку в Інтернеті є використання пошукових систем. Першочергове завдання будь-якої пошукової системи – доставляти людям саме ту інформацію, яку вони шукають.

Існують неузгодженості між інформаційними потребами конкретного користувача та їх вираження у вигляді запитів на пошук інформації, підготовлених для тієї чи іншої інформаційної системи. Для того, щоб користувач міг чітко висловити свої інформаційні потреби, необхідний певний рівень знань саме в тій галузі, інформацію з якої він бажає отримати. Виникає відомий парадокс: для того, щоб правильно сформулювати питання, потрібно по суті знати відповідь.

Крім того, користувачу недостатньо просто добре орієнтуватися в області, що його цікавить. Він повинен мати також уявлення про суміжні області, для того щоб відобразити у своєму запиті кордону тематики, що його цікавить .

Наступна проблема, з якою стикається користувач, як сформулювати запит.

Ця проблема складається із двох частин:

- Необхідністю опанувати мову запитів конкретної інформаційної системи.

Сучасні системи надають своїм користувачам багато можливостей, деякі з яких вимагають від користувача спеціальних знань, наприклад, логіки алгебри. Крім того, багато можливостей частково дублюють одна одну. Наприклад, логічні операції та операції ``+ '', ``- ''. Багато функцій додаються лише оскільки такі функції є у конкуруючих систем.

- Необхідність збору статистики про конкретну пошукову систему.

Терміни, що включаються в запит, повинні мати значну дискримінаційну силу, інакше кажучи, не розумно включати запит ті терміни, які у цій колекції є загальновживаними, тобто. зустрічаються майже у кожному документі. Далі необхідно оцінити реальну користь від використання різних можливостей пошуку, що надаються цією системою.

На відміну від мови запитів до реляційної бази даних, наприклад SQL, які дозволяють отримати саме ті дані, які запитуються, в документальних пошукових системах результат пошуку залежить від величезної кількості факторів. Один з таких факторів - алгоритм пошуку, що використовується в системі, який часто є комерційним секретом. Лише тривала активна робота з конкретною системою дозволить виявити сильні та слабкі сторони даної системи.

В результаті аналізу статистики запитів великої кількості користувачів до відомих пошукових систем з'ясувалося, що середня довжина запиту не перевищує двох слів, і зазвичай користувачі використовують найпростішу форму запиту.

Проведений аналіз дозволяє зробити наступний висновок: не можна сподіватися, що користувач пошукової системи буде формулювати досить складні ефективні запити. Сучасна пошукова система повинна виявляти інформаційні потреби конкретного користувача та враховувати їх при пошуку на користь даного користувача. Під час роботи з пошуковою

системою користувач висловлює свої інформаційні потреби мовою запитів даної системи та сподівається отримати у результаті пошуку посилання найбільш релевантні запиту документи.

Поняття релевантності використовується у світі інформаційного пошуку дуже широко. Однак його не можна чітко визначити. Можна навести лише наступний приклад, що дає опосередковано уявлення про сенс цього поняття. Для оцінки якості пошукових систем проводять різні тестові випробування. Найбільш відомою системою тестування є TREC – TExt Retrieval Conference. Це система щорічних конференцій (з 1992 року), на яких проводиться тестування різних пошукових систем на одному множині спеціально підготовлених тестів. Наприклад, для проведення тестувань на п'ятій конференції (TREC-5) були підготовлені тестові дані, що включають близько 260 тисяч текстових документів загальним обсягом більше гігабайта текстової інформації. Для проведення тестування в TREC використовуються підготовлені експертами запити, заздалегідь відомі номери релевантних документів. Звичайно, експерти не спроможні оцінити релевантність кожного документа для кожного запиту. У зв'язку з цим, на першому етапі використовується кілька різних пошукових систем, які формують для кожного запиту безліч документів, що визнаються хоча б однією із цих систем релевантними запиту [1-3].

На другому етапі всі відібрані документи аналізуються експертами, які формують безліч ідеальних релевантних документів для кожного запиту.

Приклад TREC демонструє роль експертів у визначенні релевантності того чи іншого документа заданому запиту. Різні користувачі мають різні інформаційні потреби, які не відображаються повною мірою запитом. В результаті можлива ситуація, коли два користувачі задають той самий запит, але вважають релевантними різні документи. Слід зауважити, що інформаційні системи створюються не для одного користувача, і експерт може розглядатися як представник усієї спільноти користувачів, що представляє його інформаційні потреби.

Відповідно до обраного статистичного підходу до всіх проблем інформаційного пошуку, доречно порушити питання про принципову можливість автоматичного виконання будь-яких оцінок релевантності в умовах, коли самі користувачі можуть мати різні думки з цього приводу.

Чи доступна якась об'єктивна інформація щодо релевантності.

Нехай є деяка колекція документів, і для кожного з них експерт визначив кілька інших документів з тієї ж колекції, які можна розглядати як близькі цієї колекції. Нехай є певний евристичний алгоритм оцінки близькості довільної пари документів у межах цієї колекції, який використовує статистичний підхід. Побудуємо для кожного документа його околицю - безліч документів з цієї колекції, визнаних даним алгоритмом близькими даному документу. Тоді ймовірність того, що даний документ близький з точки зору експерта документу, випадково обраному з його околиці, вища за ймовірність того, що випадково обраний з колекції документ визнається експертом близьким даному документу.

1. 2. Основні терміни та поняття пошукових систем.

Представлятимемо Web як сукупність сайтів, кожен з яких містить безліч документів.

$$Web = \underset{i}{Y}\, Сайт_i \qquad (1.1)$$

$$Сайт = \underset{j}{Y}\, Документ_j \qquad (1.2)$$

Автори створюють документи, а в певної групи людей виникає інформаційна потреба. Ця потреба часто не може бути навіть точно виражена словами, і виражається тільки в оцінці документів, що переглядаються - підходить або не підходить. У теорії інформаційного пошуку замість слова "підходить" використовують термін пертинентний (від англ. Pertinent - відноситься до справи, що підходить по суті), а замість "не підходить" - "не пертинентний". Суб'єктивно зрозуміла мета інформаційного пошуку - знайти всі пертинентні і лише пертинентні документи (знайти "тільки те, що треба, і нічого більше") [8].

Ця мета – недосяжна. Людина в більшості випадків може оцінити пертинентність документа тільки в порівнянні з іншими документами. Для того, щоб було з чим порівнювати, необхідна деяка кількість документів, які не перетинаються. Ці документи називають шумом. Занадто великий шум ускладнює виділення пертинентних документів, занадто малий - не дає впевненості у тому, що знайдено достатню кількість пертинентних документів, що перетинаються.

Практика показує, що коли кількість документів, які не перетинаються, лежить в інтервалі від 10% до 30%, той, хто шукає, почувається комфортно, не гублячись у морі шуму і вважаючи, що кількість знайдених документів – задовільно.

Коли багато документів то для пошуку використовується інформаційно-пошукова система (ІПС). У цьому випадку інформаційна потреба має бути виражена у вигляді фрази (запиту) спеціальною інформаційно-пошуковою мовою (ІПЯ) [7].

Запит рідко може точно висловити інформаційну потребу, якщо це, звичайно, не запит природною мовою. Однак багато ІПС з різних причин не можуть визначити, чи той чи інший документ відповідає запиту. Ступінь відповідності документа запиту називається релевантністю. Релевантний документ може бути непертинентним і навпаки. [13]

Існують два основні підходи до пошуку інформації:

- перший полягає у використанні спеціальних тематичних каталогів;
- другий – у пошукових машинах (search engine), заснованих на індексі.

Найбільш продуктивною схемою пошуку є комплексне використання цих двох базових підходів. Тому сучасні пошукові системи надають користувачам можливість використання обох підходів.

Інформаційно-пошукова система (ІПС), засновані на каталогах, називатимемо класифікаційними ІПС, а ІПС, що використовують пошукові

машини, - словниковими.

В основі складання тематичних каталогів лежить принцип класифікації, тобто розподіл документів (або цілих сайтів) з ієрархії тематичних рубрик (класифікатору). Класифікатор розробляється та вдосконалюється колективом авторів. Усередині рубрик документи упорядковані за рейтингами популярності. Класифікацію здебільшого також виробляють вручну, оскільки цей процес важко здійснити автоматично.

Каталоги можуть містити реферати документів та короткі описи сайтів. Кількість класифікованих документів мізерно мала проти їх загальної кількісті в Web, тому каталоги неспроможні дати вичерпних відомостей з певної тематиці. Системи, засновані на каталогах, краще підходять у тих випадках, коли треба здійснити швидкий пошук будь-яких загальних укрупнених тих чи користувачів, які займаються "вільним ковзанням" по Мережі.

Невеликі розміри та створена людьми система упорядкування матеріалу роблять їх особливо придатними для швидкого знаходження якісної інформації. У разі робиться ставка саме якість інформації, а чи не її кількість [6, 7].

Класифікаційні ІПС мають низку специфічних недоліків:

- Розробка класифікатора пов'язані з оцінкою відносної важливості різних галузей людської діяльності. Наприклад, порівнюючи класифікатори багатьох ІПС Інтернет (таких, як Yahoo, Excite, Look Smart), помічаємо, що у багатьох немає розділу "Наука".
- Крім того, у створенні таких ІПС беруть участь також колективи систематизаторів, які виносять свої суб'єктивні оцінки про відповідність документів рубрикам класифікатора.

Будь-яка оцінка є соціальною дією; вона пов'язана із суспільством, культурою, соціальною групою, до яких належить людина, яка виносить оцінку. Тому тематичні каталоги, створені різними колективами у різних країнах, можуть мати дуже різний рівень корисності при пошуку інформації

– все залежить від того, хто і що шукає. Отже, під час пошуку інформації з допомогою ІПС, заснованої на каталозі, виникає необхідність взаємодії коїться з іншими культурами - культурами авторів, творців класифікаторів і систематизаторів.

Недоліки:

- не можуть дати вичерпних відомостей з певної тематики;
- здійснюючи пошук необхідно взаємодіяти з культурами розробників класифікатора та систематизаторів.

Культурні проблеми, пов'язані з використанням класифікаційних ІПС, призвели до створення ІПС словникового типу.

З погляду «недосвідченого» користувача словникова ІПС – це засіб пошуку документів за ключовими словами. Сценарій пошуку простий: користувач за допомогою мови запитів висловлює те, що він хоче знайти, і буквально за кілька секунд отримує список посилань на документи, що задовольняють його запиту. Розглянемо принцип дії словникових ІПС. Усі словникові ІПС мають загальну структуру, яка наведена на рис. 1.1.

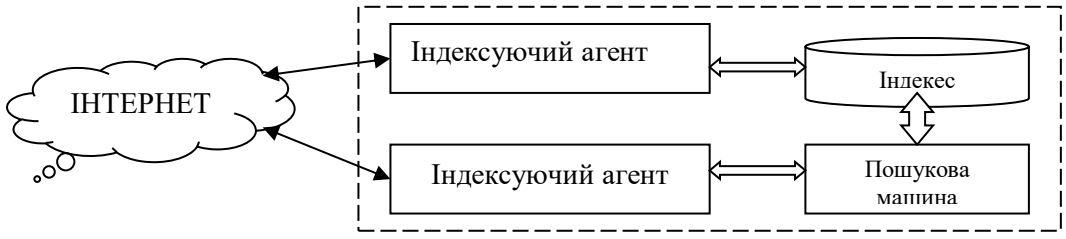

Рисунок 1.1 - Загальна структура словникової ІПС

З погляду внутрішньої організації словникова ІПС і двох частин, зазвичай, працюючих паралельно. Перша частина (індексуючий агент (agent), павук (spider), робот (robot)) відповідальна за індексування Web-документів, а друга (пошукова машина) здійснює пошук документів за індексом відповідно до

запитів користувачів.

Основна ідея словникової ІПС - створити словник (індекс) зі слів, які у документах Інтернет, у якому кожному слову відповідатиме список документів, які містять. Якщо пошук слів у такому словнику виконується швидко, можна відмовитися від дорогих послуг розробників класифікаторів і систематизаторів [9].

Індексуючий агент методично переглядає документи Web, переходячи від одного документа до іншого за допомогою гіпертекстових посилань усередині документів. До кожного зустрінутого документа проводиться виділення індексної інформації та збереження їх у базі даних, званої індексом.

З філософської точки зору кожен документ можна уявити, що складається з двох частин: «важливої» і «неважливої».

$$\{\text{Документ}\} = \{\text{Важлива частина}\} \cup \{\text{Неважлива частина}\} \qquad (1.3)$$
$$\{\text{Неважлива}\} \cap \{\text{Важлива частина}\} = \emptyset$$

В ідеалі, до індексу має потрапляти лише «важлива» частина документа, а «неважлива» частина губитися. Однак алгоритми індексування, що застосовуються різними ІПС, відрізняються.

Для зберігання індексу часто використовують системи управління базами даних (СУБД). У разі пошук по індексу здійснюється з допомогою вбудованих коштів СУБД.

Кожна ІПС має власний ІПС, на якому необхідно формулювати пошукові запити до неї. Для здійснення повнофункціонального пошуку з використанням кількох ІПС користувачеві доведеться вивчити кілька ІПС.

За участю 15 найбільших пошукових систем Інтернету в лютому 1999 року стартував проект SESP (Search Engine Standards Project), який має стандартизувати роботу пошукових служб.

Завданням стандарту є максимально зблизити синтаксис та можливості

ІСЯ різних ІПС. Зокрема, однією з обов'язкових вимог стає підтримка будь-якої пошукової системою єдиних команд запитів, що локалізують вузол за його доменним ім'ям, а документ - за URL [14].

Великі обсяги баз даних роблять словникові ІПС особливо корисними вичерпних пошуків, складних запитів чи локалізації неясної інформації. Ця гідність, однак, стає пасткою, коли відбувається швидкий пошук. Більшість таких систем полегшує сприйняття надмірної кількості інформації, впорядковуючи результати пошуку так, щоб посилання з найвищим рівнем відповідності запиту розташовувалися вище.

