Хмельницький національний університет

Факультет програмування

та комп'ютерних і телекомунікаційних систем

Кафедра телекомунікацій, медійних та інтелектуальних технологій

ДИПЛОМНА РОБОТА МАГІСТРА

Метод підвищення пертинентності результату пошуку за рахунок вдосконалення

Назва теми

алгоритму ранжування та індексації сайтів

Галузь знань 11 – Математика та статистика

Спеціальність 113 – Прикладна математика

Шифр ДРПМ. 170165.01.22.00

Виконав: студент 2 курсу, група ПМм-19-1 _____ Мурах Б.Р.
Підпис Ініціали, прізвище

Керівник _____ к.т.н., доц. Муляр І.В.
Підпис, дата Ініціали, прізвище

Нормоконтролер _____ _____
Підпис, дата Ініціали, прізвище

До захисту допускаю:

Зав. кафедри ТМІТ _____ д.т.н., проф. Підченко С.К.
Підпис, дата Ініціали, прізвище

9 12 2020 р.

Хмельницький, 2020

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Програмування та комп'ютерних і телекомунікаційних систем

Кафедра телекомунікацій, медійних та інтелектуальних технологій

Освітній рівень магістр

Галузь знань 11 Математика та статистика

Спеціальність 113 Прикладна математика

Освітня програма освітньо-професійна програма підготовки магістра

ЗАТВЕРДЖУЮ

Зав. кафедри С.К. Підченко

"3" 09 2020 р.

ЗАВДАННЯ
НА ДИПЛОМНУ РОБОТУ

Мурах Б Р .

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Метод підвищення пертинентності результату пошуку за рахунок вдосконалення алгоритму ранжування та індексації сайтів

2. Керівник проекту (роботи) к.т.н., доц. Муляр І.В

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.09.2020 р. № 118

2. Строк подання студентом проекту (роботи) на кафедру 01.12.2020

3. Вихідні дані до проекту (роботи) принципи дії та функціонування алгоритму ранжування Google – PageRank

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) Аналіз існуючих методів та алгоритмів пошуку інформації. Побудова математичної моделі алгоритму ранжування. Опис модифікованого методу ранжування. Симуляція розробленого методу

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) Тема, мета магістерської роботи, об'єкт дослідження, предмет дослідження, задачі дослідження, наукова новизна, практична цінність, апробація роботи, публікації. Типова структура пошукової системи для WWW. Принцип функціонування пошукової системи Google. Загальна модель яка описує функціонування алгоритму PageRank та формування рейтингу сайтів. Матрична модель модифікованого алгоритму в мережі з імовірностями переходів. Модифікований алгоритм та формула розрахунку ModPageRank. Порівняльна оцінка видачі результатів до модифікації та після. Науковий результат. Висновки.

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «__» _____ 20__ р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики ДРМ з керівником	2.02.2020	
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	2.03.2020	
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	1.04.2020	
4	Робота над розділом 2 – розробка моделей і методів для вирішення поставленої задачі	1.05.2020	
5	Робота над науковою статтею	1.06.2020	
6	Робота над розділом 3 – розробка алгоритмів та технологій, їх аналіз	1.09.2020	
7	Робота над розділом 4 – моделювання процесу, для вирішення поставленої задачі	1.10.2020	
8	Узгодження отриманих; оформлення пояснювальної записки згідно вимог	1.11.2020	
9	Оформлення графічної частини	11.11.2020	
10	Попередній захист ДР	15.11.2020	
11	Захист ДР на засіданні ЕК	12.12.2020	

Студент _____ Підпис _____ Ініціали, прізвище

Керівник проекту (роботи) _____ Підпис _____ Ініціали, прізвище

АНОТАЦІЯ

Тема дипломної роботи: Метод підвищення пертинентності результату пошуку за рахунок вдосконалення алгоритму ранжування та індексації сайтів

Автор роботи: Мурах Богдан Ростиславовоч

Керівник роботи: к.т.н., доц. Муляр Ігор Володимирович

Загальний обсяг роботи: 81 сторінка, 29 рисунків, 2 додатки, 36 посилань,

PAGERANK, GOOGLE, ПОШУКОВІ СИСТЕМИ, СОЦІАЛЬНІ МЕРЕЖІ

Метою дипломної роботи є підвищення пертинентності результатів пошуку за рахунок модифікації алгоритму ранжування Google – PageRank

Дана дипломна робота присвячена розробці метод виявлення інформаційного впливу соціальних мереж на індексацію сторінки та формування переліку зовнішніх гіперпосилань, дя зміни коефінієнтів ранжування відповідних сторінок. Модифікований алгоритм формування рейтингу вебресурсів доцільно використати для пошуку інформації, яка найбільш часто обговорюється на форумах та в соціальних групах

ANNOTATION

a master's degree work of Murakh Bohdan

entitled «The method of increasing the pertinence of the search result by improving the algorithm of ranking and indexing of sites».

Mentor: Ihor Muliar

Total volume of work: 81 pages, 29 figures,,2 appendices, 36 references.

PAGERANK, GOOGLE, SEARCH ENGINES, SOCIAL NETWORKS

The purpose of the thesis is to increase the pertinence of search results by modifying the Google ranking algorithm - PageRank

This thesis is devoted to the development of methods for identifying the informational impact of social networks on page indexing and the formation of a list of external hyperlinks, to change the ranking coefficients of the respective pages. The modified algorithm of formation of a rating of web resources should be used for search of the information which is most often discussed on forums and in social groups.

Дата / Date 09.12 Підпис студента / Signature

ЗМІСТ

Вступ 5
1 Аналіз існуючих підходів до пошуку інформації..... 10
1.1 Типова структура пошукової машини 10
1.2 Принципи функціонування пошукових систем…… 17
1.3 Аналіз моделей інформаційного пошуку..... 24
1.4 Принципи функціонування алгоритму пошуку в Google……... 31
1.5 Постановка задачі ……..……..……..……..... 34
2 Математична модель методу ModPR…….……..……..……..…….. 36
2.1 Математична модель, яка використовується в алгоритмах пошуку Google....…….……..……..……..…… 36
2.2 Вдосконалена математична модель ранжування сторінок……..… 45
2.3 Висновки ……..……..… 49
3 Метод підвищення пертинентності результату 50
3.1 Вдосконалений метод індексації ресурсів 50
3.2 Модифікований метод пошуку інформації ModPR 55
3.3 Висновки ……..……..…… 59
4 Застосування модифікованого методу пошуку MODPR..... 61
4.1 Алгоритми пошуку спільнот у соціальних мережах 61
4.2 Проєктування програмного продукту…….…… 66
4.3 Оцінка впливу груп в соціальних мереж на пертинентність результатів пошук……..…… 73
4.3 Висновки ……..…… 75
Висновки 77
Перелік джерел посилань 79
Додаток А Фрагмент програмного коду алгоритму …… 83
Додаток Б Копії наукових праць 86
Додаток В Презентація 92

ВСТУП

Всесвітня павутина (WWW) [11] за останні два десятиліття помітно збільшилася в розмірах. Кількість інформації та ресурсів, доступних сьогодні на WWW, зросла в геометричній прогресії, і майже будь-яка інформація присутня, якщо користувач виглядає досить довго. Google [24] припускає, що він проіндексував понад вісім мільярдів вебсторінок, і це може бути часткою всіх доступних вебсторінок. Популярність WWW можна віднести головним чином до єдиного методу доступу, який він надає різним інформаційним службам, та його підтримці гіпермедіа, яка пов'язує широкий спектр мультимедійних даних, фізично розподілених по всьому світу, в єдину гігантську віртуальну базу даних. Він забезпечує потужний засіб для поширення інформації на будь-яку тему майже для будь-якого користувача в Інтернеті. Дедалі більше інформації стає доступною в режимі он-лайн через WWW - від останніх новин до наукових звітів та супутникових знімків. Однак цей інформаційний вибух призвів до неминучої проблеми пошуку ресурсів.

Щоб знайти відповідні сторінки, користувач повинен переглядати багато веб-сайтів WWW, які можуть містити інформацію. користувачі можуть або переглядати сторінки через точки входу, такі як популярні портали, Yahoo, MSN та AOL, або використовувати пошукову систему для пошуку конкретної інформації. Початок пошуку з однієї з точок входу не завжди є найкращим підходом, оскільки немає спеціально організованої структури для WWW, і не всі сторінки доступні для інших. У разі використання пошукової машини користувач подає запит, як правило, список ключових слів, а пошукова система повертає список вебсторінок, які можуть бути релевантними відповідно до ключових слів. Для цього пошукова система повинна здійснити пошук у вже існуючому індексі всіх вебсторінок на відповідні. Пошукові системи постійно беруть участь у скануванні через WWW з метою індексації. Коли користувач подає ключові слова для пошуку, пошукова система відбирає та ранжує документи за своїм індексом у порядку зменшення релевантності та корисності щодо ключових слів.

Технологія нечіткого пошуку інформації дозволяє розширити запит за допомогою подібних орфографічних слів, що містяться в архіві документів, наприклад в електронній бібліотеці.

Оригінальний алгоритм здатний знаходити всі лексикографічно подібні слова, що відрізняються замінами, пропусками, вставками символів тощо.

Нечіткий пошук слід застосовувати при пошуку слів із друкарськими помилками, а також у випадках, коли є сумніви щодо правильності написання назви, назви організації тощо. Унікальні алгоритми, що реалізують нечіткий пошук, базуються на принципах асоціативного доступу до словосполучень, що містяться в текстовому покажчику повнотекстового архівного сховища документів.

Для прискорення пошуку створюється спеціальний покажчик, який містить фрагменти словосполучень із посиланням на слова, в яких ці фрагменти трапляються. Алгоритм пошуку дозволяє швидко виділити всі слова, фрагменти яких відповідають фрагментам слова в запиті. Вказавши розмір (відсоток фрагментів, фрагменти, що відрізняються, і допустиме позиційне розміщення їх у слові), ви можете легко відрегулювати точність і повноту пошуку - підбирайте слова, наближені до запиту.

Сьогодні зусилля багатьох найбільших дослідницьких організацій і фондів зосереджені саме на проектах подання, підтримки та використання інформації в Internet. Підтвердженням цього можуть бути дослідження International Institute for Electronic Libraries Researches, проекти eLib, DeLIver, діяльність бібліотеки конгресу США, SIGIR (Special Interest Group on Information Retrieval) - цикл конференцій, що проводяться ACM (ACM - Association of Computing Machinery) SIGIR - міжнародною групою спеціалістів з інформаційного пошуку, найбільших російських і закордонних фондів: Фонд Сороса (програми «Інтернет», «Автоматизація бібліотек»), International Science Foundation (програма Digital Library Initiative) [28, 39]. Високий авторитет вищезгаданих конференцій та участь в них провідних дослідницьких колективів, а також розробників технологій

інформаційного пошуку визначає пріоритетні напрямки і загальні принципи розвитку пошукових систем.

Перспективним способом інтелектуалізації сучасних систем обробки тексту є використання семантики. Поширені методи повнотекстового пошуку (С. Баклі, С. Робертсон, Г. Солтон,) та Інтернету (С. Брін, А. Бродер, Л. Пейдж) здійснюють пошук безпосередньо за словами-запитами або вимагають використання дорогих лінгвістичних ресурсів - тезауруси, онтології, побудовані людьми (К. Філлмор, Г. Міллер, І. В. Замаруєва, С. Ніренбург) [7, 27].

Завдання ранжирування документів, згідно з деякими заздалегідь визначеними критеріями, підпадає під відповідальність алгоритмів ранжирування. Алгоритм ранжирування є однією з найважливіших складових будь-якої пошукової системи і зазвичай вимагає великої уваги під час розробки двигуна. Різні пошукові системи використовують різні класи алгоритмів ранжування з різним ступенем ефективності та ефективності. Інтуїтивно зрозуміло, що хороша система пошуку інформації повинна представляти відповідні документи вище за рейтингом, а менш релевантні документи слід за ними. Незважаючи на те, що алгоритми ранжирування, слідуючи процесу пошуку, прагнуть до досягнення цієї мети, часто зустрічається багато нерелевантного серед відповідної запитуваної інформації. Цей неоптимальний результат призвів до кількох досліджень у галузі алгоритмів ранжування пошукових систем.

Віртуальні групи - це новий тип спільнот, що виникають та функціонують в електронному просторі (насамперед через Інтернет) з метою вирішення своїх професійних, політичних проблем, задоволення їх потреб у мистецтві, дозвіллі тощо [32].

Для дослідження механізмів інформаційного впливу на суспільство через інтернет простір у цій роботі використовуються методи мультиагентного моделювання, які найчастіше використовується для аналізу складних систем, в яких важко формалізувати всі процеси в аналітичній формі. Створення та аналіз сучасних багатоагентних моделей інформаційних впливів і, відповідно, інформаційного простору, передбачає застосування та подальший розвиток

інформаційних технологій, які повинні забезпечити функціонування та аналіз складних багатоелементних систем.

У цій роботі проведено дослідження основних алгоритмів пошуку інформації в пошукових системах, їх аналіз та порівняння. Автор пропонує алгоритм ранжирування та індексації веб-сайтів, який покликаний підвищити ефективність пошуку інформації з урахуванням їх популярності в соціальних мережах та форумах.

Актуальність роботи полягає у підвищені результатів пошуку інформації в Інтернеті потребам користувача. Пошук відбувається із врахуванням до формування рейтингу сайту соціальних спільнот та вебфорумів.

Науковою гіпотезою є припущення, що існує можливість врахувати рейтинг ресурсів в соціальних спільнотах в індексування сайту пошуковим алгоритмом.

Мета магістерської роботи полягає в підвищені пертинентності результатів пошуку за рахунок модифікації алгоритму ранжування Google – PageRank.

Для досягнення поставленої мети вирішено такі задачі:

1. Проаналізовано та досліджено принципи пошуку інформації в Google.
2. Розглянуто математичину модель існуючого алгоритму та вдосконалено її.
3. Вдосконалено алгоритм ранжування сайтів пошукової системи Google, за рахунок враховування популярності сторінки в тематичних соціальних спільнотах.
4. Розроблено метод виявлення інформаційного впливу соціальних мереж на індексацію сторінки та формування переліку зовнішніх гіперпосилань, для зміни коефіцієнтів ранжування відповідних сторінок
5. Проведено дослідження розробленого методу, та оцінено його ефективність

Об'єктом дослідження: є процес пошуку та індексації сайтів для формування їх рейтингу пошуковою системою Google.

Предметом дослідження**:** є методи і алгоритми роботи пошукових систем.

Наукова новизна результатів магістерської роботи:

1. Вдосконалено алгоритм ранжування сайтів пошукової системи Google, який враховує популярність сторінки в тематичних соціальних спільнотах.

2. Розроблено метод виявлення інформаційного впливу соціальних мереж на індексацію сторінки та формування переліку зовнішніх гіперпосилань, для зміни коефіцієнтів ранжування відповідних сторінок.

Практична цінність. Цей алгоритм підніме маловідомі корисні джерела на вищі позиції, конкретизує та звужує пошук до обсягу теми, щодо якої було зроблено запит. Модифікований алгоритм доцільно використати для пошуку інформації яка найбільш часто обговорюється на форумах та в соціальних групах.