Підсумуємо переваги ІПС словникового типу.

- широке охоплення web-ресурсів;
- не потрібна дорога ручна праця розробників класифікатора та систематизаторів.

Підсумовуючи, можна сказати, що єдиної оптимальної схеми пошуку в Інтернеті не існує. Залежно від специфіки необхідної інформації, для її пошуку слід використовувати відповідні пошукові служби. В принципі, звичайно, можна завжди користуватися якоюсь однією пошуковою системою, але чим грамотніше підібрані пошукові служби і складено запит на пошук інформації, тим якіснішими будуть результати пошуку.

1.3. Основні функції пошукових систем.

Пошукова система – це певна база даних, онлайн - служба, що надає можливість шукати інформацію в Інтернеті. Більшість із них працює з сайтами, окремі здатні знаходити файли на FTP-серверах. Основним завданням пошукової системи – систематизувати та задовольнити запит максимально релевантною та необхідною для користувача інформацією.

Основні функції пошукових систем*:*

1) Сканування Internet**.**

Багато систем пошуку мають у своєму складі мережевих роботів. Відповідно до деякого спеціального алгоритму мережевий робот переглядає

загальнодоступні інформаційні ресурси, опубліковані в Internet, та копіює нові (оновлені) документи, інформація про які відсутня (застаріла) у базі даних (індексі) пошукової системи. При цьому зазвичай виконується певна фільтрація, що дозволяє копіювати в базу даних документи, для пошуку яких призначена дана пошукова система.

2) Індексування документів.

Документи, що знову надійшли в пошукову систему, індексуються. Іншими словами, для кожного документа формується його пошуковий образ (профайл), що включає інформацію, яка далі використовуватиметься при пошуку. У найпростішому випадку сюди включаються ключові слова, що відображають зміст документа, та його URL-адресу місця розміщення цього документа в Internet..

3) Пошук.

Отримавши запит користувача, наприклад кілька слів, введених у спеціальну форму запиту, пошукова система використовує свій індекс для пошуку документів, найбільш близьких (з її точки зору) запиту. Всі такі документи ранжуються за ступенем близькості запиту, і список їх URL з деякою додатковою інформацією (наприклад, заголовок документа), повертається користувач

Для збору інформації про сайти та аналізу їхнього контенту існують боти. Наприклад, запити на Google Analytics заходять як реальні користувачі, так і безліч роботів з різних сервісів.

Робот аналізує, збирає , копіює інформацію (текстовий контент, зображення, медіафайли) та передає її в сховище даних. Далі пошукова система здійснює також аналіз, структурує та визначає релевантність сайту і його сторінок пошуковим запитам.

Саме на етапі аналізу відбувається визначення релевантності сторінок сайту і ранжування за певними запитами.

Одним з основних критеріїв пошукової видачі є її релевантність- це відповідність введеного в пошук слова, словосполучення до результатів

видачі.

Що може впливати на позиції документів у списку пошукових систем:

- наявність слів, що введені в пошуковий запит у документах. Якщо документ містить слова з вписаного користувачем запиту, то документ відповідає критеріям пошуку.
- – частота входження слів. В залежності від частоти вживання того чи іншого слова в документі, залежатиме його позиція в пошуковій видачі.

Сьогодні алгоритми роботи пошукових систем змінились і боти суттєво удосконалились. Роботи аналізують текст в цілому, а в алгоритми закладено багато різних факторів. Це значно покращало перелік релевантних посилань.

Топові запити можуть суттєво відрізнятись у різних країнах та навіть регіонах однієї держави. На них впливають сезонність, політична та безпекова ситуація, тобто дуже багато різних факторів.

Наприклад, у Google в 2021-2022 роках частіше за все шукали «Гру в кальмара» та «Венома». Гуглили інформацію про відомих особистостей. Наприклад, цікавились Олександром Усиком, Сергієм Стерненком, Джо Байденом, купувати іт-квитки на поїзд, iPhone 14, вітамінами.

Якщо подивитись на запити українців в 2022-2023 роках, то переважно цікавляться війною з рашистами , технікою, товарами для ЗСУ, генераторами, павербанками, зарядними станціями., цікавилися зверненнями президента та зведеннями генштабу.

При глобальному погляді на пошуковики, то сюди відносяться запити стосовно погоди, шукають переклади тих чи інших слів.

Слід відзначити, що є різниця між браузером та пошуковою системою.

Браузер – це інструмент, за допомогою якого користувач входить в Інтернет. В браузері можна користуватися різними пошуковими системами, або встановити одну з них за замовчуванням. А пошуковик – це система, яка знаходить релевантні запиту сайти.

1.4. Огляд пошукових систем

1.4.1. Популярні зарубіжні пошукові системи.

Серед українських користувачів найпопулярнішими залишаються зарубіжні сервіси. Першу позицію вже дуже багато років утримує Google. Гугл - одна з найпопулярніших у світі.

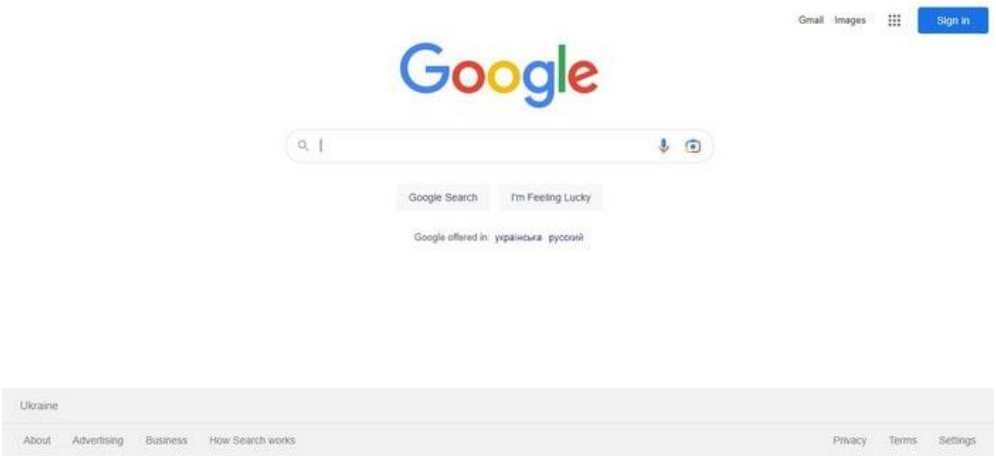

Рисунок 1.1- Google – найбільша пошукова система

Google – найбільша пошукова система в світі, заснована в 1998 році, виробляє близько 2 трильйонів запитів та проводить індексацію 25 млрд веб-торінок щомісяця. Система автоматизована, має власні алгоритми ранжування, роботи постійно сканують інтернет та індексують все нові й нові сайти.

До основних переваг системи відноситься:

- - точність та зручність – знайти релевантні результати не є проблемою;
- - велика кількість програм та додатків. (пошта, диск для зберігання даних, перекладач, карти інше)
- - потужно інтегрована морфологія – навіть при формуванні некоректного запиту фраза буде вирівняна, а результати в повній мірі відповідатимуть запиту;
- - висока комерціалізація – багато реклами;
- - збір інформації про будь-які дії користувача (можна пропонувати певні товари інтернет-магазинів, що проіндексовані системою);

- -зручний пошук (через зображення).

Більше 7% Інтернет - користувачів обирає саме цей пошуковик тому, що більшість мобільних браузерів має предустановлений Google як пошуковик за замовчуванням. Сьогодні в Інтернеті майже кожний проєкт просувається з огляду на алгоритми Google.

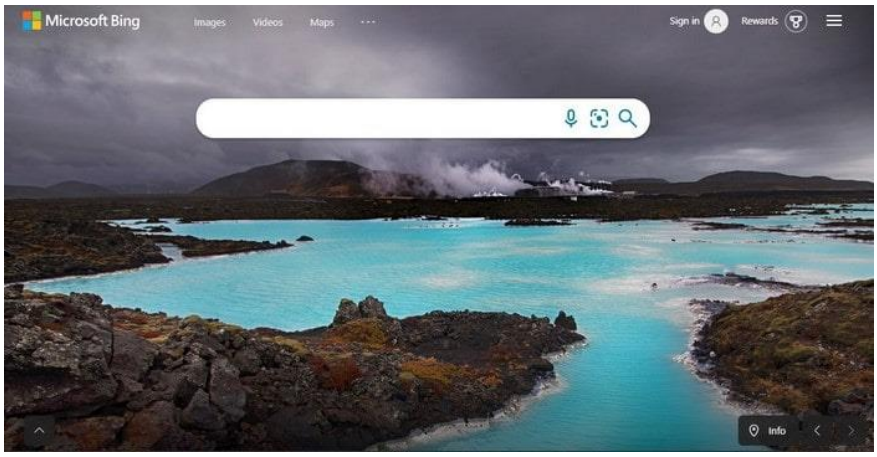

Рисунок 1.3- Bing - пошукова система від Microsoft

В світових рейтингах та в українському Інтернеті система Bing займає другу позицію, але цією пошуковою системою мало хто користується. З'явилася вона відносно нещодавно – в 2009 році. Система використовується лише один раз, щоб завантажити Google. Bing – це вбудований пошуковик Microsoft, він встановлюється разом з Windows. Особливості Bing:

- дає можливість скоригувати та налаштувати меню під потреби користувача;
- шукає тексти, зображення, відео, географічні об'єкти
- працює на основних мовах Європи та Азії;
- цензурує відверті дані;
- знижує у видачі сайти, які порушують авторські права.

Ця система користується популярністю в США та Канаді. Особливо цінують користувачі цих стран мінімалістичний дизайн, функцію візуального пошуку. Вони не прагнуть шукати якісь інші варіанти та користуються тією пошуковою

системою, що вже встановлена.

3. Yahoo! – найстаріший пошуковик (рис.1.4).

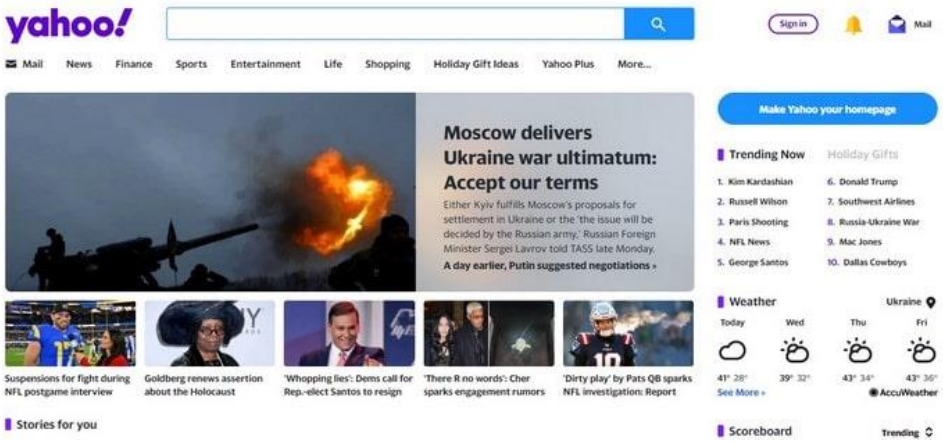

Рисунок 1.4- Yahoo! – найстаріший пошуковик

Yahoo! одна із найстаріших та найвідоміших пошукових систем. З'явилася вона в 1996 році. В Україні входить у топ-5, але частка користувачів менше 1%. Найбільшу популярність отримав цей пошуковик у США.

В 2009 році Yahoo! припинив існування в якості окремого проекту, його придбав Microsoft.

Особливості системи:

- власна пошта;
- каталог файлів;
- прогноз погоди;
- новини та інші сервіси.

Пошукову систему Yahoo! неможливо встановити, вона використовується виключно через браузер.

1.4.1. Топові українські інформаційно-пошукові системи.

В Україні помітна тенденція збільшення кількості саме українських пошуковиків. На сьогодні функціонує більше 20 пошукових систем, що

зорієнтовані на вітчизняний інтернет - простір та користувача, що проживає в Україні. Основна причина розвитку – запит суспільства, адже іноді трапляється що пошук інформації перетворюється в виключення ru сегменту для виокремлення саме українських ресурсів.

З'являються нові пошуковики регулярно, зазвичай як відгалуження новостійних порталів, поштових сервісів, тощо.

Українських пошуковиків є багато популярних, які активно використовуються, буквально 3-5. Топові українські пошуковіки .

1**).** Українська пошукова система I.ua (рис.1.4).

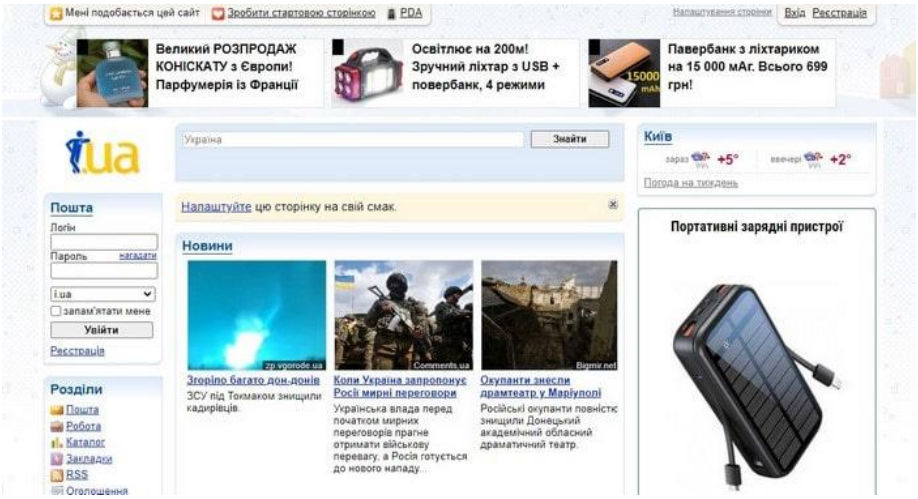

Рисунок 1.4 -Українська пошукова система i.ua

Найбільша українська пошукова система I.ua, з'явилась з інтернет-порталу Цей портал був дуже популярним і практично єдиним у своєму роді.

Пошук зорієнтований на україномовні сайти. Максимально простий інтерфейс – немає нічого зайвого, проте досить багато реклами. Крім рядка для запиту, є каталог з іншими можливостями порталу: пошта, пошук роботи, каталог, закладки, погода, гороскоп, телепрограма, курси валют, новини, кіно, рецепти, музика та інше.

2). Українська пошукова система Ukr.net (рис.1.5).

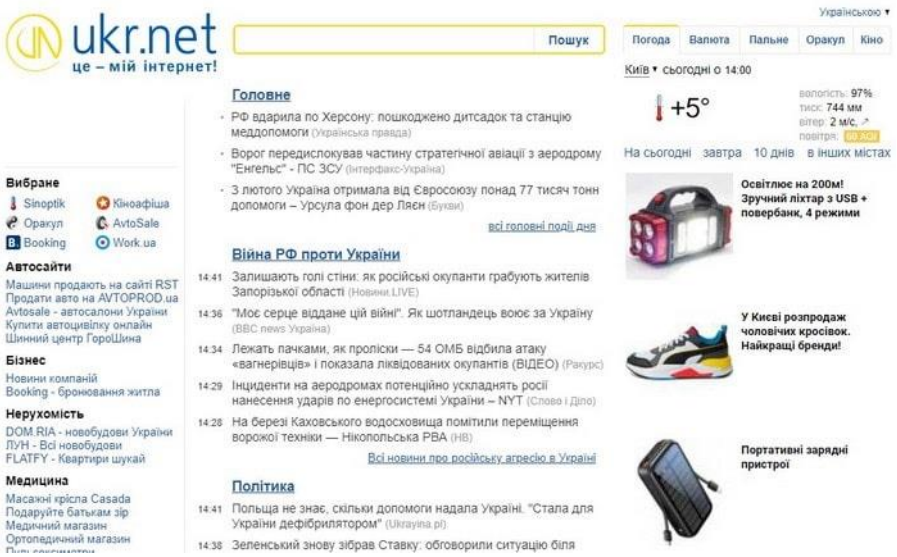

Рисунок 1.5- Українська пошукова система Ukr.net

Ukr.net – відомий портал новин, який має власний поштовий сервіс дуже популярний серед користувачів. Цікаво, що навіть враховуючи розміщення Пошуковий рядок розташован над стрічкою новин, що є помітним місцеі, але багато користувачів його просто не помічають і зі здивуванням виявляють, що через УкрНет можна ще й інформацію шукати.

Аналогічно до Google, ресурс пропонує інтелектуальні підказки. Якщо просто почати вводити свій запит, система запропонує свої варіанти. Результати можна відфільтрувати по:

- країні – Україна чи інші;
- мові – українська чи інші;
- конкретному сайту – задайте URL.

Система дає можливість активувати безпечний пошук та обрати мову – українську чи російську. Пошукова система Ukr.net достатньо зручна та добре продумана.

На цьому порталі можна створити електронну пошту та почитати останні новини, які зручно структуровані по тематиці.

3). Українська пошукова система META (рис.1.6).

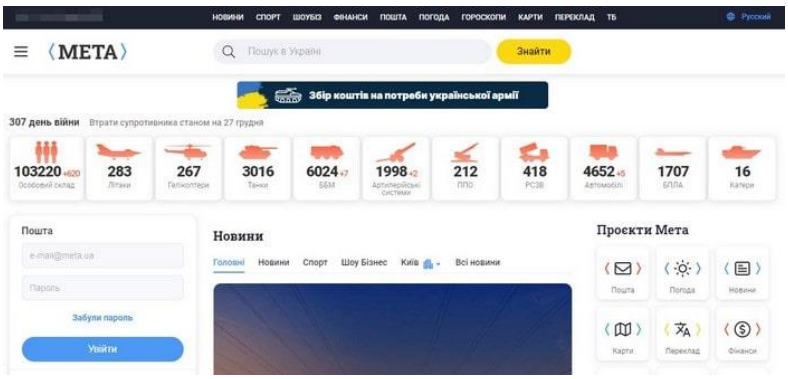

Рисунок 1.6- Українська пошукова система META

Українська пошукова система META – повнотекстовий пошуковик з оригінальною базою даних, що підтримує розвинену мову запитів. В пошуковій системі доступний пошук за окремими полями документів, відбувається сортування результатів з урахуванням морфології української, російської та англійської мов. Всі посилання супроводжуються анотаціями. Перегляд результатів зручний та швидкий.

Особливість цієї пошукової системи є те, що в зоні пошуку знаходяться всі ресурси, що стосуються України, а для створення запитів використовується Google. В системі також є додаткові корисні функції: пошук через слова, на сторінці чи вибране на пошук.

Приналежність сайтів до ua сегменту визначається наступним чином:

- домени та піддомени .ua;
- українська мова на сайті;
- хостинг на IP українських провайдерів;
- основна тематика, незалежно від мови, стосується України.

Дана пошукова система була заснована в Харкові у1998 році. Сьогодні пошукова система META має розгалужену систему додаткових сервісів, у тому числі досить специфічних: соціальну мережу для створення ділових та особистих контактів – «МетаКонтакт», відео ресурс «МетаВідео» та інші.

Висновки до першого розділу

Пошукові системи постійно розвиваються та вдосконалюються, щоб надавати своїм користувачам саме ті результати, які вони прагнуть отримати, навіть у тому випадку, коли запити формулюються не дуже коректно. Крім популярних пошукових систем відомих не лише в Україні, але і у всьому світі, активно розвиваються вітчизняні, які дають багато додаткових можливостей і орієнтовані саме на українського користувача**.**

Поява нових пошукових систем відбувається регулярно, як відгалуження порталів новин, поштових сервісів, тощо. В результаті аналізу статистики запитів з'ясувалося, що середня довжина запиту не перевищує двох слів, і зазвичай користувачі використовують найпростішу форму запиту. В цьому розділі було розглянуто основних термінів та понять пошукових систем.

Проведений аналіз дозволяє зробити наступний висновок - користувач пошукової системи не буде формулювати досить складні ефективні запити. Сучасна пошукова система повинна виявляти інформаційні потреби конкретного користувача та враховувати їх при пошуку на користь даного користувача.

РОЗДІЛ 2. МЕТОДИ ТА АЛГОРИТМИ ПОШУКОВИХ СИСТЕМ

2.1. Методи пошуку та індексації

Розглянемо методи пошуку та індексації в документах

1) Пошук фрагмену тексту в документах.

Пошук фрагмену тексту в документах є пошуком, що використовується в документах Microsoft Office або Windows Commander. Реалізується метод мовою Паскаль за допомогою однієї команди – pos.

Метод має ряд недоліків:

- неможливість знайти текст, якщо слова запиту безпосередньо не йдуть одне за одним;
- •неможливо визначити релевантність знайдених документів запиту користувача.

Алгоритми метода використовують для пошуку на локальних сторінках одного або декількох сайтів, а у великих пошукових системах не використовуються

2) Пошук за частотою слів.

В цьому методі потрібно сформувати базу даних, що містить частоту входження слів у документ і пошук здійснюється в цій базі. Результати видаються відповідно до частотних характеристик. Чим частіше зустрічаються слова, то релевантніший результат.

Складністю методу в оцінки релевантність документа. Наприклад, частота зустрічі слів у одному запиті різна : структура – 5%, а комп'ютера – 2%.

Особливістю цього алгоритму є необхідність створення пошукової бази і пошук інформації тільки в ній. Цей метод не застосовується для пошуку на локальних сайтах.

3) Пошук за морфологічними особливостями мови документа.

Кожен документ представляється як певна математична модель. Пошук враховує всі варіації мови – закінчення, коми, крапки, тощо. Метод більше підходить для знаходження інформацію на запит, ніж для визначення

релевантність документа.

Метод, як складова частина, використовується на кількох серверах із запитами у вигляді звичайних пропозицій Ask.com.

4) Метод пошуку по структурі документа.

Документи розбиваються на конструктивні елементи – заголовки, підзаголовки, малюнки тощо. і потім зміст цих елементів заноситься до пошукової бази.

При пошуку враховується, де саме знайдені слова запиту і відповідно до якого алгоритму ранжуються результати. Якщо слова знайдено у заголовку, то документ вважається більш релевантним, ніж той документ, де слова знайдені у підзаголовку.

Цей метод забезпечує прекрасну оцінку релевантності, але є досить трудомістким і містить складнощі у визначенні методів ранжирування конструктивних елементів.

2.2. Пошукови алгоритми.

Переважна більшість пошукових алгоритмів ґрунтується на так званій "Векторній моделі тексту", запропонованій Дж. Солтоном (Salton G.) у 1975 році. Робота Солтона є теоретичною основою сучасних пошукових систем у їхній класичній реалізації.

Різні автори називають цю модель індексування та пошуку по-різному: векторної, лінійної, або алгебраїчної.

Векторною, будемо називати модель опису інформаційного масиву, а лінійною – модель пошуку інформації в масиві. Такий поділ зумовлений тим, що документи записуються у вигляді двійкових векторів, тоді як пошукові запити – це лінійні перетворення над цими векторами.

У векторній моделі інформаційного потоку можна виділити кілька основних понять: словник, документ, потік та процедури пошуку та корекції запитів.

Поняття інформаційного потоку:

1. Під словником розуміють впорядковану множину термінів, потужність

якого позначають як D.

2. Документ - це двійковий вектор розмірності D. Якщо термін входить у документ, то у відповідному розряді цього двійкового вектора проставляється 1, інакше - 0. Усі операції в лінійній моделі індексування та пошуку у документах виконуються над пошуковими образами документів, але при цьому їх, зазвичай, називають просто документами.

3. Інформаційний потік або масив L представляють у вигляді матриці розмірності *NxD*, де як рядки виступають пошукові образи N документів.

$$L = \left\{ \forall i = \overline{1,N};\ j = \overline{1,D} : b_{ij} = \begin{cases} 0, t_j \notin l_i \\ 1, t_j \in l_i \end{cases} \right\} \quad (2.1.)$$

де: t_i – термін, l_i – документ.

При такому розгляді можна сформулювати процедуру звернення до інформаційної системи так:

$$L \times q = r;$$

де: q – запиту, , r – вектор відгуку системи на запит.

Для опису роботи розподілених інформаційно - пошукових систем (ІПС) Інтернету, використовуються інформаційно-пошукові мови типу "Like This". Цей підхід поклала система WAIS. Саме в ній було вперше сформульовано відмову від використання традиційних інформаційно-пошукових мов бульового типу, і перенесення центру тяжкості інформаційного пошуку мовами, заснованими на обчисленні міри близькості "документ-запит".

Система проводила нормалізацію лексики та видаляла зі списку термінів запиту загальні та стоп-слова, тобто виконувались умови лінійної моделі індексування та пошуку. Потім система обчислювала міру близькості

за виразом і відповідно до отриманих значень ранжувала інформаційний масив. Майже всі ІПС Інтернету влаштовані за цим принципом.

Можна навести багато заходів близькості, однак у сучасних розподілених ІПС Інтернет реально використовуються лише кілька

$$R_{i,q} = \sum_{j=1}^{M} C_{i,j} \qquad (2.2)$$

$$R_{i,q} = \sum_{k=1,k=i}^{N} \left(Li_{i,k} \times \sum_{j=1}^{M} C_{k,j} \right) \qquad (2.3)$$

$$R_{i,j} = \frac{\sum_{t \in q} (0.5 + 0.5 \frac{TF_{i,j}}{TF_{i,\max}}) \cdot IDF_j}{\sqrt{\sum_{t \in P_i} (0.5 + 0.5 \frac{TF_{i,j}}{TF_{i,\max}})^2 \cdot (IDF_j)^2}} \qquad (2.4)$$

Де: $R_{i.q}$ – міра близькості документа i та запросу q;

M – число термінів запиту;

P_i – i-ий документ індексу;

$Li_{i,k}$ – {1, якщо документ k має посилання на документ i ; 0, інакше };

$Lo_{i,k}$ – {1, якщо з документу i є спосилання на документ k; 0, інакше},

$C_{i,j}$ – {1, якщо документ i містить термин j; 0, інакше}

Найбільш популярними є:

- Розширений двійковий алгоритм пошуку;

- Алгоритм найбільшого цитування;

- TFxIDF алгоритм (запропонований та уточнений Солтоном у 1979 році);

- Розширений векторний алгоритм пошуку.

Слід зазначити, що найбільш ефективним із цих алгоритмів є TFxIDF, який і використовується в більшості ІПС (RBSE, WAIS, WebCrawler, Lycos, OpenText, Lycos та Altavista).

Одним з компонентів міри близькості TFxIDF є частота термінів в масиві документів.

Зазвичай щільність функції розподілу частоти термінів, що зустрічаються, описують гіперболічним розподілом, відомим як закон Зіпфа.

Якщо щільність підпорядковується гіперболічному закону, немає жодних чітких меж виділення термінів зі словника. Якщо щільність задається розподілом з яскраво вираженим максимумом, терміни повинні вибиратися з околиці цього максимуму.

Суть алгоритму Солтона в тому, що для індексування використовують ті терміни, які мають високу частоту всередині документа і низьку у всьому інформаційному масиві. Сама характеристика обчислюється як відношення частоти терміна в документі до частоти терміна в масиві.

З огляду експериментальних результатів можна зробити два висновки:

- щоб використовувати зважування, слід мати насичений словник;

- терміни індексування знаходяться на околицях максимуму частотного розподілу термінів.

Насичення словника – дуже важлива властивість систем із вільним словником. Говорити про векторну модель інформаційного потоку та її застосування для інформаційних систем можна, коли потужність словника (число представлених у ньому термінів) фіксована.