Публікації. За матеріалами магістерської роботи опубліковано 1 стаття у нефаховому журналі та 1 теза доповіді на міжнародній конференції.

1 АНАЛІЗ ІСНУЮЧИХ ПІДХОДІВ ДО ПОШУКУ ІНФОРМАЦІЇ

1.1 Типова структура пошукової системи

Актуальність досліджень у галузі інформаційного пошуку також обумовлена тим, що при пошуку інформації в мережі Інтернет множина документів, які є результатом відповіді на запит, як правило, виходить дуже великою за рахунок величезного числа «шумових» документів. Це обумовлює необхідність підвищення якості методів інформаційного пошуку. Для порівняння ефективності різних методів необхідно визначити, які критерії будуть використані при оцінці ефективності. Звичайно, обчислювальна продуктивність методу є одним із критеріїв оцінки ефективності, але набагато більш важливими показниками зазвичай є критерії, що характеризують якість результатів пошуку. До таких показників найчастіше відносять два параметри:

1) точність (precision) - частка релевантного матеріалу у відповіді на запит пошукової системи;

2) повнота (recall) - частка знайдених релевантних документів у загальній кількості релевантних документів колекції [36].

Проблема полягає в тому, що будь-який пошуковий сервіс - універсальний, розрахований на роботу з усіма користувачами, без врахування їх індивідуальних потреб.

Існуючі пошукові засоби, спрямовані на пошук інформації в Інтернет, діляться на два класи [9, 18]: пошукові каталоги і пошукові машини, що розрізняються між собою як структурою, так і призначенням. Крім того, виділяють клас автономних пошукових агентів, які з'явилися і досить успішно використовуються тільки завдяки існуванню великої кількості пошукових каталогів і пошукових машин.

Пошукові каталоги представляють собою системи, в яких зв'язки між інформаційними об’єктами організовуються при безпосередній участі людини. Інформаційні об'єкти, під якими маються на увазі інформаційні ресурси Інтернет,

класифіковані в каталогах за тематичними ознаками. Можливість знаходження релевантної інформації за допомогою пошукових каталогів залежить від компетентності редакторів каталогу в разі ручної організації інформації, її структурування та попереднього наповнення тематичного каталогу, а у разі автоматичної класифікації документів - від використовуваних методів автоматичної класифікації. Ручний підбір документів для таких каталогів, до якого залучаються досвідчені експерти з конкретних галузей, зокрема, з технічної діагностики є більш якісним.

Структура пошукових машин передбачає наявність інформаційного масиву, в якому зберігаються текстові складові інформаційних ресурсів Інтернет. Знаходження потрібної інформації полягає в організації пошуку в цьому масиві, який також називають індексом або базою даних ІПС [18, 21]. Такі пошукові машини як AltaVista (www.altavista.com), InfoSeek (www.infoseek.com), отримують інформацію з кожного окремого вузла, індексують її, реєструють всю знайдену інформацію і додають до своїх баз даних. Інші, наприклад, і WebCrawler (www.webcrawler.com) і Excite (www.exite.com), мають механізми, які находять вузли з високим трафіком та додають їх до своїх архівів, класифікуючи їх по спаданню популярності [19]. Таким чином, один з компонентів пошукової машини організує пошук в якомусь власному інформаційному просторі (базі даних, архіві і т.д). Інша складова, яка називається пошуковим роботом, дозволяє постійно зв'язуватися з вебсерверами в усьому світі, завантажує з них усі доступні документи, аналізує вміст і включає документи в індекс. Повна автоматизація поповнення зазначеного інформаційного масиву, досить висока частота його оновлення, можливість знаходження будь-якої інформації у глобальній мережі робить пошукові машини більш привабливим засобом пошуку інформації в Інтернет. Пошукові системи діляться на два класи: багатозадачні та спеціалізовані [21]. Багатозадачні системи (Altavista, Infoseek, Google, і т. п.) призначені для пошуку інформації по будь яким запитам. Для виконання цього завдання вони намагаються проіндексувати всю доступну в Інтернет інформацію. Спеціалізовані

системи призначені для відповідей на запити, пов'язані з деякою спеціалізованою предметною областю.

Автономні пошукові агенти машини призначені для організації пошуку за допомогою кількох пошукових засобів. Їх структура відрізняється відсутністю власних будь-яких баз даних, що зберігають інформацію про ресурси Інтернет і наявністю блоку розподілу запитів користувача по відомим даній машині пошуковим засобам. Отримані відповіді пошукових засобів мета пошукова машина ранжує і видає користувачеві єдину відповідь, яка є результатом роботи декількох пошукових машин або каталогів [14].

Майже всі основні пошукові системи мають власну архітектуру, відмінну від інших. Однак ви можете вибрати загальні компоненти для всіх пошукових систем. Відмінності в побудові можуть бути лише у формі реалізації механізмів взаємодії цих модулів. Розглянемо типову архітектуру пошукової системи (рис. 1.1)

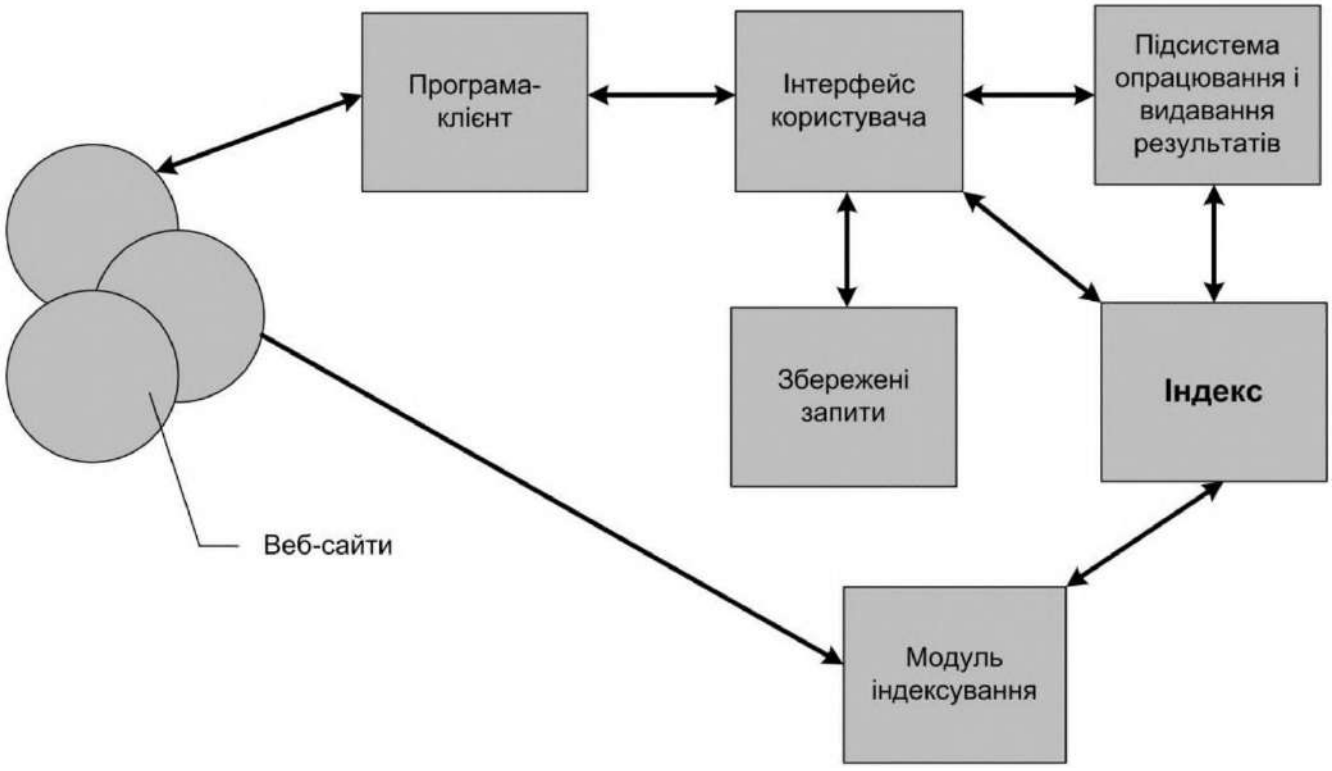


Рисунок 1.1 - Типова архітектура пошукової системи

Детально зупинимось на структурі кожного модуля.

1. Модуль індексації. Застосовується для постійного сканування Інтернету та постійного оновлення бази даних індексів. Цей модуль є основним джерелом інформації, що характеризують стан інформаційних ресурсів мережі. Його будова складається з трьох допоміжних програм (роботів):

• Павук (Spider) - програма, призначена для завантаження вебсторінок. "Павук" завантажує сторінку і видаляє з неї всі внутрішні посилання, тобто завантажується html-код кожної сторінки. Протоколи HTTP використовуються для завантаження робочих сторінок. Павук працює за такою схемою: він посилає на сервер запит "get / path / document" або деякі інші команди HTTP-запиту. У відповідь робот одержує текстовий потік, якийі містить службову інформацію та сам документ. Посилання витягуються із тегів <a>, <area>, <base>, <frame>, <frameset> тощо. Поряд із посиланнями багато роботів обробляють переспрямування (перенаправлення). Кожна завантажена сторінка зберігається у такому форматі:

• URL-адреса сторінки;

• дата завантаження сторінки;

• http - заголовок відповіді сервера;

• тіло сторінки (html-код).

Тому павук робить запит на вміст сторінок так само, як це робить будь-який Інтернет-браузер, надсилаючи запит на сервер HTTP і отримуючи від нього відповідь. Після завантаження вмісту сторінки воно надсилається сканеру та індексатору.

• Сканер (Crawler) - програма, яка автоматично переходить за всіма посиланнями, знайденими на сторінці. Сканер аналізує шляхи, що ведуть від поточної сторінки до інших розділів сайту або до сторінок зовнішніх Інтернет-ресурсів, і визначає подальший порядок павука, який повзає по нитках всесвітньої павутини. Саме сканер знаходить нові сторінки для пошукової системи і передає їх сканеру. Його завдання - визначити, куди павук повинен йти далі, на основі посилань або на основі заздалегідь визначеного списку адрес.

• Робот-індексатор (Indexer) - програма, яка аналізує вебресурси,

завантажені павуками. Індексатор аналізує сторінку на складові частини та аналізує їх, використовуючи власні лексичні та морфологічні алгоритми. Індексатор виконує первинний аналіз вмісту завантаженої сторінки, відбирає основні частини (заголовок сторінки, опис, посилання, заголовки тощо) і розкладає їх на відповідні розділи бази даних пошуку - поміщає в індекс пошукової системи. Цей процес називається індексуванням веб-ресурсів, звідси і назва самої підсистеми. На підставі первинного аналізу індексатор також може вирішити, що сторінка, як правило, "негідна", щоб бути в індексі. Причини цього рішення можуть бути будь-якими, наприклад: сторінка не має заголовка, це точна копія іншої, яка вже є в індексі, або містить посилання на заборонені законом ресурси.

2. Індекс пошукової системи (database index) - це база даних, що містить посилання на індексовані ресурси та зменшені копії веб-сторінок. Він зберігається в пошуковій системі. В індексі пошукова машина зберігає свій "словниковий запас", тобто набір слів і фраз, які знаходяться на Інтернет-сторінках. Він реалізований у формі перевернутого файлу, в якому кожне слово або фраза поєднується з адресами веб-сторінок, на яких вони трапляються. Коротка копія веб-сторінки зберігається у вигляді списку слів, які присутні в тексті сторінки, для кожного з яких перелічені позиції, на яких вона зустрічається на цій сторінці. У цьому випадку стоп-слова відкидаються, а інші слова можна зменшити до початкової форми. Індекс використовується системою для пошуку сторінок із ключовими словами, які вказані в запиті користувача. Індекс постійно поповнюється новою інформацією, зібраною павуком пошукової системи. Для того, щоб сайт з’явився у списку пошукових систем за певними запитами, його або, принаймні, певну частину його сторінок, потрібно включити до індексу пошукової системи. Павук у пошуковій системі може дізнатись про новий сайт одним із двох способів - зв’язавшись із власником сайту або за допомогою посилань з індексованих сайтів на цей сайт.

3. Підсистема для обробки та публікації результатів (Search Engine і Results Engine). Це серце будь-якої пошукової системи. Алгоритми цієї підсистеми

розробники тримають у суворій таємниці, оскільки вони є комерційною таємницею. Ця частина пошукової системи відповідає за адекватність реакції пошукової системи на запит користувача. Застосовується для перекладу запиту користувача з мови пошуку інформації у формальний системний запит, пошуку посилань на інформаційні ресурси в Інтернеті та надання користувачеві результатів цього пошуку. Його можна розділити на два основні компоненти:

• Підсистема ранжирування. Рейтинг - це сортування вебсторінок відповідно до їх запиту. Релевантність сторінки - це ступінь відповідності вмісту сторінки змісту пошукового запиту, і це значення визначається самою пошуковою системою на основі величезної кількості параметрів. На рейтинг сторінки, крім її структури та змісту, впливає також: кількість та якість посилань, направлених на цю сторінку з інших сайтів; вік домену самого сайту; поведінкова характеристика користувачів, які переглядають сторінку, та багато інших факторів.

• Підсистема доставки результатів. Завдання цієї підсистеми включає інтерпретацію запиту користувача, переклад його на мову структурованих запитів до індексу та формування сторінок результатів пошуку. На додаток до тексту запиту, пошукова система може також врахувати:

• Контекст запиту, який формується на основі вмісту раніше зроблених запитів користувачем. Наприклад, якщо користувач досить часто відвідує сайти спортивних новин, коли його запитують про слово "Дніпро" чи "Говерла", він, ймовірно, шукає інформацію про ці футбольні клуби, а не річкову чи гірську систему з однойменною назвою. Це називається персоналізованим пошуком. При цьому результати для одного і того ж звернення для різних користувачів можуть сильно відрізнятися.

• Вподобання користувачів, які пошукова система може "вгадати", аналізуючи посилання, які користувач вибирає на сторінках результатів пошуку. Це ще один з можливих способів скорегувати контекст запиту: дії користувача, здається, повідомляють машині, що саме вона хоче знайти. Зазвичай пошукові машини намагаються додати сторінки до результатів пошуку, які мають відношення до запиту, але стосуються різних сфер життя. Наприклад, коли

користувач цікавиться музикою то пошукова система робить припущення і посилається на сторінки про музикальні групи та їх репертуар, навіть якщо ці сторінки не повністю відповідають вихідному запиту. На наступний запит цього користувача пошукова система може віддати перевагу музичним сторінкам, які містять слова з тексту запиту.

• Регіон, який особливо важливий при обробці комерційних запитів, пов'язаних із придбанням товарів та послуг у місцевих постачальників. Наприклад, якщо користувач проживає у Хмельницькому і хоче придбати пральну машину, швидше за все, цю людину не цікавить ціна пальної машини, наприклад, у Маріуполі, якщо це чітко не прописано в тексті запиту. Очевидно, що результати повинні спочатку включати ціни на пральні машини у Хмельницькому. Тому сучасні пошукові машини поділяють запити на геозалежні та негеонезалежні. Отже, якщо пошукова машина визначає, що запит користувача є геозалежним, він автоматично додає до нього атрибут регіону, який визначається на підставі інформації від провайдера.