Поки йшлося про локальні інформаційні системи, то питання про розмір словника не стояло. За час експлуатації системи (з моменту

завантаження документів і до моменту актуалізації) інформаційний масив та словник системи не змінювалися, були фіксованими. В Інтернеті справа зовсім інша.

По-перше, немає єдиного інформаційного масиву. Система постійно здійснює сканування мережі та корекцію свого пошукового апарату – словника, що визначається індексом, який постійно змінюється.

По-друге, через відсутність єдиної інформаційної служби не можна організувати систему з контрольованим словником.

Таким чином, в ІПС Інтернет відбуваються два процеси: постійне зростання інформаційного масиву, з одного боку, і постійне збільшення словника системи, з іншого.

Але і Lycos, і OpenText, і Altavista, та інші системи Інтернету застосовують лінійну модель індексування та пошуку, використовуючи різницю терміна в алгоритмах автоматичного індексування та пошуку. Алгоритми, що застосовуються, обмежують словник, допускаючи його незначне зростання, в словник потрапляють тільки терміни пошукових образів. Джерелом термінів індексування є не весь документ, а окремі його частини: заголовок, гіпертекстові посилання, підзаголовки, спеціальні поля. Отже вдається контролювати розмір словника і залишатися в рамках лінійної моделі індексування та пошуку.

2.3. Порівняння параметрів для визначення релевантності документів

Конструктивні елементи, які використовуються щодо релевантності деякими пошуковими системами .

Розглянемо слова - теги: Title, Description, Keywords, H1-H6,Alt, Посилання,Текст.

Повні алгоритми пошуку тримаються в секреті, все ж таки можна зробити деякі зауваження щодо них.

Усі системи враховують конструктивні елементи сторінок, такі як теги H1-H6, виділені слова (Strong). Популярний елемент Alt тега IMG, що є

поясненням до малюнка, використовується у всіх системах

З мета-тегами системи працюють менш ефективно. Так, ключові слова Keywords у жодній системі беруть участь у визначенні релевантності, а опис Description використовується лише частково на деяких. Заголовок сторінки Title враховують усі системи.

Деякі системи використовують для визначення релевантності сторінки кількість посилань на неї з інших сайтів (так званий індекс цитування). Іноді системи враховують кількість знайдених слів у документі (частота слів) та відстань між словами запиту.

Зробимо висновок: не все різноманіття тегів враховується пошуковими системами і немає не однієї, яка враховувала б всі з них. З іншого боку, аналіз показує, що значимість тегів щодо релевантності також відрізняється. Самі алгоритми пошуку дуже відрізняють пошукові системи друг від друга.

2.4 Розробка інформаційної бази

Нехай необхідно занести в пошукову базу інформацію із набору сайтів. Під сайтом розумітимемо набір ресурсів, розташованих в одному домені та його піддоменах.

Представимо безліч сайтів, що індексуються S множиною Кожен конкретний сайт уявимо як $S_x \in S$.

Тепер розглянемо безліч ресурсів (сторінок, файлів), які стосуються сайту. Кожен сайт може бути представлений як безліч сторінок $P_y \in P$. Тот факт, що сторінка відноситься до сайту S_x будемо позначати як $S_x.P_y$

Для індексації необхідно виділити конструктивні елементи, якими може проводитися пошук. У цьому бажано враховувати як найбільш поширені, а й потенційні елементи, які можуть стати в нагоді надалі. Такими елементами є різні теги.

Розглянемо теги, які має сенс індексувати (заносити до пошукової бази):
1) TITLE – один із головних тегів, який використовується при пошуку. Більшість пошукових систем так чи інакше використовують його для пошуку

та видачі результатів.

2) H1 – служить позначення заголовків (найбільших) всередині документів.

3) H2 – служить позначення підзаголовків заголовка H1.

4) H3 – заголовок.

5) H4 – заголовок.

6) H5 – заголовок.

7) H6 – заголовок

8) Мета-тег KEYWORDS – служить завдання ключових слів документа.

9) Мета-тег DESCRIPTION – служить завдання опису документа.

10) Alt – служить позначення спливаючої підписи для рисунка.

Безліч тегів, що належать документу $S_x.P_y$, будемо позначати як $S_x.P_y.D$ Кожен документ складається з тегів $D_i \in D$. Поставимо у відповідність кожному слову в документі елемент множини D, що має вигляд структури: $D_i.w$ - слово в документі, $D_i.r$ - вигляд тега, $D_i.p$ - номер сторінки, якій належить тег.

Тепер потрібно визначити розташування слів у тегу – щоб простежити сусідні слова та їх послідовність. Для цих цілей кожному слову необхідно поставити у відповідність його позицію. Крім того, у зв'язку з тим, що в документі може бути кілька однойменних тегів, необхідно встановити також приналежність кожного слова до певного тегу $D_i.g$ - Група (пропозиція), в якій знаходиться слово, $D_i.n$ - позиція в групі (пропозиції)

Наведені вище міркування слід доповнити наступним. Часто ключові слова та теги не відповідають змісту документа (або через помилку розробника або через так званий пошуковий спам, коли сторінки будуються спеціально для того, щоб здаватися пошуковим системам найбільш релевантними не є такими насправді).

Для перевірки відповідності ключових слів необхідно підрахувати кількість їх входження до документа. Ця статистика – досить показово, оскільки визначає дійсний стан речей у документі.

На основі перерахованих вище міркувань була розроблена структура інформаційної бази пошукової системи(рис. 2.1)

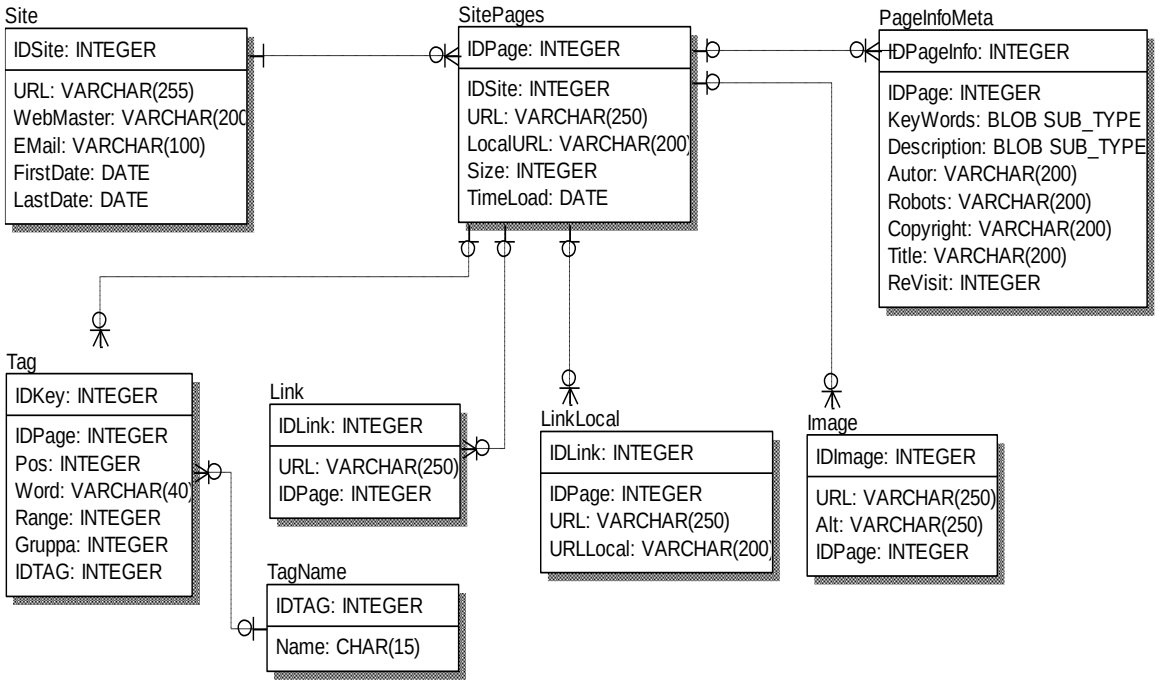

Рисунок 2.1 – ER-діаграма пошукової бази

Опишемо ці таблиці:

1) Site – таблиця індексованого сайту. Містить інформацію про сайт повністю. Ця таблиця відповідає множині S

Таблиця 2.1 - Структура таблиці Site

Поле	Тип	Опис
IDSite	INTEGER	ID сайта
URL	Varchar(255)	Адреса сайту
WebMaster	Varchar(200)	Дизайнер
Email	Varchar(100)	E-mail дизайнера
FirstDate	Date	Дата занесения в базу
LastDate	Date	Дата останньої переіндексації

Таблиця 2.2 – Структура таблиці SitePages

Поле	Тип	Опис
IDPage	INTEGER	ID страницы
IDSite	INTEGER	ID сайта
URL	Varchar(255)	Адрес сайта
LocalURL	Varchar(200)	Локальный адрес сайта

Size	INTEGER	Размер страницы
TimeLoad	Date	Время загрузки

3) TagName – таблицая імен тегів.

Таблица 2.3 – Структура таблицы TagName

Поле	Тип	Описание
IDTag	INTEGER	ID тега
Name	Char(15)	Имя тега

4. PageInfoMeta – таблиця агрегованої інформації про сторінку сайту.

Таблиця 2.4 – Структура таблиці PageInfoMeta

Поле	Тип	Опис
IDPageInfo	INTEGER	ID записи
IDPage	INTEGER	ID сторінки
Keywords	BLOB	Ключеві слова
Description	BLOB	Опис
Autor	Varchar(200)	Автор
Robot	Varchar(200)	Робот
Copyright	Varchar(200)	Права
Title	Varchar(200)	Заголовок
Revisit	INTEGER	Як часто відновлювати

5) Tag - таблиця проіндексованих слів сторінки (множина D).

Таблиця 2.5 – Структура таблиці Tag

Поле	Тип	Опис
IDPage	INTEGER	ID сторінки
IDKey	INTEGER	ID слова
Word	Varchar(40)	Слово
Pos	INTEGER	Позиція
Gruppa	INTEGER	Група
Range	INTEGER	Частота зустрічі
IDTAG	INTEGER	ID тега

6) Link – таблиця посилань зі сторінки на інші сайти

Таблиця 2.6 – Структура таблиці Link

Поле	Тип	Опис

IDPage	INTEGER	ID сторінки
IDLink	INTEGER	ID посилання
URL	Varchar(255)	Адреса сторінки (посилання)

7. LinkLocal – таблиця посилань із сторінки на інші сторінки цього сайту.

Таблиця 2.7 – Структура таблиці LinkLocal

Поле	Тип	Опис
IDPage	INTEGER	ID сторінки
IDLink	INTEGER	ID посилання
URL	Varchar(255)	Адреса сторінки (посилання)
URLLocal	Varchar(200)	Локальна адреса сторінки

8. Image – таблиця рисунків на сторінці

Таблица 2.8 – Структура таблиці Image

Поле	Тип	Опис
IDPage	INTEGER	ID сторінки
IDImage	INTEGER	ID рисунка
URL	Varchar(255)	Адреса рисунку
ALT	Varchar(255)	Пояснювальний допис

Основною таблицею для пошуку буде таблиця Tag - таблиця проіндексованих слів сторінки (множина)..

2.5. Разработка алгоритму релевантності

Більшість пошукових систем використовують складні алгоритми неповного пошуку. Зазвичай пошук здійснюється за набором тегів. Для алгоритму релевантності важливим є не лише знаходження слів запиту (не обов'язково поряд один з одним) у документі, а й урахування важливості тегів. Наприклад, важливість тега TITLE значно перевищує тег H5 чи Strong

1). **Алгоритм під назвою "наближений" або “алгоритм-подібність**”.

Нехай безліч тегів T, тоді кожен конкретний тег можна обозначити як $t_i \in T$. Нехай користувач вводить запит W, що складається зі n слів , $w_j \in W \quad j = 1..n$

Кожному виду тегів ставиться у відповідність певна вага (або ваговий коефіцієнт) $v_i \in V$, який визначатиме важливість відповідного тега. Позначимо кількість входжень послідовності W в тег t_i через функцію $f(W,t_i)$ чи просто f_i. Тоді можна визначити вагу кожного документа

$$r_k = \sum_i f(W,t_i) \times v_i = \sum_i f_i \times v_i \qquad (2.5)$$

У цьому випадку, чим більша вага документа r_k, тим вища релевантність запиту користувача.

Таким чином, на основі (2.5) можна сформувати алгоритм пошуку цього алгоритму.

Знайти всі документи, в яких зустрічається шукана комбінація слів всередині одного тега і впорядкувати їх за сумарною вагою документів, яка визначається як сума тегів, в яких знайдено шукана комбінація слів.

У наведеному алгоритмі завдання зводиться до визначення ваг тегів $v_i \in V$.

Позначимо теги за спаданням їх важливості в документі: Title, Keywords, Description, H1, H2, …, H5, Strong, Alt (тег IMG).

Пошукова система повинна шукати не тільки за заданою послідовністю слів $w_j \in W$, що знаходяться поруч один з одним, але й розташовані на певній відстані один від одного (наприклад, "*розробка ІС*" та "*розробка* сучасних *ІС*"). У цьому випадку ми можемо ускладнити функцію $f(W,t_i)$, яка визначатиме не тільки кількість запитів, але й відстань між словами $w_j \in W$.

Позначимо її як $f'(W,t_i)$При цьому чим більше відстань між словами, тим менше значення набуває функція $f'(W,t_i)$.

Таким чином, вираз (2.6) набуває вигляду: .

$$r_k' = \sum_i f'(W,t_i) \times v_i = \sum_i f_i' \times v_i \qquad (2.6)$$

2**.) Частковий алгоритм**.

Часто необхідно знайти документ, у якому шукані слова перебувають у різних тегах. Звичайно при цьому релевантність документів нижча, але кількість документів набагато більша.

Знайти всі документи, в яких зустрічається набір шуканих слів і впорядкувати результати за сумою тегів, в яких знайдені шукані слова.

$$r_k'' = \sum_{i,j} f''(w_j, t_i) \times v_i \qquad (2.7)$$

Алгоритм передбачає знаходження будь-яких документів, де зустрічаються шукані слова, байдуже у яких тегах вони перебувають. Пошукові слова можуть бути у різних видах тегах. Це суттєво знижує релевантність одержуваних документів, але значно збільшується шанс знайти документ.

Тепер розглянемо безпосередньо вираз функцій $f(W,t_i)$, $f'(W,t_i)$, $f''(w_j,t_i)$

Розглянемо ці функції термінах реляційної алгебри, оскільки пов'язані саме з операціями над даними. Відповідно до термінів реляційної алгебри введемо такі позначення: означає операцію селекції, - σ проекції, π - операцію природної сполуки, $\triangleright\triangleleft$ - операцію природного поєднання

Тоді функцію $f(W,t_i)$ можна уявити як

$$f(W,t_i) = \pi_{D_i^1.r,\, D_i^1.p}\left(\sigma_{\substack{(D_j^1.w=W_1)\&(D_j^2.w=W_2)\&(D_j^1.r=D_j^2.r)\& \\ (D_j^1.p=D_j^2.p)\&(D_j^1.g=D_j^2.g)\&(D_j^1.r=t_i)}}(D_1, D_2)\right) \qquad (2.8)$$

Функция $f'(W,t_i)$ буде дорівнювати:

$$f'(W,t_i) = \pi_{D_i^1.r,\, D_i^1.p}\left(\sigma_{\substack{(D_j^1.w=W_1)\&(D_j^2.w=W_2)\&(D_j^1.r=D_j^2.r)\& \\ (D_j^1.p=D_j^2.p)\&(D_j^1.r=t_i)}}(D_1, D_2)\right) \qquad (2.9)$$

Функцию $f''(w_j,t_i)$ можно представити як:

$$f''(W,t_i) = \pi_{D_i^1.r,\, D_i^2.r,\, D_i^1.p}\left(\sigma_{(D_j^1.w=W_1)\&(D_j^2.w=W_2)\&(D_j^1.p=D_j^2.p)}(D_1, D_2)\right) \qquad (2.10)$$

Кожен алгоритм потрібно дослідити на показники точності та повноти.

Висновки до другого розділу.

В розділі розглянуто методи пошуку та індексації в документах: пошук за частотою слів , пошук за морфологічними особливостями мови документа, метод пошуку по структурі документа.

Аналіз цих методів показав недоліки кожного з них. Була розроблена інформаційну базу у вигляді ER-діаграми пошукової бази для здійснення у пошуку за певною тематикою та структура таблиць інформаційної бази.

Аналіз алгоритмів «наближений» та «частковий алгоритм» пошукових системи показав, що існуючі пошукові системи містять низку недоліків, але самий найсуттєвіший із них – це недооцінення алгоритму оцінки релевантності знайдених документів. Жодна з пошукових систем не містить алгоритму, що враховує релевантность всіх конструктивних елементів документів. З цієї причини розробка пошукової системи, що включає новий алгоритм визначення релевантності, стоїть досить актуально.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ

3.1. Розробка функціональної моделі

При розробці інформаційних систем для аналізу та проєктування, використовуються CASE-засоби, передбачається побудова структурних або інших діаграм.

Інформаційна модель, це сукупністю функціональної моделі та моделі даних.

Розробка функціональної моделі.

Для побудови функціональної моделі об'єкта будь-якої предметної області, використовується методологія SADT, яка є сукупністю методів, правил і процедур

Побудова моделі починається з представлення всієї системи у вигляді одного блоку та дуг, що зображають інтерфейси з функціями поза системою.

Блок, как єдиний модуль, деталізується за допомогою декількох блоків, з'єднаних інтерфейсними дугами. Ці блоки є основними підфункціями вихідної інформації.

Підфункції містять елементи, що входять у вихідну функцію. Результатом застосування методології SADT є модель, що складається з діаграм, фрагментів текстів та глосарію, які мають посилання один на одного.

Організація інформаційної бази.

Найбільш поширеним засобом моделювання даних є діаграми "сутність-зв'язок" (ERD). З їхньою допомогою визначаються важливі для предметної області об'єкти (сутності), їх властивості (атрибути) та відносини один з одним (зв'язки). ERD безпосередньо використовують для проектування реляційних баз даних.

3.2 Розробка структури та модулів пошукової системи

Рисунок 3.1 - Структура пошукової системи.

У найзагальнішому вигляді пошукову систему можно представити як:

1) Індексуючий агент,

2) Інформаційна (пошукова) база,

3) Пошуковий агент

Індексація сторінок складається з наступних етапів:

1. Отримання адреси початкової сторінки.
2. Визначення кодування сторінки та у разі потреби – перекодування.
3. Розбір сторінки та отримання інформації для запису до пошукової бази.
4. Підрахунок статистики народження слів.
5. Пошук адрес сторінок та додавання їх до списку індексування.
6. Поки список індексування не порожній, повторення кроку 2.

Модуль "Розбір сторінки" (Parser).

Завданням модуля є послідовний перегляд документа та знаходження в ньому тегів певної групи приналежності (TITLE, H1, H2 тощо). Далі

проводиться виділення тексту з тега і потім розбиття тексту на слова. Parser записує слова (WORD) у пошукову базу із збереженням порядку їхнього прямування в тілі тега (POS). Gruppa вказує у якому порядку тегу виявлено слово. Практична реалізація Parser досить складна, тому було обрано готовий компонент THTMLParser.

Підрахунок статистики зустрічі слів слугує компонент THTMLParser, як готових рішень у цій галузі. Підрахунок статистики слів проводиться на основі Parser, причому кількість слів заноситься в RANGE.

Модуль "Павук" (Spider).

Під павуком розуміється програма, яка послідовно переглядає документи і знаходить у них посилання на нові сторінки. Знайдені нові адреси додаються до списку сторінок. Дія продовжується для новознайдених сторінок і т.д., доки не буде перевірено весь сайт. Надалі сторінки цього списку підлягають розбору за допомогою Parser.

Модуль «Пошуковий агент».

Розглянемо структуру пошукового агента. З запиту користувача відбувається формування запиту до бази даних. Запит являє собою один або кілька запитів SQL до пошукової бази.

В результаті виходить набір даних результатів пошуку. Модуль оцінки релевантності забезпечує безпосередньо сам алгоритм, який ми називаємо пошуковим алгоритмом. На основі оцінки алгоритму виходить список результатів, відсортованих за спаданням релевантності.

Індексатор

Програма індексатора призначена для індексації сторінок. Робота програми складається з кількох етапів:

- Занесення інформації про сайт, що індексується
- Пошук сторінок на сайті за допомогою Spider (Павука)
- Розбір сторінок та занесення даних до пошукової бази даних

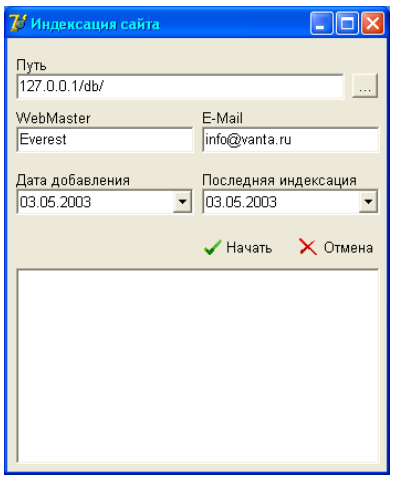

Рисунок 3.2 –Вікно «Індексація сайта»

Задаються такі дані про сайт:

- Шлях до сайту (URL).
- Ім'я розробника (WebMaster).
- E-mail розробника.
- Дата індексації (додавання до пошукової бази).
- Дата останньої індексації (при додаванні ідентична попередньому параметру).

Ця інформація додається вручну та заноситься до пошукової бази. Далі починається безпосередньо індексація сайту. Адреси оброблених сторінок відображаються у полі в нижній частині вікна.

Доступ до бази даних (MySQL) забезпечується за допомогою компонентів dbExpress. Для команд роботи з базою даних використовуються оператори мови SQL. Принцип роботи з даними більш детально описаний нижче.

3.3. Розробка програми.

3.3.1. Структура програми

Робота програми відбувається через основну форму. Навігація по функціональним елементам здійснюється за допомогою вкладок. Ієрархічної структури програми представлена як взаємодія функціональних елементів (рис.3.3).

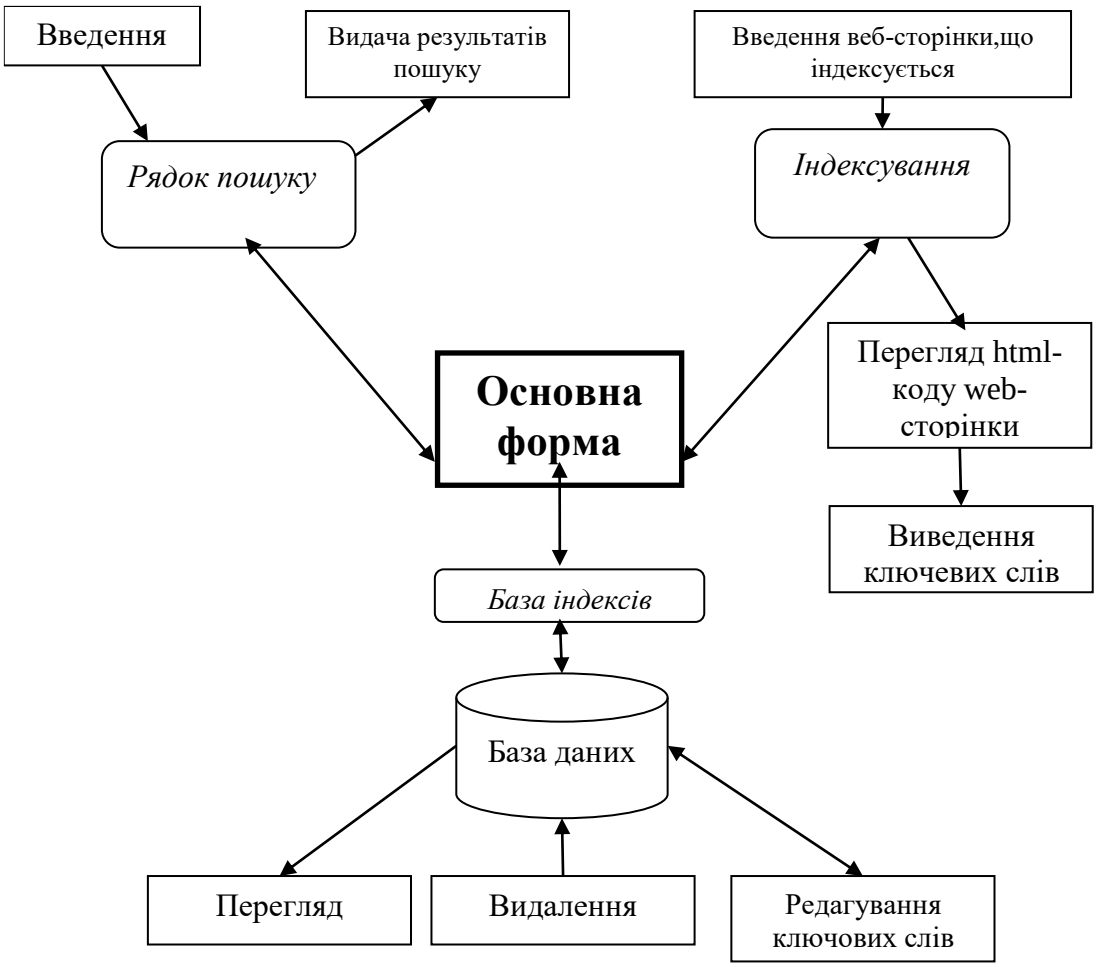

Рисунок 3.3 – Структура програми

Опис бази даних

Програма має у своєму складі базу даних, що складається з таблиці "Index", призначенням якої є зберігання та редагування індексів web-сторінок (таблиця 3.1).

3.1.- Таблица Index

Поле	Тип	Додотково	Опис
site	Текстовий	Ключе	Найменування web–сторінки
keywords	Поле MEMO		Список ключових слів, наявних на відповідній web–сторінці
Rating	Числовий		«Рейтінг» web–сторінки, тобто кількість її запитів

3.3.2. Опис процедур

Програма надає інтерфейс до таблиці бази даних та стандартні засоби керування даними за допомогою компонентів, що використовують технологію ADO (Active Data Objects).

Механізм доступу до даних через ADO та численні об'єкти та інтерфейси реалізовані у VCL Delphi у вигляді набору компонентів, розташованих на сторінці ADO. Усі необхідні інтерфейси, що забезпечують роботу компонентів, оголошено та описано у файлах OleDB.pas та ADODB.pas

Компонент TADOConnection забезпечує з'єднання із джерелами даних через провайдери OLE DB. Компоненти TADOTable та TADOQuery забезпечують використання наборів записів у програмі. Властивості та методи компонентів дозволяють створювати повнофункціональні програми

Компонент TADOConnection забезпечує доступ до сховища даних компонентів ADO, що інкапсулюють набір дани

Застосування цього компонента дає розробнику низку переваг:

- усі компоненти доступу до даних ADO звертаються до сховища даних через одне з'єднання;
- можливість безпосередньо задати об'єкт провайдера з'єднання;

- даних через одне з'єднання;
- можливість безпосередньо задати об'єкт провайдера з'єднання;
- доступ до об'єкта з'єднання ADO;
- можливість виконувати команди ADO;
- виконання транзакцій;
- розширене управління з'єднанням за допомогою методів-обробників подій.

На сторінці ADO Палітри компонентів Delphi, крім компонентів з'єднання, є стандартні компоненти, що інкапсулюють набір даних і адаптовані для роботи зі сховищем даних ADO відповідно до рисунку 3.4.