• Час. Пошукові системи не рідко аналізують, коли відбулися події, описані на сторінці. Так як інформація має властивість старіти, то користувач спочатку очікує на останні новини, оголошення, поточні прогнози, та інформацію про теперішні події. Тому пошукова система повинна розуміти, що актуальність сторінки залежить від часу, і порівнювати її з часом запиту.

4 Інтерфейс користувача (user interface) - це засіб комунікації користувача з пошуковою системою, тобто з модульом формування запитів та перегляду результатів пошуку. Це форма HTML, яка відкривається за допомогою клієнтської програми, наприклад Chrome, Opera, Mozilla Firefox тощо, в якій користувач вводить запит і натискає кнопку пошуку.

5 Збережені запити (saved queries) - це база даних запитів, які надходять від користувачів. Вони дають можливість пошуковій системі запропонувати "підказку" користувачам, прогнозуючи її в результаті аналізу попередніх збережених запитів.

2. Програма-клієнт (client) - це засіб, для перегляду інформаційних ресурсів

у мережі інтернет. За допомогою програми-клієнта здійснюється звернення, зокрема, до інтерфейсу користувача системи інформаційного пошуку.

3. Вебсайти (WWW sites) - це ті інформаційні ресурси, перегляд яких забезпечується програмами-браузерами.

1.2 Принципи функціонування пошукових систем

Інформаційний пошук (рис 1.2) в інтернеті умовно можна поділити на 2 процеси:

1) фоновий або внутрішній процес
2) зовнішній процес.

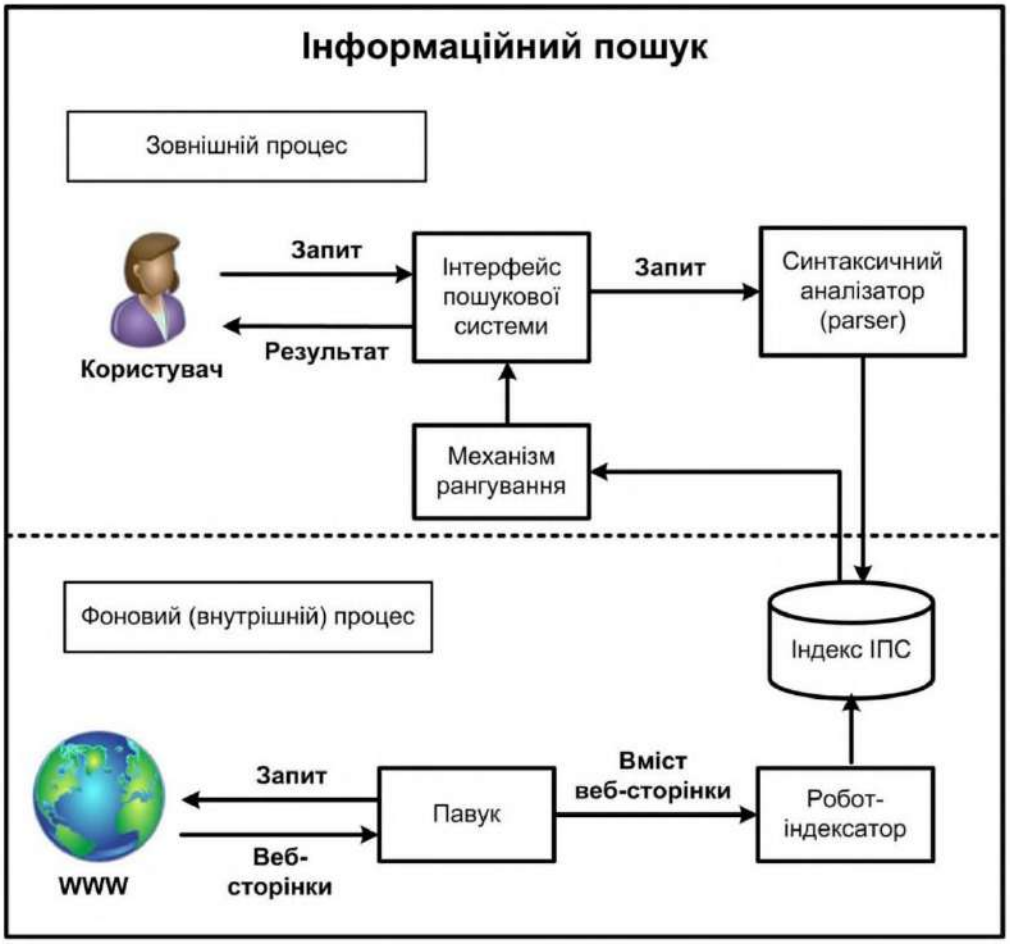


Рисунок 1.2 - Структура інформаційного пошуку

Фоновий процес - це фактично сукупність процесів індексації

інформаційних ресурсів та оновлення індексу пошукової системи. Фоновий процес відбувається непомітно для користувачів та без їх участі. Це відбувається "на тлі" пошукової системи безпосередньо з користувачем. Однак без фонового процесу основний процес неможливий, тобто робота з користувачем. Це пов'язано з тим, що пошукова система на момент отримання запиту від користувача здійснює пошук лише в межах попередньо сформованого індексу. Якщо індекс не існує, пошук не може відбутися. Крім того, індекс пошукової системи потрібно періодично оновлювати, щоб результати пошуку були актуальними. Тому фоновий процес є безперервним. Павуки сканують посилання на заданий набір ресурсів, завантажують вміст сторінки, витягують посилання на нові сторінки з отриманих ресурсів. Зміст сторінок передається до робочого індексу, який аналізує сторінки та вносить інформацію до покажчика літака.

Зовнішній процес - це поєднання процесу обробки запиту користувача та видачі результатів.

Через інтерфейс пошукова система отримує запит від користувача. Запит передається синтаксичному аналізатору та піддається морфологічному аналізу. Для кожного документа, що міститься в базі даних, створюється інформаційне середовище, яке згодом відображатиметься як фрагмент. Фрагмент - це невеликий фрагмент тексту сайту, який відображається в результатах пошуку, що дозволяє оцінити вміст веб-сторінки без необхідності переходити за посиланням, і який виділяє ключові слова, введені користувачем у форму пошуку.

Дані, отримані з індексу, передаються як вхідні параметри до спеціального модуля ранжування. Він обробляє дані з усіх документів, в результаті чого кожен документ обчислює власну оцінку, яка характеризує релевантність запиту, введеного користувачем, та різних компонентів цього документа, що зберігаються у індексі пошукової системи.

Залежно від вибору користувача, сформований рейтинг може коригуватися додатковими умовами (наприклад, розширені параметри пошуку). Потім генерується фрагмент, тобто для кожного знайденого документа з таблиці документів витягується заголовок, короткий текстовий фрагмент - реферат, в

якому виділяються ключові слова, а також посилання на документ.

Збір інформації іноді більш конкретно називають скануванням вебсторінок в контексті пошукової системи. Вебсканери (також їх називають павуками, хробаками або мандрівниками) - це програми, які використовуються для методичного сканування мережі інтернет для збору інформації про вміст та структуру вебсторінок. Вони переглядають мережу від імені пошукової системи, подібно до того, як користувач переходить за посиланнями на різні сторінки. Спочатку сканер читає з переліку вихідних URL-адрес і відвідує документи за цими URL-адресами. Кожен ресурс, який відвідується, обробляється, а URL-адреси, що містяться на цьому ресурсі, виділяються та додаються в чергу. Потім сканер вибирає наступну URL-адресу з черги і продовжує процес, поки не буде завантажено певну кількість документів або переглянуто всі локальні обчислювальні ресурси. Сканер передає зібрану інформацію до бази даних ресурсів, поки всі документи в сховищі не закінчаться, або поки пошук безпосередньо не досягне заздалегідь визначених критеріїв. Документи, отримані сканером, зберігаються в базі вебресурсів. Щоб зменшити обсяг необхідного місця для зберігання, сторінки часто стискаються перед їх збереженням.

Цей базовий алгоритм сканування в Інтернеті може бути змінений або вдосконалений за допомогою багатьох опцій, які надають пошуковим системам різні рівні охоплення або упередженості. Наприклад, сканери можуть попереджати якомога більше ресурсів, не випускаючи сторінок, глибоко прихованих у кожному вебресурсі. Інші сканери можуть спеціалізуватися на ресурсах в одному конкретному домені, таких як урядові сторінки. Модуль керування скануванням відповідає за контроль роботи сканування.

Після того, як пошукова машина пройшла щонайменше один повний цикл сканування, модуль керування скануванням може бути проінформований кількома індексами, створеними під час попереднього сканування. Модуль управління може, наприклад, використовувати графік посилань попереднього сканування, щоб вирішити, які посилання сканери повинні досліджувати в даний час, а які посилання вони можуть пропускати. Контроль сканування може також

використовувати зворотний зв'язок із шаблонами застосування для керівництва процесом сканування. Деякі вебсайти, можливо, доведеться дуже регулярно сканувати, оскільки їх вміст може часто змінюватися, наприклад газети або вебресурси, що стосуються букмекерських контор.

Індексатор обробляє сторінки у сховищі та створює базовий індекс пошукової системи. Індексатор визначає кожну сторінку словами та реєструє появу кожного слова на сторінці. Наступним кроком є вилучення ключових слів зі сторінок та створеним індексом ключових слів та відповідних ресурсів, на яких вони були знайдені. Індексатор також використовується для вираховування таких оцінок, як частота документів кожного слова, які можна використовувати для ранжування результатів або для подальшого опрацювання.

Метод індексування, який надає безпосередній вплив на використовуваний метод пошуку, є одним з необхідних критеріїв, за яким можна робити класифікацію пошукових машин. Залежно від цього критерію інформаційно пошукові системи (ІПС) діляться на наступні класи [9, 18]:

- ІПС, які підтримують бінарне індексування;
- ІПС, які підтримують морфологічне індексування;
- системи, які підтримують індексування за ключовими словами.

Побудова індексів методом бінарного індексування (хешовані індекси, B-дерева, T-дерева) пріоритетна, внаслідок своєї контекстної та мовної незалежності. При такому індексуванні пошук ведеться на основі алгоритмів "нечіткого пошуку", тобто пошуку з помилками. У цьому випадку допускається частковий (з заданою кількістю помилок на початку, середині і кінці слова) збіг слів з шаблоном запиту. Морфологічне індексування проводиться з урахуванням морфології та семантики мови, що робить даний метод контекстно-залежним. При використанні даного методу слова перетворюються в словоформи з відсіканням суфіксів і закінчень, що дозволяє шукати відмінки шаблонів. Напрям «ключового» індексування є подальшим розвитком індексування по всьому документу. Даний метод значно скорочує об'єм індексу, що позитивно впливає на час пошуку.

В основі методу аналізу текстової інформації і реалізації варіантів її пошуку лежать моделі пошуку [18, 21]. Модель пошуку - це поєднання способу подання документів, пошукових запитів та виду критерію релевантності документів. Саме різні варіації цих складових і визначають численну кількість реалізацій систем текстового пошуку.

Кількість ключових слів дуже велика, і тому їх повний перелік згубно позначиться на пошуковій системі. На основі деяких спостережень двигун ІПС повинен мати можливість оцінити важливість висловів як ключових слів. На основі цього типу евристичних спостережень визначаються ключові слова та їх важливість.

Після визначення ключових слів вони індексуються. Пошукові системи, як правило, підтримують таблицю індексів та інвертовану таблицю індексів. Індексна таблиця складається з ключового слова та посилання на сторінку, на якій вона знаходиться. Інвертована таблиця індексу містить те саме поле, але в зворотному порядку посилання на сторінку та всі ключові слова, представлені на ній.

Отримані результати пошуку передаються користувачеві у формі SERP (Search Engine Result Page) – видані сторінки пошукових результатів [41].

Сторінки результатів пошукової системи - це веб-сторінки, що надаються користувачам, коли вони щось шукають в Інтернеті за допомогою пошукової системи, наприклад Google. Користувач вводить свій пошуковий запит (часто використовуючи конкретні терміни та фрази, відомі як *ключові слова*), після чого пошукова система представляє їм SERP.

Кожен SERP унікальний, навіть для пошукових запитів, що виконуються в одній пошуковій системі з використанням тих самих ключових слів або пошукових запитів. Це пояснюється тим, що практично всі пошукові системи налаштовують досвід для своїх користувачів, представляючи результати на основі широкого кола факторів, що перевищують їх пошукові терміни, таких як фізичне місцезнаходження користувача, історія перегляду та соціальні налаштування. Два

SERP можуть здаватися однаковими та містити багато однакових результатів, але часто мають незначні відмінності.

Поява сторінок результатів пошукової системи постійно змінюється завдяки експериментам, проведеним Google, Bing та іншими провайдерами пошукових систем, щоб запропонувати своїм користувачам більш інтуїтивну та чуйну роботу. Це в поєднанні з новими та швидко розвиваються технологіями у пошуковому просторі означає, що сучасні SERP значно відрізняються за зовнішнім виглядом від своїх попередніх попередників.

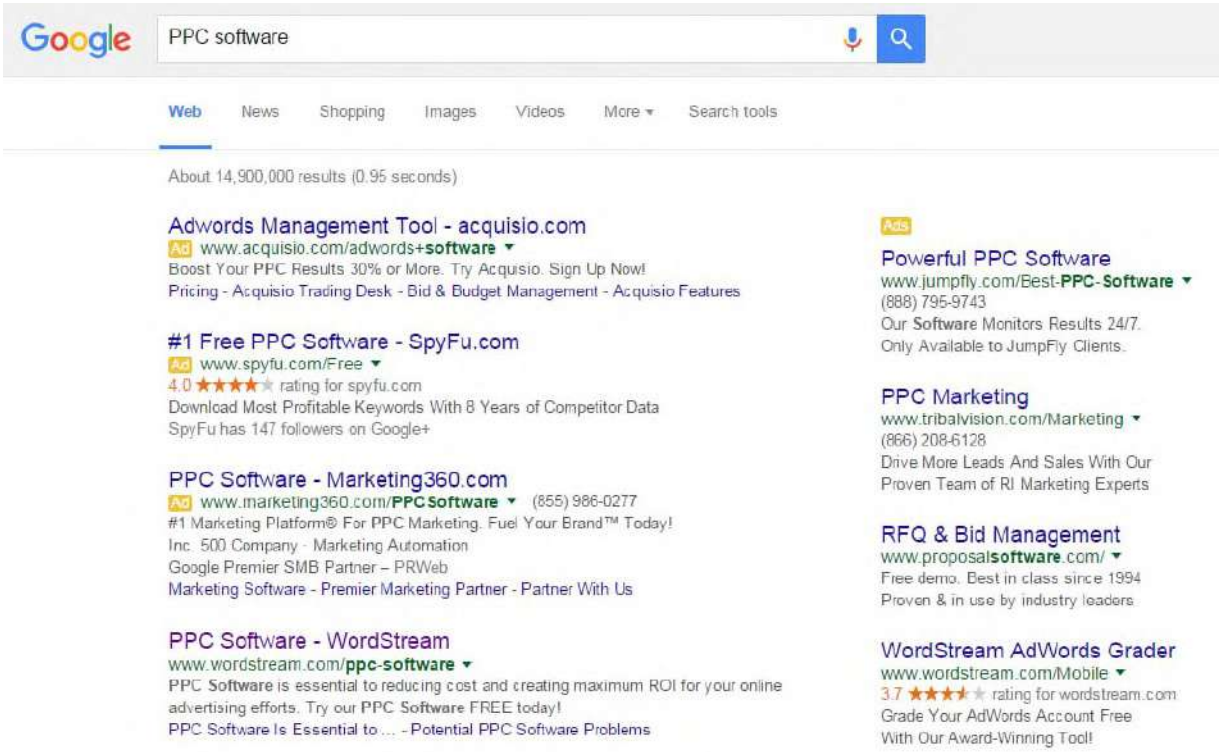


Рисунок 1.3 - Приклад першої *сторінки з SERP*

SERP, як правило, містять два типи вмісту - "органічні" результати та платні результати. Звичайні результати - це списки вебсторінок, які з’являються в результаті роботи алгоритму пошукової системи (про це коротше). Фахівці з оптимізації пошукових систем, широко відомі як SEO, спеціалізуються на оптимізації веб-контенту та веб-сайтів для більш високого рейтингу в органічних результатах пошуку.

Органічні результати на SERP. Поле з правого боку цього SERP відоме як Графік знань (також іноді його називають Вікно знань). Це функція, яку Google представив у 2012 році, яка збирає дані на поширені запитання з джерел у Інтернеті, щоб надати стислі відповіді на запитання в одному центральному місці на SERP. У цьому випадку ви можете побачити широкий спектр інформації про Авраама Лінкольна, таку як дату та місце його народження, зріст, дату вбивства, його політичну приналежність та імена його дітей - багато з них які факти мають власні посилання на відповідні сторінки.

Деякі SERP матимуть значно більше органічних результатів, ніж інші, наприклад, приклад вище. Це пов’язано з різними намірами різних пошуків. Існує три основних типи пошуку в Інтернеті :

- Інформаційний
- Навігаційний
- Транзакційна

Інформаційні пошуки - це ті, в яких користувач сподівається знайти інформацію з певної теми, наприклад, Абрахам Лінкольн. Немає сенсу розміщувати рекламу чи інші види платних результатів на такому SERP, оскільки пошуковий запит "Авраам Лінкольн" має дуже низький *комерційний намір* ; переважна більшість шукачів, які використовують цей пошуковий запит, не хочуть щось придбати, і як такі, лише інформаційні результати відображаються на SERP.

Навігаційні запити - це запитання, в яких користувач сподівається знайти певний вебсайт за допомогою пошуку. Це може бути для осіб, які шукають певний веб-сайт, намагаючись знайти веб-сайт, URL-адресу якого вони вже не можуть запам'ятати, або інший тип навігаційної мети.

Нарешті, транзакційними є ті пошукові запити, в яких результати оплати найімовірніше відображатимуться на SERP. Транзакційні пошуки мають високий комерційний намір, і пошукові запити, що ведуть до транзакційних SERP, можуть включати такі ключові слова, як "купити" та інші терміни, що вказують на сильне бажання зробити покупку.

Платні результати. На відміну від звичайних результатів, платними є ті, які були сплачені за показ рекламодавцем. Раніше платні результати майже виключно обмежувались невеликими текстовими оголошеннями, які зазвичай відображалися зверху та праворуч від органічних результатів. Однак сьогодні платні результати можуть приймати широкий спектр форм, і існує десятки рекламних форматів, які відповідають потребам рекламодавців.

Звичайні результати - це списки, проіндексовані пошуковою системою на основі ряду факторів, також відомих як " рейтингові сигнали ".

Наприклад, алгоритм пошуку, який використовує Google, містить сотні факторів ранжування, і хоча ніхто за межами Google точно не знає, якими вони є, деякі вважаються важливішими за інші. У минулому профіль посилань на сайт - кількість зовнішніх посилань, які посилаються на певний веб-сайт або веб-сторінку з інших веб-сайтів - був важливим сигналом рейтингу. Це все ще певною мірою (саме тому Вікіпедія займає так важливе місце в органічних результатах для стількох запитів), хоча пошук просувається настільки швидкими темпами, що ранжування сигналів, які колись мали вирішальне значення для алгоритму пошуку, сьогодні може бути менш важливим

1.3 Аналіз моделей інформаційного пошуку

Класичні моделі пошуку інформації трактують документи як набори ключових слів, що представляють ці документи, які називаються термінами. Терм, як правило, - це просто слово, семантика якого допомагає характеризувати основний зміст документа.

Формально, в загальному опис моделі інформаційного пошуку складається з 4 частин [11]: $M = \langle D, Q, F, R(q, d_i) \rangle$, де D - множина типів представлень документів; Q - множина типів опису інформаційних потреб користувача, тобто звернень; F - загальний базис, в рамках якого відбувається моделювання характеристик документів і запитів, а також опис взаємозв'язків поміж ними;

$R(q,d_i)$ - функція ранжування, яка парі документ-звернення зіставляє деяке дійсне число.

Моделі інформаційного пошуку можна поділити на три класи [30]:

Теоретико-множинні моделі. Моделі цього класу використовують як базис теорію множин. Класичний приклад - булева модель, яка дає можливість користувачеві формувати запит у вигляді булевого виразу, використовуючи І, АБО, НІ. У рамках цієї моделі документи і запити видаються у вигляді множин термів.

Імовірнісна модель. Базисом для таких моделей виступає теорія імовірності. В якості оцінки релевантності документа запиту користувача використовується ймовірність того, що користувач визнає документ дійсно релевантним. Найвища ефективність пошуку досягається у випадку, коли результуючі документи ранжуються за зменшенням ймовірності їх релевантності запиту. Спочатку для кожного документа здійснюється оцінка ймовірності релевантності запиту, а потім за цими оцінками виконується ранжирування документів.

Алгебраїчна модель. У рамках алгебраїчних моделей документи і звернення описуються у вигляді векторів у багатовимірному просторі. Каркасом для таких моделей виступають алгебраїчні методи. Вони дають можливість ранжувати результуючу множину документів запиту. Визначення ваг термів і оцінка міри близькості векторів різними способами дає можливість різноваріантної модифікації зазначеної моделі пошуку документів.

У рамках кожного з класів було запропоновано багато альтернативних моделей. Враховуючи ряд недоліків, на практиці класичні теоретико-множинні моделі досить популярні через свою простоту. Хоча імовірнісні моделі надають найбільш природні можливості формально описати проблему інформаційного пошуку, чомусь їх популярність відносно невелика. Найбільш популярними в пошуку є алгебраїчні моделі, оскільки їх практична ефективність зазвичай виявляється вищою. Нові моделі, які з’являються останнім часом є гібридними і володіють властивостями моделей різних класів.

Розглянемо моделі стратегій інформаційного пошуку. Розрізняють моделі, що базуються на допошуковій і післяпошуковій взаємодії користувача з інформаційною системою [35]. В процесі допошукової взаємодії користувачу надаються відомості з бази знань, корисні при формуванні і корекції запиту [12]. Післяпошукова взаємодія базується на оцінці і використанні на наступних етапах пошуку проміжної видачі найденої інформації [29]. Існує три варіанти післяпошукової взаємодії, в залежності від способу формування запиту і його корекції. Перший ґрунтується на "ручному" формулюванні і корекції запиту користувачем. Для уточнення запиту використовуються результати попередньої ітерації. Другий вид моделей базується на ручному формуванні запиту і його автоматичній корекції на наступних етапах пошуку [12]. До третього різновиду відносяться моделі стратегій, у відповідності до яких на основі бази вибірки відповідних документів автоматично формується початковий запит. Результати повторних кроків пошуку використовуються для автоматичної корекції запиту. Такі самонавчаючі стратегії забезпечують досить високі показники ефективності пошуку. Ефективність стратегій інформаційного пошуку залежить від правил ототожнення опису об'єктів і запитів.

Критерій видачі – це формальне правило, відповідно до якого, в інформаційному масиві визначаються документи, що підлягають видачі у відповідь на запит, що надійшов у систему [9]. Розрізняють три види критеріїв видачі, сформульованих у термінах теорії множин [21]. При цьому пошукові образи документів і запитів розуміються як множини лексичних одиниць ІПМ.

Критерій *«на збіг»* - для видачі потрібно, щоб лексичні одиниці пошукового образу документа і пошукового розпорядження збіглися.

Критерій *«на включення»* – у відповідь на запит видаються ті документ, пошукові образи яких включають цілком пошукове розпорядження запиту. Якщо пошуковий образ документа представити у вигляді множини M_d, а пошукове розпорядження – M_q, то повідомлення видається, коли $M_q \subset M_d$.

Критерій *«на перетинання»* вимагає не повного, а часткового збігу лексичних одиниць документа і запиту. Цей критерій часто підсилюється за

рахунок обчислювальних операцій, виконуваних у процесі пошуку. Наприклад, розповсюджений метод «вагових» коефіцієнтів: при формулюванні пошукового розпорядження споживач інформації оцінює значимість кожної лексичної одиниці таким коефіцієнтом. Для цього використовується спеціальна шкала з повними і негативними значеннями. Крім того, споживач задає граничну чисельну величину K . Повідомлення видається, якщо сума «вагових» коефіцієнтів тих індексів пошукового розпорядження запиту, що збіглися з індексами пошукового образа документа, більша або рівна K .

Існує два підходи до моделювання навігаційних маршрутів. Перший передбачає попередню побудову маршрутів і їх класифікацію у відповідності до прогнозованих інформаційних потреб користувача. Інший полягає в тому, що користувачам пропонується інструментарій для самостійного перегляду гіпертексту. При цьому використовуються ієрархічні вказівники зв’язків між текстами гіпертексту і когнітивні карти в вигляді багатовіконного подання інформації.

Задача аналізу вхідних повідомлень системи діагностування включає три підзадачі – морфологічний, синтаксичний і семантичний аналіз [26].

Під морфологічним аналізом розуміють обробку слів без зв’язку з контекстом, з метою ототожнювання їх різноманітних форм і отримання граматичної і семантичної інформації [23]. Морфологічний аналіз використовується для приведення слів до базової форми у статичних методах аналізу.

Існує два основних підходи до морфологічного аналізу декларативний і процедурний [25]. При першому підході в базах знань зберігаються всі можливі словоформи кожного слова. При процедурному підході в базі знань зберігаються основи слів. Процес аналізу полягає в їх виділенні у вхідних повідомленнях і пошуку необхідних морфологічних ознак.

Декларативний підхід значно простіший ніж процедурний, але потребує більших об’ємів пам’яті, а при процедурному підході збільшується трудоємність підготовки морфологічної частини бази знань і складність алгоритмів аналізу.

Синтаксичний аналіз повідомлення полягає в побудові його синтаксичної структури на основі інформації, отриманої на етапі морфологічного аналізу [22]. Синтаксичний аналіз на основі забезпечення роботи з більш узагальненими семантичними елементами підвищує інтелектуальність опрацювання текстової інформації. В лінгвістиці розглядають три найбільш поширенні моделі для опису структури речення: системи складових, дерева синтаксичного підпорядкування і системи синтаксичних груп [24].

Система складових являє собою дужкову структуру з різноманітною глибиною вкладень, де кожна пара дужок обмежує з двох сторін синтагму чи синтагматичну структуру.

Дерево синтаксичного підпорядкування – це орграф, вершинам якого відповідають слова повідомлення, а дуги з'єднують всі пари вершин, що знаходяться у відношенні синтагматичного підпорядкування типу *"той, якого визначають*", "той, що визначає", причому кожній синтагмі повідомлення відповідає цього дерева. Крім того, на множині вершин дерева підпорядкувань встановлено лінійний порядок, узгоджений з звичайним порядком слів у реченні. Доведено, що кожній системі складових може відповідати єдине дерево синтаксичного підпорядкування [35, 81].

Модель для опису синтаксису речення у вигляді системи синтаксичних груп розроблена на основі двох попередніх. В рамках цієї моделі синтаксичну структуру повідомлення зображають у вигляді орграфа, вершинами якого є окремі слова і словосполучення [35, 81, 106, 111, 141].

Глибина синтаксичного аналізу повідомлень залежить перш за все від ступеня залучення семантичних ознак. Тому розрізняють поверхневий і глибинний аналіз [35]. При поверхневому аналізі встановлюють існування синтаксичного зв'язку між словами і його напрям. Метою глибинного аналізу є диференціація вказаних зв'язків, для чого використовується семантика.

Існує два найбільш поширених підходи для встановлення синтаксичної структури повідомлення – метод послідовного аналізу (локальний) і метод фільтрів (глобальний) [18].

При локальному синтаксичному аналізі для кожного слова повідомлення реалізується послідовність дій, з метою виявлення для нього керуючого ланцюга і типу синтаксичного зв’язку. Для цього перевіряють морфологічні признаки слова і його оточення, використовуючи інформацію, отриману для слів на попередніх кроках алгоритму аналізу [21].

Суть методу фільтрів полягає у побудові наборів можливих зв’язків між словами повідомлення, що претендують на роль синтагм, і вилученню інших зв’язків [35].

Відомі два принципово різних підходи до семантичного аналізу повідомлень [32]. Ці підходи справедливі при формуванні інформаційного забезпечення. Першому підходу передує синтаксичний, другий оснований на перевазі семантичного над синтаксичним. При цьому синтаксична інформація використовується як допоміжний засіб. Семантичний аналіз як продовження синтаксичного полягає в тому, що отримана на виході останнього синтаксична структура перетворюється до вигляду, що задається семантичним графом, вершинами якого є певні поняття, а дугами – відношення між ними. Процедура побудови семантичного графа полягає в покроковій заміні фрагментів синтаксичного подання повідомлень їх семантичними еквівалентами з використанням бази знань [11].

Задача інтерпретації повідомлень системою діагностування на основі їх семантичних графів включає два етапи – синтаксичний і морфологічний синтез [35]. При синтаксичному синтезі вирішуються дві задачі: побудова синтаксичної структури вихідного повідомлення і впорядкування його слів [30]. При вирішенні першої задачі використовують граматики, правила висновків яких в лівій частині мають деякі підграфи семантичного графа, а в правій – відповідні їм дерева синтаксичного підпорядкування. Друга задача вирішується в три етапи: з дерева синтаксичного підпорядкування формуються послідовності суміжних слів, зв’язані локальними синтаксичними відношеннями, потім послідовності об’єднуються в групи слів, які впорядковуються з врахуванням існуючих правил їх розташування [13].

Морфологічний синтез слів слід розглядати як процес, обернений до морфологічного аналізу. Існує два шляхи реалізації морфологічного синтезу – декларативний і процедурний [5]. При декларативному із словника вибирають основи слів з необхідною морфологічною інформацією. При процедурному підході словоформи синтезують за допомогою таблиць морфологічних ознак.

Щоб оцінити результати пошуку розглядають пертинентність та релевантність.