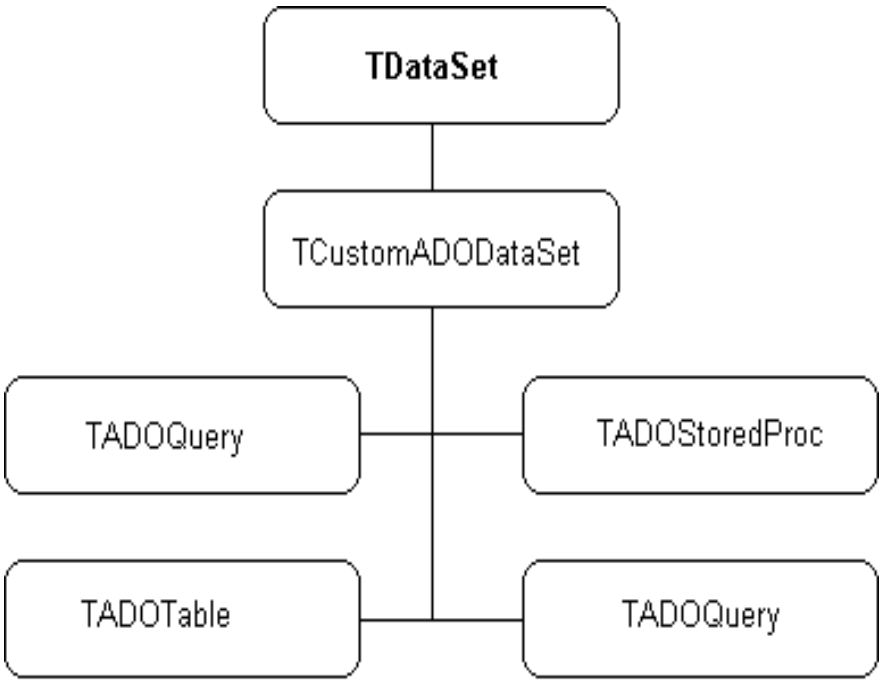

Рисунок 3.4 – Ієрархія класів наборів даних

Розглянемо ці компоненти:

TADODataSet–універсальний набір даних

TADOtable – таблиця БД;

TADOQuery – запит SQL;

TAoostoredProc–. процедура, що зберігається

1. Компонент TDataSet

Для компонентів, що інкапсулюють набір даних, їх спільним предком є клас TDataSet, що надає базові функції управління набором даних.

2. Компонент TADOTable

Компонент TADOTable забезпечує використання у додатках Delphi таблиць БД, підключених через провайдери OLE DB. За своїми функціональними можливостями та застосуванням він подібний до стандартного табличного компоненту.

Для компонентів, що інкапсулюють набір даних, їх спільним предком є клас TDataSet, що надає базові функції управління набором даних

3. Компонент TADOTable

Компонент TADOTable забезпечує використання у додатках Delphi таблиць БД, підключених через провайдери OLE DB. За своїми функціональними можливостями та застосуванням він подібний до стандартного табличного компоненту.

Ім'я таблиці БД задається властивістю

property TableName: WideString;

4. Компонент TADOQuery

Компонент TADOQuery забезпечує застосування запитів SQL під час роботи з даними через ADO. За своєю функціональністю він подібний до стандартного компонента запиту.

Текст запиту задається властивістю

property SQL: TStrings;

Параметри запиту визначаються властивістю

property Parameters: TParameters;

Число оброблених запитом записів повертає властивість

property RowsAffected: Intege

У роботі з компонентом TADOQuery використана мова запитів SQL.

3.3.3. Мова запитів - SQL

Мова SQL (Structured Query Language) - мова структурованих запитів, яка за своєю суттю орієнтована на доступ до даних, і її зазвичай включають до складу різних засобів розробки

Всі SQL-запити можна умовно розділити на два види:

1) статичний SQL-запит – включається до коду програми під час його розробки і не змінюється під час виконання програми.;

2) динамічний SQL-запит - створюється і змінюється в ході виконання програми.

Усі оператори та команди мови SQL можна розділити на три групи.

1. Оператори визначення даних – призначені для створення, видалення та зміни структури даних. Основні з них перераховані у таблиці 3.2.

Таблиця 3.2. Основні оператори визначення даних

Оператор	**Опис**
CREATE TABLE	Для створення таблиці бази даних
ALTER TABLE	Видаляє таблицю
DROP TABLE	Змінюють структуру таблиці
CREATE INDEX	Створює індекс
DROP INDEX	Видаляє індекс
CREATE VIEW	Створює уявлення
DROP VIEW	Видаляє уявлення

2. Оператори керування даними – призначені для керування привілеями доступу до даних. Основні оператори представлені у таблиці 3.3.

Таблиці 3.3-Основні оператори управління

Оператор	**Опис**
GRAND	Призначає привілеї користувачам
REVOKE	Видаляє привілеї користувачів

3. Оператори маніпулювання даними – призначені до роботи із записами таблиць. Основні оператори коротко описані у таблиці 3.4.

Таблиця 3.4 - Основні оператори маніпулювання даними

Оператор	**Опис**
SELECT	Для вибірки запусу по певним формату
UPDATE	Призначений для зміни та оновлення записів
INSERT	Вставляє нові записи до таблиці
DELETE	Видаляє записи з таблиці

Для обробки записів у базі даних виконуються такі основні процедури:

- Додавання індексу;
- Пошук індексу;
- Перегляд індексу;
- Редагування ключових слів;
- Очищення індексів.

Приклади процедур роботи з базою даних:

1.Додавання індексу.

При первинному індексуванні web-сторінки до бази даних заноситьсяінформація про її адресу та ключові слова. Рейтинг web-сторінки у своїй обнулюється, тобто. у поле rating таблиці Index заноситься значення "0".Процедура додавання індексу до бази індексів представлена на рисунку 3.5.

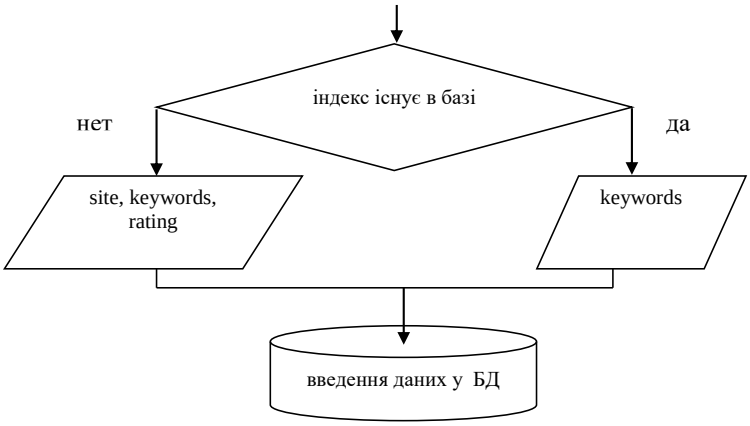

Рисунок 3.5.

Блок-схема процедури додавання індексу

2. Редагування ключових слів.

У разі редагування ключових слів відбувається зміна даних. Процедура редагування інформації про ключові слова представлена рисунку 3.6.

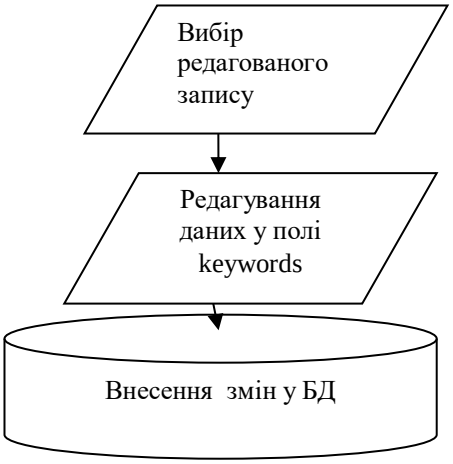

Рисунок 3.6. Блок-схема редагування ключових слів.

3.4. Пошуковий агент під Windows

Пошуковий агент призначений для пошуку інформації у пошуковій базі.. При завантаженні користувач бачить основне вікно (рис. 3.7)

Користувач може ввести від одного до п'яти слів запиту, які шукатимуться у документі. Нижче користувач може вибрати один з алгоритмів пошуку та оцінки релевантності:

- Подібність;
- Наближений;
- Частковий.

Щоб розпочати пошук, необхідно натиснути кнопку “Пошук”. Результати пошуку видаються у правій частині вікна програми у сітці. Щоб переглянути знайдений документ, необхідно двічі клацнути по ньому. Документ відкриється у новому вікні браузера.

Слід зазначити, що завантаження документа здійснюється так само, як і будь-якої іншої програми чи документа Windows. Усі функції із завантаження покладаються на браузер.

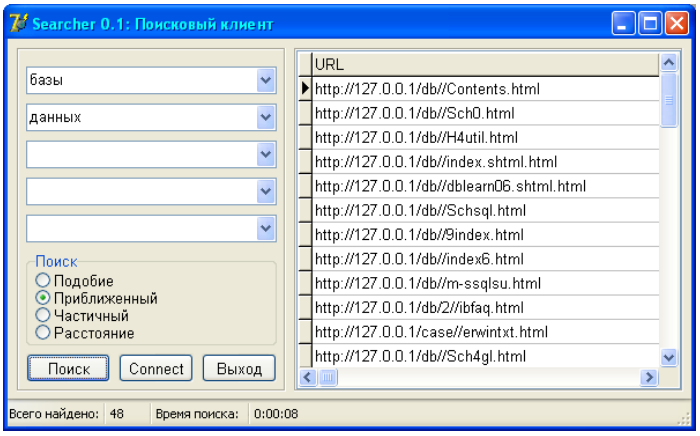

Рисунок 3.7 – Основне вікно пошукового клієнта

Для зручності користувачів у рядку стану видається загальна кількість знайдених документів та час пошуку на запит.

Для налаштування MySQL можна використовувати вікно "З'єднання" (кнопка Connect).

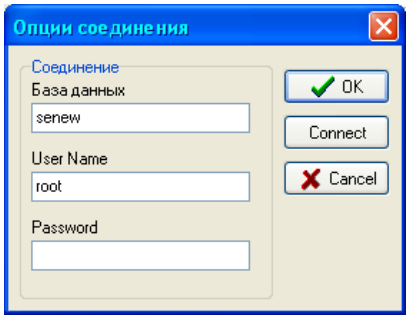

Рисунок 3.8 – Вікно налаштування з'єднання

У цьому вікні (рис3.8) необхідно вказати ім'я бази даних (за промовчанням SENEW), ім'я користувача (за промовчанням ROOT) та пароль (за промовчанням “”).

Пошуковий клієнт реалізований за допомогою Borland Delphi 7.
Програма складається з наступних модулів:

- вікно інтерфейсу (головне вікно) FormMain,
- вікно налаштування з'єднання FormConnect,
- модуль даних DataModuleSearcher.

3.5 . Розробка модуля данних DataModuleSearcher

Центральним модулем роботи з даними є модуль даних DataModuleSearcher, в якому розташовуються всі компоненти, відповідальні за зв'язок з пошуковою базою.

Для пошуку генерується SQL-запит до пошукової бази. Для з'єднання з MySQL використовується компонент TSQLConnection (вкладка dbExpress). Ім'я з'єднання встановлено у MySQLConnection. Як бібліотеки доступу до бази даних використовуються libmysql.dll і dbexpmysql.dll, які повинні також знаходитися на локальному комп'ютері.

Для виконання пошукового запиту (SQL-запиту) використовується зв'язування із чотирьох компонентів. Компонент TSQLQuery використовується для доступу до бази даних MySQL. Він пов'язаний із компонентом TSQLConnection. Цей компонент посилає запит до бази даних і отримує результат, але безпосередньо не бере участі в їх відображенні. Він є своєрідною проміжною ланкою між MySQL і програмою.

Зв'язок із цим компонентом здійснюється за допомогою TDataSetProvider (у властивості DataSet зазначений TSQLQuery). Доступ до даних, що відображаються, здійснюється за допомогою компонента TClientDataSet. У нього як ProviderName вказаний компонент TDataSetProvider. Встановлення з'єднання здійснюється шляхом

встановлення властивості Active компонента TClientDataSet у True. З цим компонентом пов'язаний також TDataSource.

Перевагою пошукового клієнта є те, що для його роботи не потрібні BDE, ODBC, ADO та інші додаткові засоби роботи з базами даних.

Вікно інтерфейсу крім функцій безпосередньо взаємодії з користувачем здійснює дві дуже важливі функції:

- генерацію запитів до пошукової бази (на основі мови SQL),
- визначення релевантності одержуваних документів.

Запити до пошукової бази формуються динамічно, в залежності від кількості слів для пошуку та вибраного алгоритму.

Визначення релевантності документів здійснюється шляхом обчислення сумарних ваг документів. Програма фактично транслює формальний опис алгоритмів на основі математичних формул у виразі реляційної алгебри та мови SQL.

Вагові коефіцієнти тегів зберігаються в базі даних MySQL спільно з пошуковою базою (таблиця TAGRATE). Таким чином, сама програма може перелаштовуватися на інші вагові коефіцієнти без будь-яких змін коду.

Коротко розглянемо структуру SQL-запитів, що формуються**.**

```
SELECT sitepages.URL, sitepages.IDPage, sum(tagrate.RATE) as Rating
FROM sitepages, tagrate , tag t , tag t1 , tag t2
WHERE
(t.Word LIKE "%слово1%") AND
((t1.IDPage = t.IDPage) AND
(t1.IDTAG = t.IDTAG) AND
(t1.Gruppa = t. Gruppa) AND
(t1.Word LIKE "% слово2%")) AND
((t2.IDPage = t.IDPage) AND
(t2.IDTAG = t.IDTAG) AND
(t2.Gruppa = t. Gruppa) AND
```

```
(t2.Word LIKE "% слово3%")) AND
(tagrate.TAG = t.IDTAG) AND
(sitepages.IDPAGE = t.IDPAGE)
GROUP BY t.IDPage
ORDER BY Rating DESC
```

Наведений запит відповідає алгоритму "Подібність".