Релева́нтність (від англ. relevance) - ступінь відповідності отримуваного результату бажаному. В термінах інформаційного пошуку — це ступінь відповідності результатів пошуку до запиту.

Пертинентність (від англ. pertinent) - відношення корисної інформації до загального об’єму отриманої інформації. Тобто відповідність знайдених пошуковою системою результатів інформаційним потребам користувача. Скорочено - ця відповідність може бути виражена у вигляді відсотків як ефективність пошуку.

Внаслідок надмірності і недостатності звичайна мова не може використовуватися в якості інформаційно-пошукової мови (ІПМ), тому що це привело б при пошуку до великих втрат інформації [18].

Штучна ІПМ вводиться для того, щоб забезпечити повноту і точність видачі інформації при пошуку.

Основні вимоги до ІПМ зводяться до наступного [17]:

- однозначність - кожен запис повинен мати тільки один зміст, і, навпаки, будь-який зміст повинен одержувати однакове подання не ІПМ (відсутність синонімів, антонімів, омонімів);
- експліцитне (явне) вираження корисних для пошуку логічних відношень і психологічних асоціацій між словами ІПМ;
- можливість коректування і доповнення, тобто відкритість ІПМ;
- зручність користування – компактність записів ІПМ;

Інформаційному забезпеченню процесу діагностування характерні такі особливості: різноманітність форм подання інформації, висока інтенсивність і

великий об’єм корегувань інформаційної бази; адаптивність стратегій інформаційного пошуку і маршрутів навігації в базах знань великого об’єму [8].

1.4 Принципи функціонування алгоритму пошуку в Google

Велика кількість інформації доступна в Інтернеті, тому знайти необхідні дані було б практично неможливо без певної допомоги та сортування. Системи позиціонування Google сортують сотні мільярдів веб-сторінок по пошукових індексах, щоб швидко знайти найрелевантніші результати та відобразити їх зручним для користувача способом.

Ці системи позиціонування складаються з ряду алгоритмів. Щоб надати найбільш корисну інформацію, алгоритми пошуку враховують багато факторів, включаючи слова у запиті, релевантність та зручність використання сторінок, компетентність джерел, а також ваше місцезнаходження та налаштування. Важливість кожного фактору залежить від структури вашого запиту. Наприклад, час редагування відіграє більшу роль у відповіді на запити щодо поточних тем новин, ніж у запитах про отримання слів зі словників.

Пошукова система Google має три основні функції:

1. Сканування: Проглядає вміст в Інтернеті, переглядаючи код / вміст кожної знайденої URL-адреси.
2. Індекс: Зберігає та впорядковуйте вміст, знайдений під час сканування. Після того, як сторінка потрапила в індекс, вона вже запущена і відображатиметься як результат для відповідних запитів.
3. Рейтинг: Надає ті фрагменти вмісту, які найкраще відповідатимуть на запит шукача, а це означає, що результати впорядковуються найбільш релевантними до найменш релевантних.

Алгоритми Google - це складна система, яка використовується для отримання даних із його індексу пошуку та миттєвого отримання найкращих можливих результатів для запиту. Пошукова система використовує комбінацію

алгоритмів та численні сигнали ранжування для розміщення вебсторінок, ранжованих за релевантністю на своїх SERP.

У перші роки Google лише кілька разів оновлював свої алгоритми. Зараз Google щороку вносить тисячі змін.

Більшість з цих оновлень настільки незначні, що залишаються абсолютно непоміченими. Однак іноді пошукова машина випускає основні алгоритмічні оновлення, які суттєво впливають на SERP [25]:

- Fred
- Intrusive Interstitials Update
- RankBrain
- Mobilegeddon
- Panda
- Penguin
- Hummingbird
- Payday
- Pigeon
- EMD (Exact Match Domain)
- Page Layout Algorithm

Розглянемо 8 основних оновлень алгоритму Google.

Panda призначає вебсторінкам так званий "показник якості"; цей бал потім використовується як коефіцієнт ранжування. Спочатку Panda була фільтром, а не частиною рейтингу Google, але в січні 2016 року вона була офіційно включена в основний алгоритм. Розгортання панд почастішали, тому штрафні санкції та стягнення зараз відбуваються швидше.

Мета Google Penguin - знизити рейтинг сайтів, посилання яких він вважає маніпулятивними. З кінця 2016 року «Пінгвін» є частиною основного алгоритму Google; на відміну від Panda, він працює в режимі реального часу.

Hummingbird допомагає Google краще інтерпретувати пошукові запити та надавати результати, що відповідають задуму шукача (на відміну від окремих термінів у запиті). Незважаючи на те, що ключові слова продовжують залишатися

важливими, Hummingbird дає можливість сторінці класифікуватися за запитом, навіть якщо він не містить точних слів, які ввів пошуковий запит. Це досягається за допомогою обробки природної мови, яка спирається на приховане семантичне індексування, спільні терміни та синоніми.

Pigeon впливає на ті пошукові запити, в яких місцезнаходження користувача відіграє важливу роль. Оновлення створило тісніші зв'язки між локальним алгоритмом та основним алгоритмом: традиційні фактори SEO тепер використовуються для ранжирування місцевих результатів.

Оновлення мобільних пристроїв Google (відоме як Mobilegeddon) гарантує, що сторінки, зручні для мобільних пристроїв, посідають перше місце серед мобільних пошукових запитів, тоді як сторінки, не оптимізовані для мобільних пристроїв, відфільтровуються з результатів пошуку або серйозно занижуються.

RankBrain є частиною алгоритму Google Hummingbird. Це система машинного навчання, яка допомагає Google зрозуміти значення запитів та забезпечити найкращі результати пошуку у відповідь на ці запити. Google називає RankBrain третім за значимістю фактором рейтингу. Хоча ми не знаємо тонкощів і результатів RankBrain, загальна думка полягає в тому, що він визначає особливості релевантності вебсторінок, класифікованих за даним запитом, які в основному є факторами ранжування, специфічними для запиту.

Оновлення Possum забезпечило, що місцеві результати більше відрізняються залежно від місцезнаходження шукача: чим ближче ви знаходитесь до адреси компанії, тим більша ймовірність побачити її серед місцевих результатів. Поссум також призвів до більшої різноманітності серед рейтингу результатів за дуже схожими запитами, такими як «дантист Мінесота» та «стоматолог Мінесота співпраця». Цікаво, що Поссум також дав поштовх компаніям, розташованим за межами фізичного району міста.

Останнє з підтверджених оновлень Google Фред націлює вебсайти, які порушують інструкції Google для веб-майстрів. Більшість уражених веб-сайтів - це блоги з низькоякісними публікаціями, які, як видається, створюються здебільшого з метою отримання доходу від реклами.

Під час сканування Google виявляє нові сторінки і додає їх до своєї базу. Для автоматизації роботи він використовує «боти». «Googlebots» відвідують список URL-адрес, отриманих в під час минулого сканування і доповнених даними карти сайту, яку надають веб-майстри і аналізують їх зміст.

Основна мета процесу індексації – це швидко реагувати на пошукові запити. При цьому результати, які вважаються більш потрібними для користувача, навмисно отримують більш високий рейтинг (ранг) [7].

Google використовує понад 250 факторів для визначення релевантності і значущості конкретної сторінки. Коли хтось виконує пошук, пошукові системи переглядають індекс ресурсів, щоб знайти дуже релевантний вміст, сподіваючись вирішити запит шукача. Таке впорядкування результатів пошуку за релевантністю називається рейтингом. Загалом, можна припустити, що чим вище рейтинг веб-сайту, тим більш доречною пошукова система вважає, що сайт відповідає запиту.

1.5 Постановка задачі

У першому розділі розглядаються основні теоретичні положення по пошуку інформації в мережі інтернет.

Як і більшість компаній у сферах пошуку, Google ніколи не буде публічно розкривати свої алгоритми пошуку і фактори ранжирування результатів. Але, для того, щоб зменшити впроваджуваний контент в результатах пошуку, компанія інформує веб-майстрів про те, коли і як змінилися головні стандарти якості відбору.

Сформулюємо основні завдання магістерського дослідження:

1. Проаналізувати та детально дослідити принципи пошуку інформації в Google.
2. Розглянути математичну модель існуючого алгоритму та вдосконалити її.
3. Вдосконалити алгоритм ранжування сайтів пошукової системи Google, за рахунок враховування популярності сторінки в тематичних соціальних спільнотах.

4. Розробити метод виявлення інформаційного впливу соціальних мереж на індексацію сторінки та формування переліку зовнішніх гіперпосилань, для зміни коефіцієнтів ранжування відповідних сторінок.
5. Провести дослідження розробленого методу, та оцінити його ефективність

2 МАТЕМАТИЧНА МОДЕЛЬ МЕТОДУ MODPR

2.1 Математична модель, яка використовується в алгоритмах пошуку Google

Пошук корисної інформації в мережі інтернет - це те, що багато хто з нас сприймає як належне. Завдання перебору всіх цих ресурсів, щоб знайти корисну інформацію, є монументальним. Ось чому пошукові системи використовують складні алгоритми - математичні інструкції, які вказують комп'ютерам, як виконувати призначені завдання.

Алгоритм Google виконує для вас роботу шляхом пошуку вебсторінок, що містять ключові слова, за якими ви шукали, а потім присвоєння рангу кожній сторінці на основі кількох факторів, включаючи кількість разів, коли ключові слова з'являються на сторінці. Сторінки з вищим рейтингом з'являються далі на сторінці результатів пошуку в пошуковій системі Google SERP, що означає, що найкращі посилання, що стосуються вашого пошукового запиту, теоретично є першими, які перелічує Google.

Для адміністраторів вебресурсів, якщо їх видно в Google, це може призвести до значного збільшення відвідуваності та видимості вебсайтів. У 2007 році Google перевершив Microsoft як найбільш відвідуваний вебсайт [35]. При такому великому обсязі відвідування хорошого місця на Google SERP може означати значний приріст кількості відвідувачів сайту.

Функція пошуку ключових слів від Google подібна до інших пошукових систем. Автоматизовані програми, звані павуками або сканерами**,** подорожують Інтернетом, переходячи від посилання до посилання та створюючи індексну сторінку, що включає певні ключові слова. Google посилається на цей індекс, коли користувач вводить пошуковий запит. У пошуковій системі перелічені сторінки, що містять ті самі ключові слова, що були в пошукових термінах користувача. Павуки Google можуть також мати деякі більш розширені функції, такі як можливість визначати різницю між вебсторінками з фактичним вмістом та

переспрямовуючими сайтами - сторінками, які існують лише для перенаправлення трафіку на інший ресурс.

Розміщення ключових слів відіграє важливу роль у пошуку Google сайтів. Google шукає ключові слова на кожному ресурсі, але деякі розділи важливіші за інші. Наприклад, включення ключового слова в заголовок веб-сторінки - це гарна ідея. Google також здійснює пошук ключових слів у заголовках. Заголовки мають різний розмір, і ключові слова у більших заголовках цінніші, ніж якщо вони в менших заголовках. Розподіл ключових слів також важливий. Веб-майстрам слід уникати надмірного використання ключових слів, але багато людей рекомендують регулярно використовувати їх на всій сторінці.

Найважливішою функцією алгоритму Google є, мабуть, система PageRank, запатентований автоматизований процес, який визначає, де кожен результат пошуку відображається на сторінці повернення пошукової системи Google. Більшість користувачів, як правило, концентруються на перших кількох результатах пошуку, тому потрапляння місця у верхній частині списку зазвичай означає більший трафік користувачів [42]. Розглянемо, як Google визначає рейтинг результатів пошуку. Відомо, що:

- PageRank присвоює рейтинг або оцінку кожному результату пошуку. Чим вищий бал сторінки, тим далі відображатиметься список результатів пошуку.
- Бали частково визначаються кількістю інших ресурсів, які посилаються на цільову сторінку. Кожне посилання зараховується як голос за ціль. Логіка цього полягає в тому, що сторінки з високоякісним вмістом будуть посилатися частіше, ніж посередні сторінки.
- Не всі голоси рівні. Голоси з високопоставленого ресурсу нараховують більше, ніж голоси з низькопоставлених сайтів. Ви не можете реально підвищити рейтинг однієї веб-сторінки, створивши купу порожніх веб-сайтів, які повертаються назад до цільової сторінки.
- Чим більше посилань дає вебсторінка, тим більш розрідженою стає її сила голосу. Іншими словами, якщо сторінка з високим рейтингом посилається на

сотні інших сторінок, кожен окремий голос не буде враховуватися настільки ж, як і якщо сторінка посилається лише на кілька сайтів.

- Інші фактори, які можуть вплинути на оцінку, включають те, як довго існує сайт, міцність доменного імені , як і де ключові слова з'являються на сайті та вік посилань, що переходять на сайт і з нього. Google, як правило, приділяє більшу цінність сайтам, які існують деякий час.
- Деякі люди стверджують, що Google використовує групу тестерів для оцінки результатів пошуку, сортуючи результати вручну, щоб вибрати найкращі посилання Google заперечує це і заявляє, що, хоча і працює мережа людей для тестування оновлених формул пошуку, вона не покладається на людей для сортування та ранжування результатів пошуку.

Сучасні пошукові системи використовують методи ранжування результатів, щоб насамперед надати "найкращі" результати, які є більш досконалими, ніж просто *ранжування у* звичайному тексті. Одним з найвідоміших та найвпливовіших алгоритмів для обчислення відповідності вебсторінок є алгоритм Page Rank, який використовується пошуковою системою Google. Його винайшли Ларрі Пейдж та Сергій Брін, коли вони були аспірантами Стенфорда, і він став товарним знаком Google у 1998 році. Ідея, яку виховував Page Rank, полягала в тому, що про важливість будь-якої веб-сторінки можна судити, переглядаючи сторінки що посилання на нього. Якщо ми створимо веб-сторінку *i* та включимо гіперпосилання на веб-сторінку *j* , це означає, що ми розглядаємо *j*важливим і актуальним для нашої теми. Якщо сторінок, що посилаються на *j,* багато, це означає, що, як вважають, сторінка *j* важлива. Якщо, з іншого боку, *j* має лише одне зворотне посилання, але це надходить з авторитетного сайту *k* (наприклад, www.google.com, www.cnn.com, www.cornell.edu), ми говоримо, що *k* передає свої повноваження *j* ; іншими словами, *k* стверджує, що *j* важливий. Якщо ми говоримо про популярність чи авторитет, ми можемо ітеративно присвоювати рейтинг кожній веб-сторінці, виходячи з рейтингу сторінок, які на неї вказують.

Для цього ми починаємо з того, що зображуємо Веб-мережу у вигляді спрямованого графіка, з вузлами, представленими вебсторінками, а ребра представлені посиланнями між ними.

Припустимо, наприклад, що у нас є невеликий Інтернет, що складається лише з 4 веб-сайтів www.page1.com, www.page2.com, www.page3.com, www.page4.com, які посилаються один на одного у спосіб (рис. 2.1)

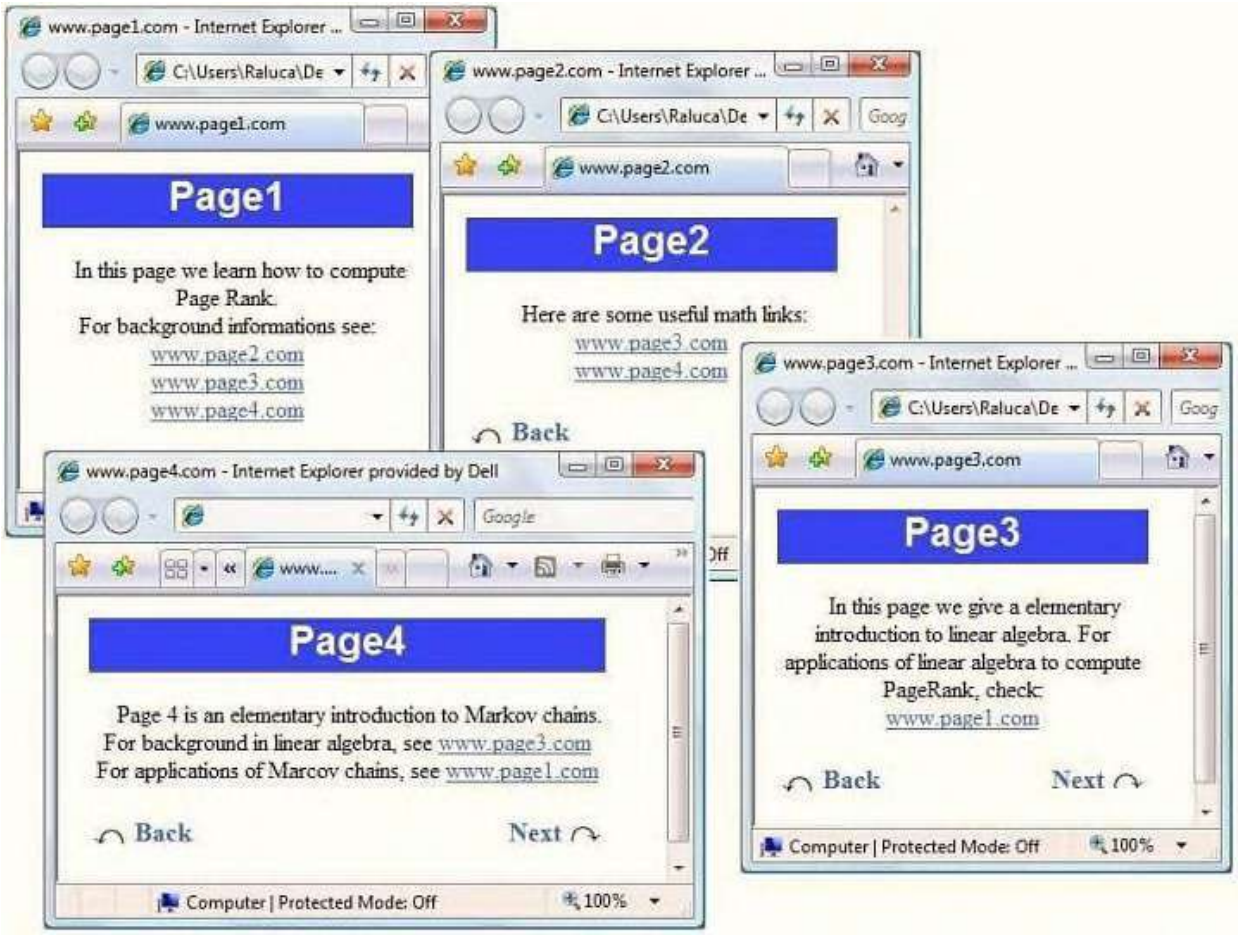


Рисунок 2.1 – Умовне зображення взаємодії вебсторінок

Переробимо рисунок у вигляді орієнтованого графу із 4 вузлами, по одному для кожного веб-сайту. Коли веб-сайт *i* посилається на *j* , ми додаємо орієнтоване ребро між вузлом *i* і вузлом *j* на графі. З метою обчислення їхнього рейтингу сторінки ми ігноруємо будь-які навігаційні посилання, такі як кнопки "Назад", "Наступна", оскільки ми дбаємо лише про зв'язки між різними ресурсами. Наприклад, Page1 посилається на всі інші сторінки, тому вузол 1 на графіку матиме вихідні краї до всіх інших вузлів. Сторінка 3 має лише одне посилання на

Сторінку 1, тому вузол 3 матиме одне вихідне ребро до вузла 1. Після аналізу кожної веб-сторінки ми отримуємо такий граф:

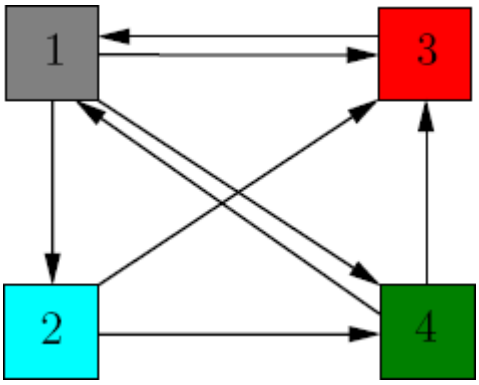


Рисунок 2.2 –Представлення взаємодії вебсторінок у вигляді графу

У нашій моделі кожна сторінка повинна рівномірно передавати своє значення сторінкам, на які вона посилається. Вузол 1 має 3 вихідні ребра, тому він передаватиме $\frac{1}{3}$ свою важливість кожному з інших 3 вузлів. Вузол 3 має лише одне вихідне ребро, тому він передаватиме все своє значення вузлу 1. Загалом, якщо вузол має *k* вихідних ребер, він передаватиме $\frac{1}{k}$ своє значення кожному з вузлів, на які він посилається. Давайте краще візуалізуємо процес, присвоївши ваги кожному ребру.

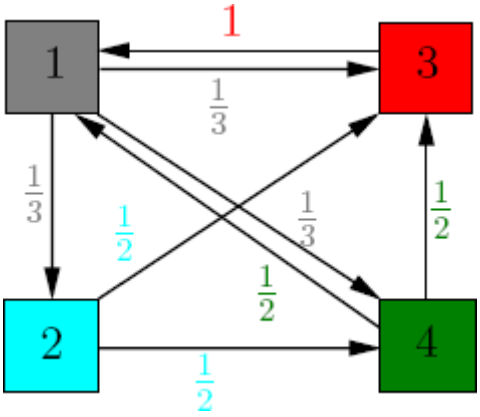


Рисунок 2.3 –Представлення взаємодії вебсторінок у вигляді зваженого графу

Позначимо через *A* матрицю переходів графіка,

$$A = \begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix}.$$

Припустимо, що спочатку значення рівномірно розподіляється між 4 вузлами, кожен отримує ¼. Позначимо через v початковий вектор рангу, маючи всі записи, рівні ¼. Кожне вхідне посилання підвищує важливість веб-сторінки, тому на кроці 1 ми оновлюємо рейтинг кожної сторінки, додаючи до поточного значення важливість вхідних посилань. Це те саме, що помножити матрицю A на v. На кроці 1 новим вектором важливості є $v_1 = Av$. Ми можемо повторити процес, отже, на кроці 2 оновлений вектор важливості має значення $v_2 = A\,(Av) = A^2 v$. Числові обчислення дають:

$$\mathbf{v} = \begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix}, \quad \mathbf{Av} = \begin{pmatrix} 0.37 \\ 0.08 \\ 0.33 \\ 0.20 \end{pmatrix}, \quad \mathbf{A^2 v} = \mathbf{A\,(Av)} = \mathbf{A} \begin{pmatrix} 0.37 \\ 0.08 \\ 0.33 \\ 0.20 \end{pmatrix} = \begin{pmatrix} 0.43 \\ 0.12 \\ 0.27 \\ 0.16 \end{pmatrix}$$

$$\mathbf{A^3 v} = \begin{pmatrix} 0.35 \\ 0.14 \\ 0.29 \\ 0.20 \end{pmatrix}, \quad \mathbf{A^4 v} = \begin{pmatrix} 0.39 \\ 0.11 \\ 0.29 \\ 0.19 \end{pmatrix}, \quad \mathbf{A^5 v} = \begin{pmatrix} 0.39 \\ 0.13 \\ 0.28 \\ 0.19 \end{pmatrix}$$

$$\mathbf{A^6 v} = \begin{pmatrix} 0.38 \\ 0.13 \\ 0.29 \\ 0.19 \end{pmatrix}, \quad \mathbf{A^7 v} = \begin{pmatrix} 0.38 \\ 0.12 \\ 0.29 \\ 0.19 \end{pmatrix}, \quad \mathbf{A^8 v} = \begin{pmatrix} 0.38 \\ 0.12 \\ 0.29 \\ 0.19 \end{pmatrix}$$

Ми помічаємо, що послідовності ітерацій v, Av, ..., $A^k v$ прагнуть до рівня рівноваги $v^* = \begin{pmatrix} 0.38 \\ 0.12 \\ 0.29 \\ 0.19 \end{pmatrix}$.

Це називається вектором PageRank нашого вебграфу.

Позначимо через x_1, x_2, x_3 та x_4 важливість чотирьох сторінок. Аналізуючи ситуацію на кожному вузлі, отримуємо систему:

$$\begin{cases} x_1 = 1 \cdot x_3 + \frac{1}{2} \cdot x_4 \\ x_2 = \frac{1}{3} \cdot x_1 \\ x_3 = \frac{1}{3} \cdot x_1 + \frac{1}{2} \cdot x_2 + \frac{1}{2} \cdot x_4 \\ x_4 = \frac{1}{3} \cdot x_1 + \frac{1}{2} \cdot x_2 \end{cases}$$

Це рівнозначно запитуванню про рішення рівнянь $A \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$.

Так як, власні вектори, що відповідають власному значенню 1, мають вигляд $c \cdot \begin{bmatrix} 12 \\ 4 \\ 9 \\ 6 \end{bmatrix}$. Оскільки PageRank повинен відображати лише відносну важливість вузлів, і власні вектори є просто скалярними кратними один одному, ми можемо вибрати будь-який з них для нашого вектора PageRank. Виберем v^*, щоб бути унікальним власним вектором із сумою всіх записів, рівною 1. (Іноді ми називатимемо його імовірнісним власним вектором, що відповідає власному значенню 1

Власний вектор $\frac{1}{31} \cdot \begin{bmatrix} 12 \\ 4 \\ 9 \\ 6 \end{bmatrix} \sim \begin{bmatrix} 0.38 \\ 0.12 \\ 0.29 \\ 0.19 \end{bmatrix}$ - це наш PageRank вектор.

З очки зору теорії імовірності, скільки важливість веб-сторінки вимірюється її популярністю (скільки вхідних посилань у неї є), ми можемо розглядати важливість сторінки і як ймовірність того, що випадковий серфер в Інтернеті, який відкриває браузер для будь-якої сторінки і починає слідувати за гіперпосиланнями та відвідує сторінку. Ми можемо інтерпретувати ваги, присвоєні краям графіка, імовірнісним чином: випадковий серфер, який зараз переглядає веб-сторінку 2, має ½ ймовірність перейти на сторінку 3 і ½ ймовірність перейти на сторінку 4. Ми можемо моделювати процес у вигляді випадкової прогулянки по графіках. Кожна сторінка має однакову ймовірність ¼ бути обраною як вихідну точку. Отже, початковий розподіл ймовірностей задається вектором стовпця $[¼\ ¼\ ¼\ ¼]^t$. Імовірність того, що сторінку i відвідають після k кроків, дорівнює $A^k x$. Послідовність $Ax, A^2x, A^3x, ..., A^kx, ...$ збігається в цьому випадку до унікального імовірнісного вектора v^*. У цьому контексті v^* називається стаціонарним розподілом, і він буде нашим вектором Page Rank. Більше того, i-й запис у вектор

v^* - це просто ймовірність того, що кожного моменту випадковий серфер відвідує сторінку i . Обчислення ідентичні тим, які ми робили в інтерпретації динамічних систем, лише значення, яке ми надаємо кожному кроку, дещо відрізняється.

Вектор Page Rank $v^{*, \text{який}}$ ми розрахували різними методами, вказує на те, що сторінка 1 є найбільш релевантною. Це може здатися дивним, оскільки сторінка 1 має 2 зворотні посилання, тоді як сторінка 3 має 3 зворотні посилання. Якщо ми подивимося на графік, то побачимо, що вузол 3 має лише один вихідний край на вузол 1, тому він передає все своє значення на вузол 1. Аналогічно, як тільки веб-серфер, який лише слідкує за гіперпосиланнями, відвідує сторінку 3, він може лише перейдіть на сторінку 1. Зверніть увагу також, як ранг кожної сторінки не є тривіально просто зваженою сумою ребер, що входять у вузол. Інтуїтивно, на кроці 1, один вузол отримує підтвердження важливості від своїх безпосередніх сусідів, на кроці 2 від сусідів своїх сусідів тощо.

Зміна веб-графіка може призвести до певних проблем.

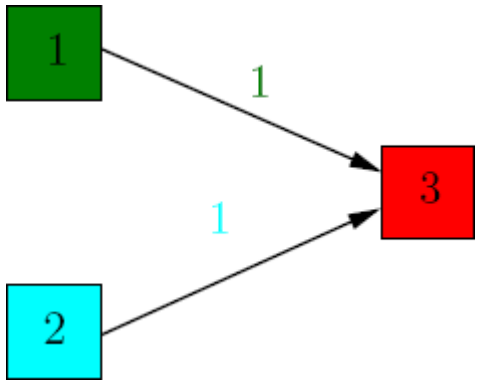


Рисунок 2.4 – Вузли без вихідних країв (звисаючі вузли)

Ми ітеративно обчислюємо ранг 3 сторінок:

$$v_0 = \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{bmatrix}, \quad v_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{2}{3} \end{bmatrix}, \quad v_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ \frac{2}{3} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Отже, у цьому випадку ранг кожної сторінки дорівнює 0. Це протиінтуїтивно, оскільки сторінка 3 має 2 вхідні посилання, тому вона повинна мати певне значення!

Легким виправленням цієї проблеми було б замінити стовпець, що відповідає звисаючому вузлу 3, на вектор стовпця з усіма записами 1/3. Таким чином, важливість вузла 3 буде однаково перерозподілена серед інших вузлів графіка, замість того, щоб бути втраченою.

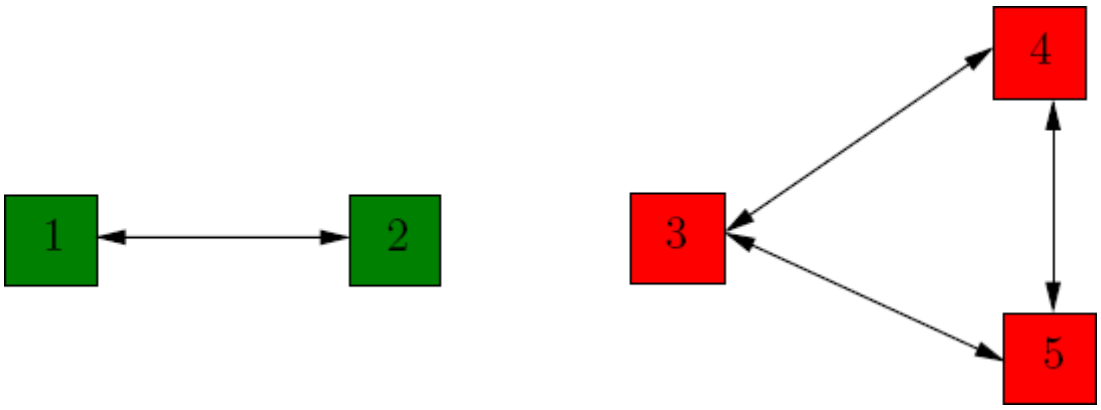


Рисунок 2.5 – Відключені компоненти

Випадковий серфер, який починається з першого підключеного компонента, ніяк не може потрапити на веб-сторінку 5, оскільки вузли 1 і 2 не мають посилань на вузол 5, за якими він може слідувати. Лінійна алгебра також не допомагає.