Пошук (вибірка) здійснюється за допомогою виразу SELECT sitepages.URL, sitepages.IDPage, sum(tagrate.RATE) as Rating.

Вихідними даними є таблиця із трьох стовпців – адреси сторінки (sitepages.URL), ідентифікатора сторінки (sitepages.IDPage) та рейтингу (сумарної ваги) сторінки sum(tagrate.RATE) as Rating.

Рядок t.Word LIKE "%слово1%" забезпечує знаходження слова “слово1” у пошуковій базі, причому використання LIKE спільно з % дозволяє здійснювати частковий пошук (пошук у частині слова).

Рядок (t1.IDPage = t.IDPage) забезпечує знаходження слів на одній сторінці, рядок (t1.IDTAG = t.IDTAG) - тільки в однакових тегах,

рядок (t1.Gruppa = t. Gruppa) – всередині однієї речення в тезі. Пошук здійснюється у таблицях TAG.

Рядок (tagrate.TAG = t.IDTAG) забезпечує знаходження вагового коефіцієнта для тега з таблиці TAGRATE. За допомогою (sitepages.IDPAGE = t.IDPAGE) виконується знаходження адреси сторінки у таблиці SITEPAGES.

Вираз GROUP BY t.IDPage дозволяє здійснити групування знайдених слів на сторінках (IDPage – ідентифікатор знайденої сторінки).

ORDER BY Rating DESC забезпечує впорядкування результатів зменшення їх релевантності (точніше сумарної ваги, знайденої при пошуку). Подібним чином формуються запити для інших алгоритмів пошуку.

Треба зауважити, що використання лише стандартних засобів Delphi 7 дає можливість згенерувати пошуковий клієнт і для Linux (з використанням засобів Kylix).

3.6. Пошуковий агент для Інтернету

Агент пошуку через Інтернет-браузер забезпечує ті самі функції, що й пошуковий агент для Windows. Для роботи з клієнтом необхідний Інтернет браузер (наприклад, Interner Explorer) та з'єднання з сервером, де розташована пошукова база та сторінка інтерфейсу.

Пошуковий агент складається із сторінки інтерфейсу index.htm, яка містить форму (FORM) пошуку, та сторінки результату search.php. Сторінка результатів - мовою PHP 4. Для роботи програми потрібне також з'єднання з сервером MySQL.

Алгоритм роботи :

1. Оголошення змінних для зв'язку з MySQL.

Фрагмент модуля :

```
define("DBName","senew");
define("HostName","www.vanta.ru");
define("UserName","root");
define("Password","");
```

2. З'єднання з MySQL-сервером.

Воно відбувається за допомогою команди

mysql_connect(HostName,UserName,Password).

3. Формування тексту SQL-запиту та його виконання

$r=mysql(DBName,$sql).

Результат міститься в змінну $r.

4. Далі відбувається рядковий аналіз результату та його виведення в браузер.

```
$num = mysql_numrows($r);
for($i=0; $i<$num; $i++)
```

```
{ $f=mysql_fetch_array($r);
$url=mysql_result($r,$i,"url");
$rating=mysql_result($r,$i,"rating");
echo "<a href=\"$url\">$url</a>";
echo "$rating";}
```

3.7. Аналіз релевантності та вагових коефіцієнтів html -документу

Розрахуємо вагові коефіцієнти на прикладі html -документу з тегами тегів. Найбільш складним завданням є визначення вагових коефіцієнтів тэгів.

Складність полягає в наступному:

- немає чіткої методики визначення коефіцієнтів,
- для різних наборів даних коефіцієнти можуть змінюватися,
- великий набір параметрів, що варіюються.

Спочатку визначимо, які теги можуть брати участь у визначенні релевантності документів. Такими тегами будуть:

- Заголовок документа Title.
- Заголовки H1-H6.
- Ключові слова мета-тегу Keywords.
- Опис документа мета-тега Description.
- Підпис під малюнком Alt (конструкція тега Img).
- Текст документа.

Тепер необхідно вибрати критерій, за яким проводитиметься підбір вагових коефіцієнтів. Таким критерієм може бути впорядкованість документів щодо спадання їхньої релевантності. Іншими словами, найбільш релевантні документи мають знаходитися на початку списку, менш релевантні – наприкінці. Нехай при наборі вагових коефіцієнтів k виходить впорядкована множина документів, релевантних запиту R^k.

Нехай є безліч R , що визначає “ідеальне” розташування документів, тобто розташування документів, встановлене експертами предметної області

(людиною). Тоді необхідно знайти такий набір вагових коефіцієнтів, щоб множини R^k і були максимально R схожі.

Виразимо міру близькості множин R^k і R як функцію

$$B = \sum_i \left| r_i^r - r_i \right|, \qquad (3.1)$$

де : $r_i^k \in R^k$ - i-тий документ у множені R^k,

$r_i \in R$ - i-тий документ у множені R.

Підходящий набір вагових коефіцієнтів k буде отриманий при $B \to min$, тобто коли різниця позицій в "ідеальній" та отриманій множині документів буде мінімальна. Звичайно, набори вагових коефіцієнтів k можуть відрізнятися при різних запитах і для різних предметних областей, тому необхідно визначити якийсь загальний, усереднений набір. При розробки бази даних ,текстовий набір для запиту до «бази даних» із 50 документів. Тестовий набір було проаналізовано (проведено так звану експертну оцінку) та сформовано впорядковану множину, що відображає релевантність кожного документа. На початку множини знаходиться найбільш підходящий (релевантний) документ, наприкінці – найменш релевантний.

Розглянемо припущення – вага всіх тегів однакова. Релевантний документ - це документ, що має найбільшу кількість слів, що зустрічаються в тексті. Поставимо у відповідність кожному тегу вагу "1". Відповідно до (3.1.) визначимо міру близькості B=354. Отримане значення велике, наше припущення невдале.

Тепер припустимо, що теги мають різну вагу. Нехай тег TITLE має найбільше значення (вага 8), теги H1-H6 – ваги 7-2 відповідно. Нехай

Keywords і Description мають вагу 4, текст - вага 1, інші теги 2. При такому розташуванні ваги міра близькості B=130.

Таблиця 3.5. Визначення міри близькості за різних вагових коефіцієнтів

Тег	Вага	Вага	Вага
TITLE	1	8	20
H1	1	7	16
H2	1	6	14
H3	1	5	12
H4	1	4	10
H5	1	3	8
H6	1	2	6
Keywords	1	4	10
Description	1	4	10
Alt	1	2	5
Посилання Href	1	2	4
Area	1	2	5
Текст	1	1	1
$B = \sum_i \left\| r_i^r - r_i \right\|$	354	130	15

Варіюючи значеннями ваг, був отриманий набір, що дає міру близькості B=15.

Набір коефіцієнтів у цьому випадку було отримано наступний:

- Тег TITLE – 20,
- Тег H1 – 16,
- Тег H2 – 14,
- Тег H3 – 12,
- Тег H4 – 10,
- Тег H5 – 8,

- Тег H6 – 6,
- Мета-тег Keywords – 10,
- Мета-тег Description – 10,
- Конструкція Alt тега IMG – 5,
- Текст посилання Href – 4,
- Область Area – 5,
- Решта текст – 1.

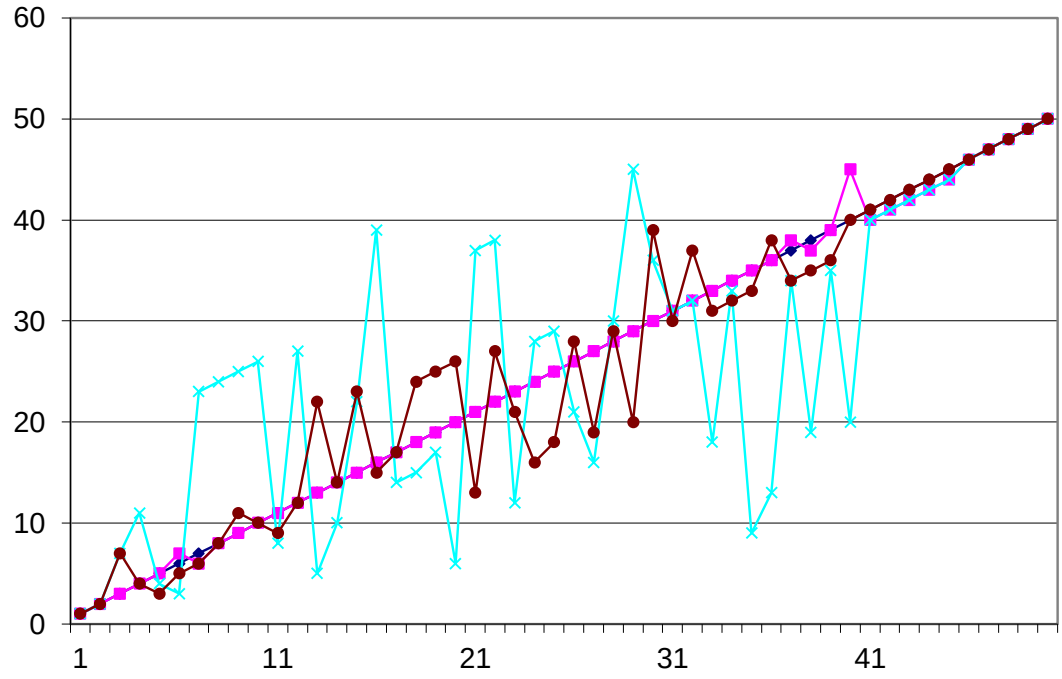

Рисунок .3.9 – Відхилення результатів від тестового набору.

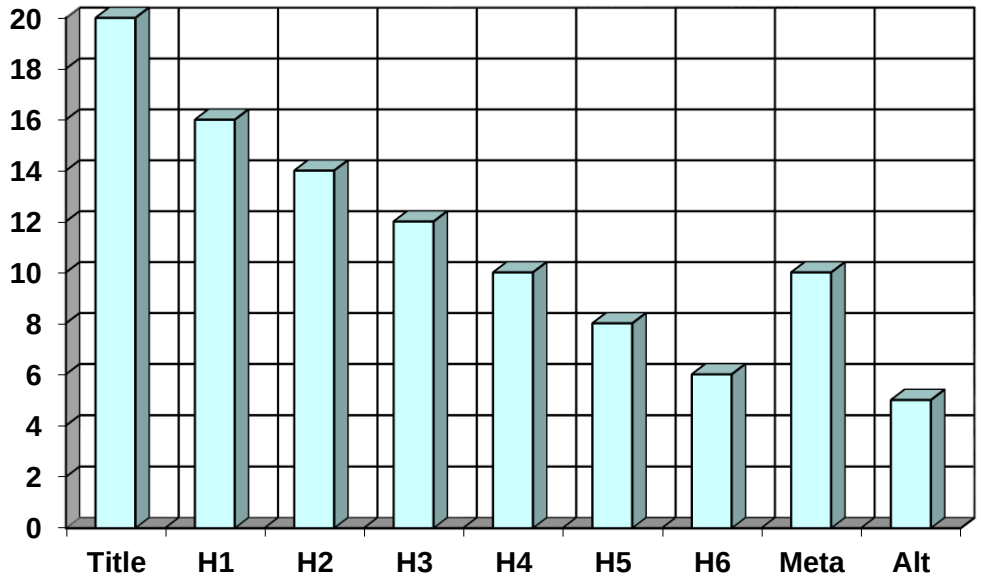

Рисунок 3.10 – Частка тегів щодо релевантності у розробленій системі

Отриманий набір перевірено на інших тестових наборах. Аналіз показав, що цей набір дає невеликі значення B всіх пошукових наборах (або запитів). Зміна ваги (наприклад, для досягнення B=0 для одного запиту) дає збільшення значення B для інших пошукових запитів. Тому вказаний набір можна вважати найбільш підходящим для різних запитів.

Висновки до третього розділ

В розділі розглянуто розробка структури та модулів пошукової системи, що поділяється на 3 класи: пошукові каталоги, повнотекстові пошукові системи та метапошукові системи.

Пошукові системи зазвичай складаються з трьох компонентів:

1) агент (павук або кроулер), який переміщається по Мережі та збирає інформацію;
2) база даних, що містить всю інформацію, що збирається павуками;
3) пошуковий механізм, який використовують як інтерфейс взаємодії з базою даних .

Показана структура та модулів пошукової системи, структура програми, опис бази даних, опис процедур, SQL - мова структурованих запитів.

Розроблена блок-схема процедури додавання індексу до бази індексів та блок-схема процедури редагування ключових слів.

Розглянуто пошуковий агент під Windows - центральним модулем роботи з даними є модуль даних DataModuleSearcherm , а пошуковий клієнт реалізований за допомогою Borland Delphi 7. Показан алгоритм роботи пошукового агента для Інтернету та фрагменти програм в мові PHP. Проведено аналіз релевантності та вагових коефіцієнтів html –документу.

ВИСНОВКИ

Інтернет продовжує розвиватися з неослабною інтенсивністю, по суті стираючи обмеження на поширення та отримання інформації у світі. Однак у цьому інформаційному океані буває нелегко знайти необхідний документ. Шляхом вирішення цієї проблеми виступають інформаційно-пошукові системи, які організовують пошук у Всесвітній глобальній мережі.

Пошукові системи вже давно стали невід'ємною частиною Інтернету. Пошукові системи зараз - це величезні і складні механізми, що є не тільки інструментом пошуку інформації, але й привабливими сферами для бізнесу.