Матриця переходу для цього графа $A = \left[\begin{array}{cc|ccc} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \end{array}\right]$. Зверніть увагу, що $v = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$, $u = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$ обидва власні вектори відповідають власному значенню 1, і вони не просто тривіально один - скалярне кратне іншому. Отже, як в теорії, так і на практиці, позначення сторінок ранжування з першого підключеного компонента щодо тих, що належать до другого підключеного компонента, є неоднозначним.

Мережа дуже неоднорідна за своєю природою і, безумовно, величезна, тому ми не очікуємо, що її графік буде пов’язаний. Подібним чином будуть сторінки, які мають просто описовий характер і не містять вихідних посилань. Нам потрібно неоднозначне значення рангу сторінки для будь-якого спрямованого веб-графіка з *n* вузлами.

Пейдж та Брін запропонували для того, щоб подолати ці проблеми, зафіксувати позитивну константу *p* між 0 і 1, яку ми називаємо коефіцієнтом демпфування (типове значення *p* становить 0,15). Визначте матрицю Page Rank (також відому як матриця Google) графу

$$M = (1-p)\cdot A + \ p\cdot B$$

де.

$$B = \frac{1}{n}\cdot\begin{bmatrix} 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix}$$

2.2 Вдосконалена математична модель ранжування сторінок

Ранній алгоритм Google базувався на теорії, згідно з якою посилання з одного веб-сайту на інший діяло як вотум довіри та повноважень. І, отже, чим більше посилань (голосів) вказує на сторінку, тим більше їй слід довіряти і, отже, посідати вище місце.

Але, як визначено в оригінальній статті, "PageRank поширює цю ідею, не рахуючи однаково посилання з усіх сторінок і нормалізуючи кількість посилань на сторінці".

Посилання - це не просто пряме голосування. Враховується повноваження сторінки. Посилання зі сторінки PageRank 6, зрештою, є більш авторитетним голосуванням, ніж посилання зі сторінки PageRank

Тому після попередніх розрахунків, статистики та аналізу принципу дії PageRank можна створити математичну модель. Це буде виглядати так:

$$PR(A) = (1-d) + d\left(\frac{PR(T_1)}{C(T_1)} + \frac{PR(T_n)}{C(T_n)}\right) \quad (2.1)$$

де: $PR(A)$ - ваговий коефіцієнт сторінки A,

D - коефіцієнт затухання який Google пропонує встановити рівним 0,85,

$PR(T_1)$ - ваговий коефіцієнт PageRank сторінки, що посилається на сторінку A,

$C(T_1)$ - число посилань із цього ресурсу,

$\frac{PR(T_1)}{C(T_n)}$ - для кожного ресурсу який вказує на ресурс A.

Для обрахунку рейтингу сторінки мережа інтернет представляється у вигляді орієнтованого графіка, вершини якого відповідають вебресурсам, а ребра - посиланням між ними.

Припустимо, що до пошукового індексу буде додано n вебресурсів. Потім створюється матриця переходів розміру M для моделювання випадкового серфінгу. Елемент цієї матриці, який знаходиться в рядку i та стовпці j, важливий, якщо сторінка з номером j має k оригінальних гіперпосилань, серед яких є одне на сторінці з номером i. Якщо таке посилання відсутнє, то елемент має значення 0 [27].

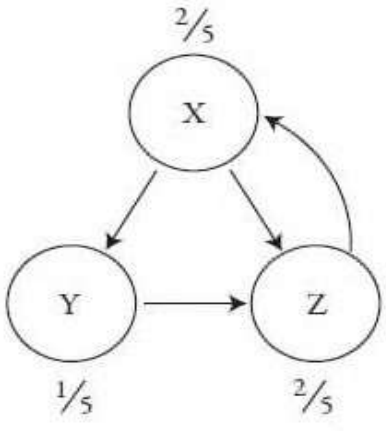


Рисунок 2.6 – Приклад орієнтованого графу мережі з трьох сторінок

Проте, через те, що орієнтований граф посилань реальної мережі інтернет не завжди зв'язний та має тупикові вузли, потрібно врахувати можливість випадкового «стрибка» на іншу вершину. Таким чином, для обрахування розподілів знаходження скористаємося формулою:

$$v' = \beta M_v + (1-\beta)\frac{e}{n}$$

де:

β - константа, яка має значення в діапазоні 0.8…0.9 ;

e - вектор, всі елементи якого дорівнюють 1;

n - кількість вершин графу (індексованих вебресурсів);

βM_v - користувач з ймовірністю β вирішує обрати вихідне посилання з поточного вебресурсу;

$(1-\beta)\frac{e}{n}$ - вектор, кожен член якого дорівнює $\frac{(1-\beta)}{m}$ і який прогнозує появу користувача на будь-якому ресурсі з ймовірністю $(1-\beta)$.

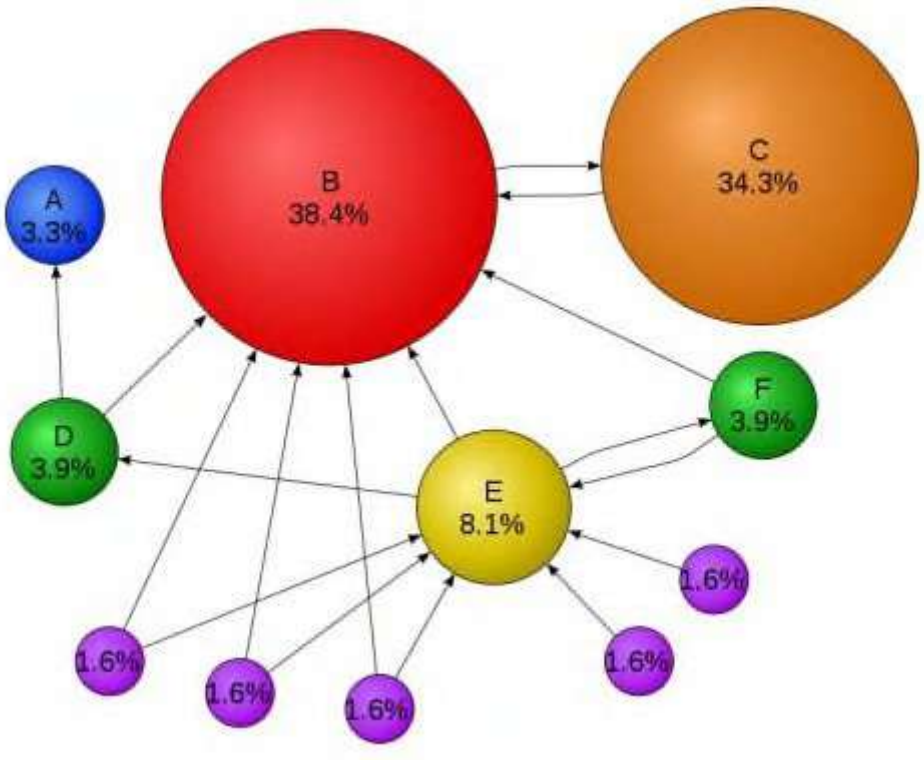


Рисунок 2.7 – Базовий принцип роботи PageRank

Як видно з моделювання основного принципу роботи - рейтинг будь-якої сторінки пов’язаний і формується на основі інших сторінок, навіть якщо вони безпосередньо з ним не пов’язані. Сторінка В має найвищий рейтинг, оскільки з нею пов’язано багато вторинних ресурсів [12,13].

Модель функціонування алгоритму ранжування зображено на рисунку 2.5.

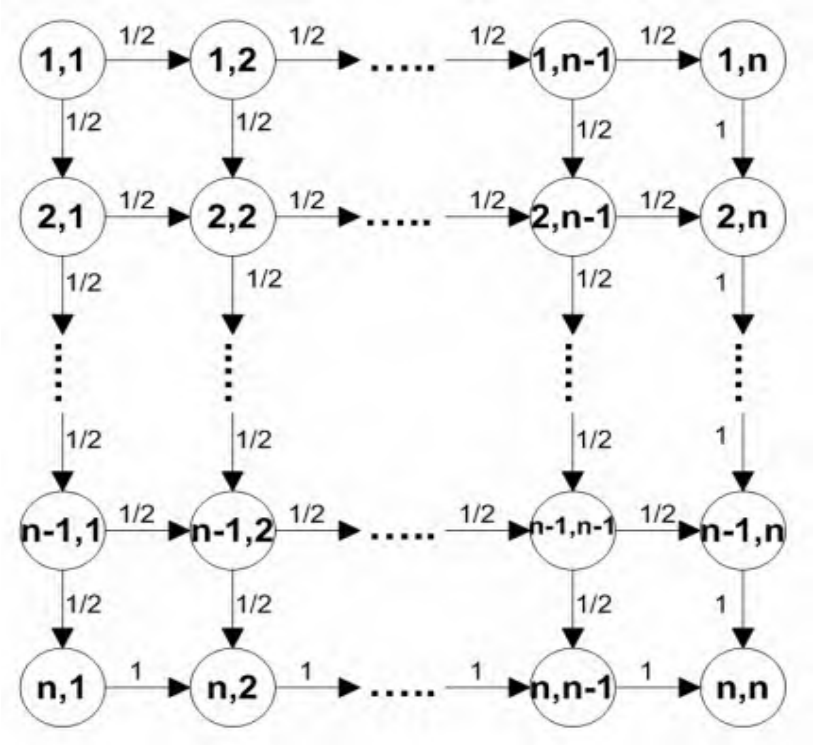


Рисунок 2.8 – Матрична модель функціонування алгоритму PageRank

Кількість вихідних посилань з батьківського ресурсу має велике значення для розрахунку рейтингу сайту потомка. Goocle допускає, що один вебресурс може передати максимум 85% свого вагового коефіцієнту, тоді як її рейтинг не зменшується. Але якщо є 2 посилання, вага між ними ділиться навпіл, і.т.д. [16].

Автором розроблено модифіковану математична модель, яка враховує популярність ресурсу в соціальних спільнотах по тематиці запиту:

$$PR(A) = (1-d) + d\left(\frac{PR(T_1)}{C(T_1)} + \ldots + \frac{PR(T_n)}{C(T_n)}\right) + s \times T(C)_s \qquad (2.2)$$

де: $PR(A)$ - ваговий коефіцієнт сторінки A,

D - коефіцієнт затухання який Google пропонує встановити рівним 0,85,

$PR(T_1)$ - ваговий коефіцієнт PageRank ресурсу що посилається на сторінку A,

$C(T_1)$ - кількість посилань із цього ресурсу,

$\frac{PR(T_1)}{C(T_n)}$ - для кожного ресурсу який вказує на ресурс A.

$T(C_s)$ – кількість посилань в соціальних групах,

s – коефіцієнт ModPR, який пропонується обрати рівним 0.15

2.3 Висновки

Функція пошуку ключових слів від Google подібна до інших пошукових систем. Автоматизовані програми, звані павуками або сканерами**,** подорожують Інтернетом, переходячи від посилання до посилання та створюючи індексну сторінку, що включає певні ключові слова. У пошуковій системі перелічені сторінки, що містять ті самі ключові слова, що були в пошукових термінах користувача. Павуки Google можуть також мати деякі більш розширені функції, такі як можливість визначати різницю між вебсторінками з фактичним вмістом та переспрямовуючими сайтами - сторінками, які існують лише для перенаправлення трафіку на іншу веб-сторінку.

Проведені дослідження показали, що поведінкові фактори та штучні маніпуляції здійснюють істотний вплив на позиції та рейтинг сайту в пошукових системах.

Для вдосконалення ранжування та індексації сторінок пропонується включити в математичну модель розрахунку рангу коефіцієнт впливу сторінки соціальних груп та форумів.

3 МЕТОД ПІДВИЩЕННЯ ПЕРТИНЕНТНОСТІ РЕЗУЛЬТАТУ ПОШУКУ

3.1 Вдосконалений метод індексації ресурсів

PageRank названий на честь співзасновника Google Ларрі Пейджа і використовується для ранжирування вебресурсів у результатах пошуку Google. Він підраховує кількість та якість посилань на сторінку, що визначає оцінку важливості сторінки. Основне припущення полягає в тому, що важливі сторінки частіше отримують більший обсяг посилань з інших сторінок.

PageRank можна застосовувати в широкому діапазоні доменів. Нижче наведено кілька відомих випадків використання:

- Персоналізований PageRank використовується Twitter для надання користувачам рекомендацій щодо інших облікових записів, за якими вони можуть побажати підписатися. Алгоритм запускається на графі, який містить спільні інтереси та спільні зв’язки.
- PageRank використовувався для ранжирування громадських приміщень або вулиць, прогнозуючи рух транспорту та рух людей у цих районах. Алгоритм запускається на графіку, що містить перехрестя, з’єднані дорогами, де оцінка PageRank відображає тенденцію людей припаркуватися або закінчити свою подорож на кожній вулиці.
- PageRank можна використовувати як частину системи виявлення аномалій або шахрайства у сферах охорони здоров’я та страхування. Це може допомогти знайти лікарів або постачальників, які поводяться незвично, а потім подати оцінку в алгоритм машинного навчання.

Обмеження - коли не використовувати алгоритм PageRank.

Є декілька речей, про які слід пам’ятати, використовуючи алгоритм PageRank:

- Якщо всередині групи сторінок немає посилань на інші сторони групи, то ця група вважається павутиною.

- Зниження рейтингу може статися, коли мережа сторінок утворює нескінченний цикл.
- Тупикові ситуації трапляються, коли сторінки не мають вихідних посилань. Якщо сторінка містить посилання на іншу сторінку, яка не має вихідних посилань, посилання буде називатися звисаючим посиланням.

Якщо ви бачите несподівані результати від запуску алгоритму, варто провести дослідницький аналіз графу, щоб з'ясувати, чи є причиною якась із цих проблем.

Вага посилання - це ступінь впливу кожного посилання на індексування сторінки пошуковою системою.

Кількість вихідних посилань з батьківського ресурсу має велике значення для розрахунку рейтингу сайту потомка. Goocle допускає, що один вебресурс може передати максимум 85% свого вагового коефіцієнту, тоді як її рейтинг не зменшується. Але якщо є 2 посилання, вага між ними ділиться навпіл, і.т.д.

Розглянемо приклад реалізації PageRank.