В кваліфікаційній магістерський роботі розглянуто докладний механізм пошуку інформації у Всесвітній Глобальній мережі та проведено огляд найбільш популярних всесвітніх та українських інформаційно-пошукових систем. Розроблено алгоритми пошуку інформації та оцінки релевантності знайдених документів. Дія алгоритмів полягає в обліку різних конструктивних особливостей документів – насамперед тегов. Кожному виду тегів було поставлено у відповідність певну вагу, яка використовується надалі визначення релевантності документів. Були проведені дослідження щодо визначення цих вагових коефіцієнтів, у ході яких було визначено набір, що дає найкращу якість пошуку для більшості запитів. Кожен алгоритм був досліджений на показники точності та повноти. "Частковий" алгоритм має найвищі показники за цими критеріями - в переважній кількості випадків обидва показники становлять 100%. Алгоритм "подібність" дозволяє знаходити документи з високим рівнем релевантності. Показана структура та модулів пошукової системи, структура програми, опис бази даних, опис процедур, SQL - мова структурованих запитів.

Розроблена блок-схема процедури додавання індексу до бази індексів та блок-схема процедури редагування ключових слів.

Розглянуто пошуковий агент під Windows - центральним модулем роботи з даними є модуль даних DataModuleSearcherm , а пошуковий клієнт

реалізований за допомогою Borland Delphi 7. Показан алгоритм роботи пошукового агента для Інтернету та фрагменти програм в мові PHP. Проведено аналіз релевантності та вагових коефіцієнтів html –документу

Розроблена система може бути корисною як для досліджень у галузі пошуку документів та оцінки релевантності, так і для промислового використання

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Астістова Т. І., Розробка інформаційно-пошукової системи для автоматизованого збору даних / Т. І. Астістова, Б. Г. Множинський // Інформаційні технології в науці, виробництві та підприємництві : збірник наукових праць молодих вчених, аспірантів, магістрів кафедри комп'ютерних наук та технологій / за заг. наук. ред. В. Ю. Щербаня. – Київ : ТОВ "Фастбінд Україна", 2023. – С. 108-111.
   https://er.knutd.edu.ua/handle/123456789/24122
2. Астістова Т. І. Seo – оптимізація в системі моніторингу web – ресурсів// Т.І. Астістова // Технології та інжиніринг («Вісник КНУТД. Серія Технічні науки») Київський національний університет технологій та дизайну, Україна No1(12), 2023 р. С. 9-17. ID: 6506601603. https://vistnuk.knutd.edu.ua/wpcontent/uploads/sites/2/2023/05/1-1- 2023.pdf
3. Аносов А. Критерії вибору СУБД при створенні інформаційних систем [Електронний ресурс]. – Режим доступа: http://www.google.ua (опубліковано у 2016р.)
4. Методи представлення, збереження та аналізу даних інформаційних систем, колективна монографія / В. Ю. Щербань, С. М. Краснитський, Т. І. Астістова, В. М. Яхно. – Київ-2023р. : ТОВ "Фастбінд Україна
   https://er.knutd.edu.ua/handle/123456789/24076
5. Вертузаєв М. С. та ін. Безпека комп'ютерних систем: злочинність у сфері комп'ютерної інформації та її попередження/ Запоріжжя , 1998;
6. Маркарова Т. Отраслевий тезаурус інфрмаційно- пошукової системи// Наук. і тех. б-ка. /Київ, 2002. № 5;
7. Кулік О. Автоматизована каталогізація: досвід, проблеми, перспективи // Библиотеч. форум Украины. 2003. № 2;

8. Яковлєва Ю. Оцінка інформативності документів як напрям інтелектуалізації пошукових систем // БВ. 2018. № 1.
9. Популярні пошукові системи в світі: цікаві факти [Електронний ресурс] Режим доступу до ресурсу: https://project-seo.net/uk/blog-uk/rejtyng-poshukovyh-system-2018-2019/
10. Пошукові системи [Електронний ресурс] . Режим доступу до ресурсу: https://ube.nlu.org.ua/article/%D0%86%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%BE-%D0%BF%D0%BE%D1%88%D1%83%D0%BA%D0%BE%D0%B2%D0%B0%20%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0
11. Сучасні інформаційно- пошукові системи [Електронний ресурс] , Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/%D0%95%D0%BD%D1%86%D0%B8%D0%BA%D0%BB%D0%BE%D0%BF%D0%B5%D0%B4%D1%96%D1%8F_%D1%81%D1%83%D1%87%D0%B0%D1%81%D0%BD%D0%BE%D1%97_%D0%A3%D0%BA%D1%80%D0%B0%D1%97%D0%BD%D0%B8
12. Інформація та документація. Бібліотечно-інформаційна діяльність. Терміни та визначення понять: ДСТУ 7448:2013. — Київ : Мінекономрозвитку України, 2014. — III, 41 с. — (Національний стандарт України) — Зі скасуванням в Україні ГОСТ 7.26–80 — Текст укр., рос., англ., фр.
13. Великий інформаційний смітник [укр.]: [Електроннийресурс] [арх.28 січня 2021 року]/Олена Голуб// day.kyiv.ua: газета.—День, 2020.— 23 грудня.— Дата звернення: 28 січня 2021 року.

14. Zhelezniak, Mykola (2019).Encyclopedia of Modern Ukraine in the challenges oftoday [Електроннийресурс], Режим доступу до ресурсу :https://web.archive.org/web/20210127034837/http:/evu.encyclopedia.kyiv.ua/en/vol-11/encyclopedia-of-modern-ukraine-in-the-challenges-of-today/*The Encyclopedia Herald of Ukraine* (англ.). doi:10.37068/evu.11.1. Архів оригіналу за 27 січня 2021. Процитовано 23 грудня 2020.
15. Бойко Юлія. (2016), Енциклопедія Сучасної України— основний науковий проєкт Інституту енциклопедичних досліджень НАН України. Режим доступу до ресурсу*Гілея: науковий вісник*, *111*, 96–99.
16. *Іщенко Олександр,*Енциклопедія Сучасної України» vs. «Вікіпедія»: рівень популярності в українському інформаційному просторі// [Електронний ресурс] Режим доступу до ресурсу:*https://evu.encyclopedia.kyiv.ua/volume-11/entsyklopediia-suchasnoi-ukrainy-vs-vikipediia-riven-* Енциклопедичний вісник України.*— 2019.—№ 11.—С. 23–30.*Архівовано *з джерела 24 вересня 2021.*
17. Інформаційно-пошукові системи / В. В. Тарасюк, Р. П. Судик // Енциклопедія Сучасної України [Електронний ресурс] Режим доступу: https://esu.com.ua/article-12483 / Редкол.: І. М. Дзюба, А. І. Жуковський, М. Г. Железняк [та ін.] ; НАН України, НТШ. К. : Інститут енциклопедичних досліджень НАН України, 2011.
18. Akopkokhyants S. Mastering Dart // Sergey Akopkokhyants, 2014. – 346 c.
19. Bracha G. The Dart Programming Language // Gilad Bracha, 2015. – 224 c.
20. Belchin M. Web Programming with Dart // M. Belchin, P. Juberias, 2016. – 472 c
21. Кожичкін Євген, Пошукова система, заснована на нейронній мережі. [Електронний ресурс]. — Режим доступу: http://miem.edu.ua/
22.. Капустін В.А. Основи пошуку інформації в Інтернеті / Методичний посібник[Електронний ресурс]. – Режим доступу:.

http://www.edc.samara. ua /gr/basics.htm

23..Талантов Михаїл, Професійний пошук в Інтернеті: повнота, достовірність, швидкість. [Електронний ресурс].– Режим доступу:http://nur.yamal.ua/internet/search/prof_search01.shtml

24. Остапенко, В. П. Перспективні напрямки вдосконалення інформаційних систем обліку [Електронний ресурс] Режим доступу : http://www.kpi.kharkov.ua/archive/microcad/2013/s25.pdf.

25. Попадюк, С. В. Інформаційні технології та системи в обліку [Електронний ресурс] Режим доступу : http://repository.vsau.org/getfile/1497.pdf

26.. Міністерство ідей. [Електронний ресурс] // [сайт]. – Режим доступу: http://blog.mid.ua/2012/04/rfid.html.

27. Демківська Т.І. Розробка інформаційно- пошукової системи підприємства з використаніям концепції MVC// Т.І Демківська, О.М. Адвена//Інформаційні технології в науці, виробництві та підприємництві: зб. наук. праць молодих вчених, аспірантів, магістрів кафедри інформаційних технологій проектуванн. – К. : КНУТД, 2018. – С. 223– 226. – ISBN 978-966

28. Грошев А. С. Основи роботи з базами даних [Электроний ресурс] / А. С. Грошев. –Режим доступа : www.intuit.ua/department/database/basedb

29. Все про бази даних, системах управління базами даних (СУБД), мова SQL [Електронний ресурс] , режим доступа : http://www.sql-home.org.ua.

30. Маннінг К. Д., Введеня в інформаційний пошук./ П. Рагхаван., Х М. Шютце / Львів: Вільямс, 2011

31. Ташков П.А. Веб-мастерінг. HTML, CSS, JavaScript, PHP, CMS, AJAX, , Київ- 2010 – 512 с.

32. MySQL: особливості та сфери застосування. [Електронний ресурс] Режим доступа: https://www.bytemag./articles/detail.php?ID=6547

33. Комер Д. Принципи функціювання Інтернет: Пер. с англ./ Д. Комер.//

Львів ; Харьков; 2002.–379 с.

34. Hershel Harris, Bert Nicol. SQL/DS: IBM's First RDBMS // IEEE Annals of the History of Computing, Volume 35, Number 2, April–June 2018, [Електронний ресурс].– Режим доступу https://muse.jhu.edu/article/522037
35. Пошукова _система [Електронний ресурс].– Режим доступу: www.ua.wikipedia.org/wiki/
36. Попадюк, С. В. Інформаційні технології та системи в обліку [Електронний ресурс]. Режим доступу http://repository.vsau.org/getfile/1497.pdf
37. Ситник В. Ф. Основи інформаційних систем: Навч. посібник. Вид. 2-ге, перероб. і доп. / В. Ф. Ситник, Т. А. Писаревська, Н. В. Єрьоміна, О. С. Кра- єва; За ред. В. Ф. Ситника.// К.: КНЕУ, 2001. – 420 с.
38. Організація баз даних: практичний курс: Навч. посіб. для студ. / А. Ю. Берко, О. М. Верес; Нац. ун-т «Львівська політехніка» - 2003.
39. Резніченко, В.А. (2021). 60 років базам даних. «Проблеми програмування» (№ 3) - 2022 р.
40. Silberschatz, Abraham; Sudarshan, S. (2011). Database system concepts (вид. 6). New York: McGraw-Hill – 2011 р.
41. Копанєва В. О. Формати опису мережевих інформаційних ресурсів. Інформаційна діяльність: Проблеми науки, освіти, практики, Київ, 17-19 травня 2017 р.
42. Керівництво для настільних персональних комп'ютерів [Електронний ресурс] - Режим доступу: https://docs.microsoft.com/enus/dotnet/desktop/wpf/overview/?view=netdesktop-5.0.
43. Гуманітарний портал [Електронний ресурс] - Режим доступу https://gtmarket/concepts/7091 – 2016 р.
44. Іцик Бен - Ган - Microsoft SQL Server 2008. Основи T-SQL.
45. Карвіш Б.К 21 Програмування баз даних SQL. Типові помилки та їх усунення Б. Карвін. - М: Рід Груп, 2016 р.

46. Кравець П.о. К 77 об'єктно - орієнтоване програмування: навч. посібник / П.о. Кравець. - Львів: Видавництво Львівської політехніки, 2018.
47. DataSet Class [Електронний ресурс] Режим доступу: https://docs.microsoft.com/en-us/dotnet/api/system.data.dataset?view=net-5.0.
48. DataSets, DataTables, and DataViews [Електронний ресурс] Режим доступу:https://docs.microsoft.com/enus/dotnet/framework/data/adonet/dataset-datatable-dataview/.
49. DataSet и DataTable [Електронний ресурс] Режим доступу: https://metanit.com/sharp/adonet/3.6.php.
50. ADO.NET | DataSet - Professor Web [Електронний ресурс] Режим доступу: https://professorweb.ru/my/ADO_NET/base/level2/2_1.php.
51. Бази данних і мова SQL [Електронний ресурс] Режим доступу: https://function-x.ru/sql_join.html.
52. Електронний ресурс – Режим доступу: https://www-terraform-io.translate.goog/docs/language/providers/index.html?_x_tr_sl=en&_x_tr_tl=uk&_x_tr_hl=ru
53. MongoDB.[Електронний ресурс]-Режим доступ : https://proselyte.net/tutorials/mongodb/advantages
54. Грицюк Ю. І., Аналіз вимог до програмного забезпечення. Навчальний посібник/ Ю. І. Грицюк .// Київ,-2018, С.425
55. Майкл Віттіг. «Amazon Web Services in Action ». США, Manning, 1237 – 1469 с.
56. Майєрс, Г. Надійність програмного забезпечення/Г. Майєрс. – К.: Світ, 2018. – 360 с.
57. Інформаційні технології у суспільстві [Електронний ресурс] – Режим доступу:https://sites.google.com/site/informacijnesuspilstvo26/informacijni-tehnologiie-u-suspilstvi – Назва з екрана
58. Г.Г.Швачич Сучасні інформаційно-комунікаційні технології:

навчальний посібник / Г.Г.Швачич, В.В.Толстой, Л.М.Петречук, Ю.С.Іващенко, О.А.Гуляєва, О.В. Соболенко - Дніпро: НМетАУ, 2017. –230 с. 76. 16.

59. Астістова Т.І. Аналіз програм та сервісів для поширення сайтів в інтернет- просторі / Т. І Астістова // Інформаційні технології в науці, виробництві та підприємництві: зб. наук. праць молодих вчених, аспірантів, магістрів кафедри інформаційних технологій проектування. – К. : Освіта України, 2020. – С170-175
60. Грицюк Ю. І., Аналіз вимог до програмного забезпечення. Навчальний посібник/ Ю. І. Грицюк .// К-,2018, С.425