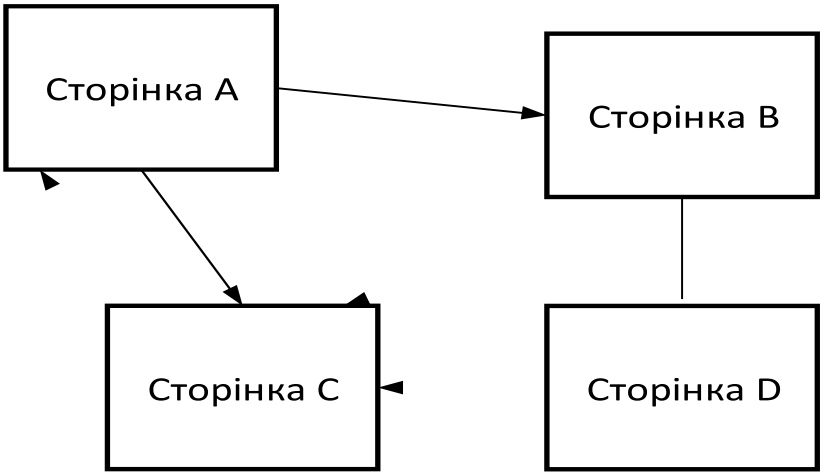


Рисунок 3.1 – Початкова діаграма взаємозв'язку сторінок

Присвоїмо початковий ваговий коефіцієнт вебресурсам. Отримаємо наступну діаграму (рис. 3.1).

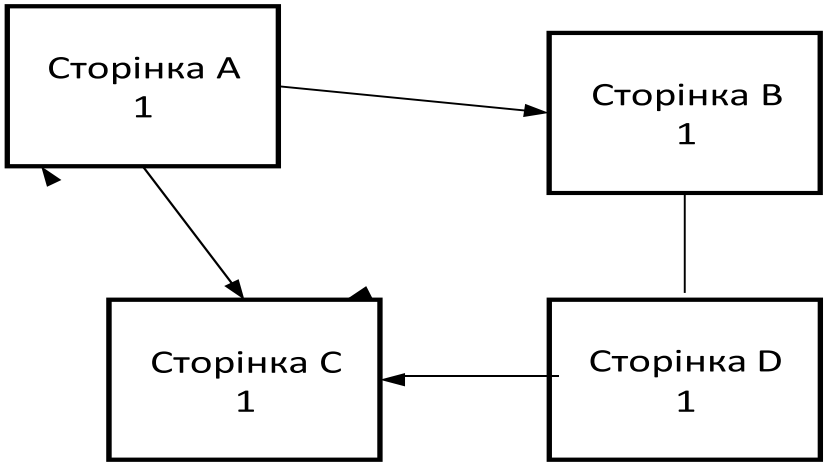


Рисунок 3.2 – Діаграма взаємозв’язку, після присвоєння ваги

Спочатку застосовуємо коефіцієнт загасання. (Коефіцієнт загасання, як правило, вказує на те, що сторінка не може голосувати, щоб інша сторінка стала такою ж важливою, як і вона.) Тоді відбувається ділення ваги сторінки на кількість посилань. Ми розраховуємо остаточну вагу, яку слід додати на всі сторінки, перш ніж ми зможемо остаточно додати її.

Отже, дивлячись спочатку на ресурс A, бачимо, що значення вагового коефіцієнту, доступне для передачі, після затухання дорівнює 1 × 0,85 = 0,85. Зі сторінки йде два гіперпосилання, тому, після закінчення цього кроку, ми додамо 0,425(0,85 ÷ 2) до вагового коефіцієнту ресурсу B і 0,425 до вагового коефіцієнту ресурсу. Але для цього нам спочатку потрібно розрахували все посилання сторінки, тому що це вплине на кінцевий результат [12,13]..

Розглянемо ресурс B. Він містить тільки одне гіперпосилання. Тому, він зможе надати 1 × 0,85 = 0,85 ресурсу C, коли буде проведено всі розрахунки для посилань.

Ресурс C також має одне посилання. Тому вона надасть ваговий коефіцієнт 1 × 0,85 = 0,85 ресурсу A.

Ресурс D має одне гіперпосилання, тому він надасть ваговий коефіцієнт 0,85 ресурсу C.

Зараз ми можемо перерахувати вагові коефіцієнти кожного ресурсу.

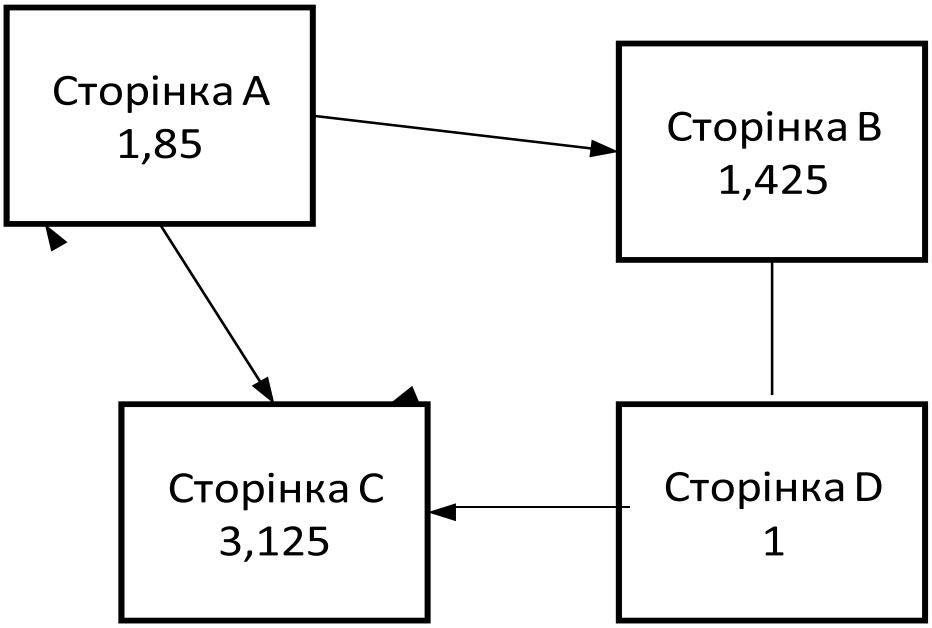


Рисунок 3.3 – Діаграма рейтингу на другому кроці

Отримані значення вагових коефіцієнт демонструють, наскільки важливий ресурс C. Але процес не завершено. Наразі враховано тільки популярність в посиланнях (link popularity). Суть ранжування така, що ресурс, на який найчастіше посилаються, має отримати більше голосів; тому виконаємо попередні дії ще раз! На цей раз ресурс C має більший вплив, тому що її поточний рейтинг вище.

Звернемося до ресурсу A. Його поточний ваговий коефіцієнтдорівнює 1,85. Величина ранжування, доступна для передачі, після застосування затухання дорівнює 1,85 × 0,85 = 1,5725. Так, як є два гіперпосилання з ресурсу, тому по завершенню цього кроку ми додамо 0,78625 до вагового коефіцієнту ресурсу B і вазі рейтингу ресурсу C. Перейдемо до ресурсу B. У нього є тільки одне посилання. Отже, вона надасть 1,425 × 0,85 = 1,21125 вагового коефіцієнту ресурсу C, після завершення всіх обрахунків з посиланнями.

Ресурс C також має одне гіперпосилання, але при цьому має великий рейтинг 3,125. Тому віна надасть 3,125 × 0,85 = 2,65625 ресурсу A [12,13]..

Так, як ресурс D має одне посилання, тому він лоласть до рейтингу ресурсу C 0,85.

Що таке поведінкові фактори формування рейтингу Google і для чого за ними варто слідкувати [14]?

Поведінкові фактори ((ПФ)) - це показники, які базуються на діях користувача щодо певного сайту, його сторінок і враховуються пошуковими системами при ранжуванні та в процесі отримання результатів для ключових запитів.

Таким чином, покращення поведінкових характеристик відіграє важливу роль для будь-якого типу вебсайту, особливо, коли мова йде про Інтернет-магазини та інші комерційні проекти.

Перш за все, інформація про поведінку відвідувачів збирається пошуковими системами шляхом аналізу людських дій у процесі взаємодії з пошуковою системою. Але, крім того, ми використовуємо дані, отримані із систем веб-аналітики (Google Analytics) через наші власні браузери (Google Chrome) та інші технічні служби (наприклад, Google Search Console).

Варто зазначити, що збір даних про поведінкові фактори, їх аналіз і оновлення пошуковими системами здійснюється неодноразово. Тобто роль і вплив поведінкових факторів на результати пошукових систем змінюються так само швидко, як і технічний прогрес у всьому світі.

Також існує продукт для SEO - це SEO-сповіщення, ви можете автоматично отримувати відповідну інформацію щодня про семантику та SEO-аудит.

Збираючи інформацію з усіх цих джерел, пошукова система Google отримує вичерпну інформацію про те, як поводяться відвідувачі сайту, та враховує дані поведінкових факторів при формуванні результатів пошуку.

Насправді все дуже просто. Вплив поведінкових характеристик на просування сайту в Google очевидний, оскільки зі зростанням можливостей в Інтернеті кількість конкуренції постійно зростає і, відповідно, збільшується можливість вибору одного і того ж товару в різних компаніях. Щоб зробити ТОП 3 - ТОП 10 найкращими з кращих, пошукова система Google вирішила, що не лише якість продуктів чи наявність SEO-текстів допоможуть пробитися в ряди слави. Іншим важливим фактором рейтингу повинні бути поведінкові фактори користувачів на сайті. Зрештою, якщо користувач не знайшов необхідної

інформації, товару чи послуги на сторінках вашого сайту, не зміг зробити замовлення через незручний сервіс - пошукова система Google автоматично дозволить вам покращити всі показники, але на 2-й або 5-й сторінці результатів пошуку

SEO-спеціалісти виділяють 2 типи ранжування поведінкових факторів сайту в Google:

1. "Добре" - поведінкові фактори ранжування сайту в Google, які позитивно впливають на зростання позиції вашого сайту та загальне його просування. Ці "хороші" поведінкові фактори ранжування сайтів слід постійно вдосконалювати та контролювати їх ефективність;

2. "Погано" - поведінкові фактори ранжування сайту в Google, які негативно впливають на просування вашого сайту та покращують продажі в Інтернеті. За цими поведінковими факторами слід ретельно стежити та при найменшому погіршенні аналізувати причину.

3.2 Модифікований метод пошуку інформації ModPR

Новизна модифікованого алгоритму полягає в тому, що застосовується нова формула для розрахунку рангу ресурсу - ModPR. Введено додатковий коефіцієнт, який популярність того самого ресурсу на форумах та в соціальних групах. Це дозволить пошуковій системі оцінити певний ресурс не тільки за кількістю посилань з інших ресурсів на нього, але і врахувати його популярність на форумах та в соціальних групах.

Для отримання пертинентних результати пошуку в Інтернеті та формування реальних рейтингів вебресурсу, PageRank буде модифікований іншим фактором, який буде безпосередньо враховувати популярність сайту в соціальних спільнотах та його згадувань на форумах.

Додамо у математичну модель ранжування PageRank змінну s – коефіцієнт який буде рівний 0,15, для врахування кількості посилань з форумів та соціальних спільнот.

Розглянемо промодельовану в 2 розділі мережу (рис 2.4)). Роботу алгоритму ранжування ModPR продемонстровано на рис. 3.4.

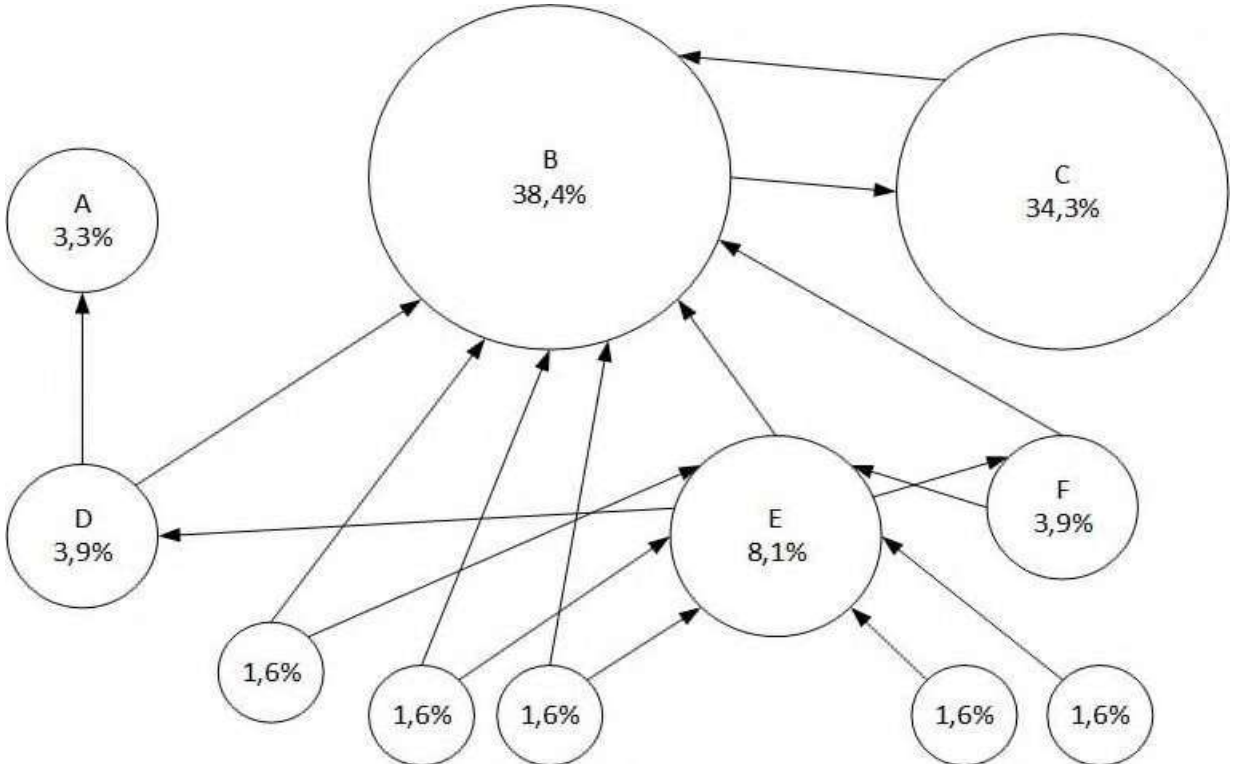


Рисунок 3.4 – Ранжування ресурсів до модифікації

Ресурс С має більш високий ранг, ніж ресурс Е, хоча є менше посилань на С, ніж на Е, але одна з гіперпосилань на С відбувається з вагомішого ресурсу і, отже, має більш високий ранг. Якщо припустити, що умовний користувач, який знаходиться на випадковому ресурсі, має 85% ймовірність якогось гіперпосилання на поточному ресурсі, і 15% переходу на будь-який інший ресурс, то ймовірності відвідування ресурсу Е з інших гіперпосилань дорівнює 15%*0.85 =8,1. Без загасання користувачі в кінцевому підсумку потрапляють на ресурси А, В або С, і всі інші ресурси будуть мати рейтинг, рівний нулю [12,13].. Модель модифікованого алгоритму ранжування зображено на рис. 3.5.

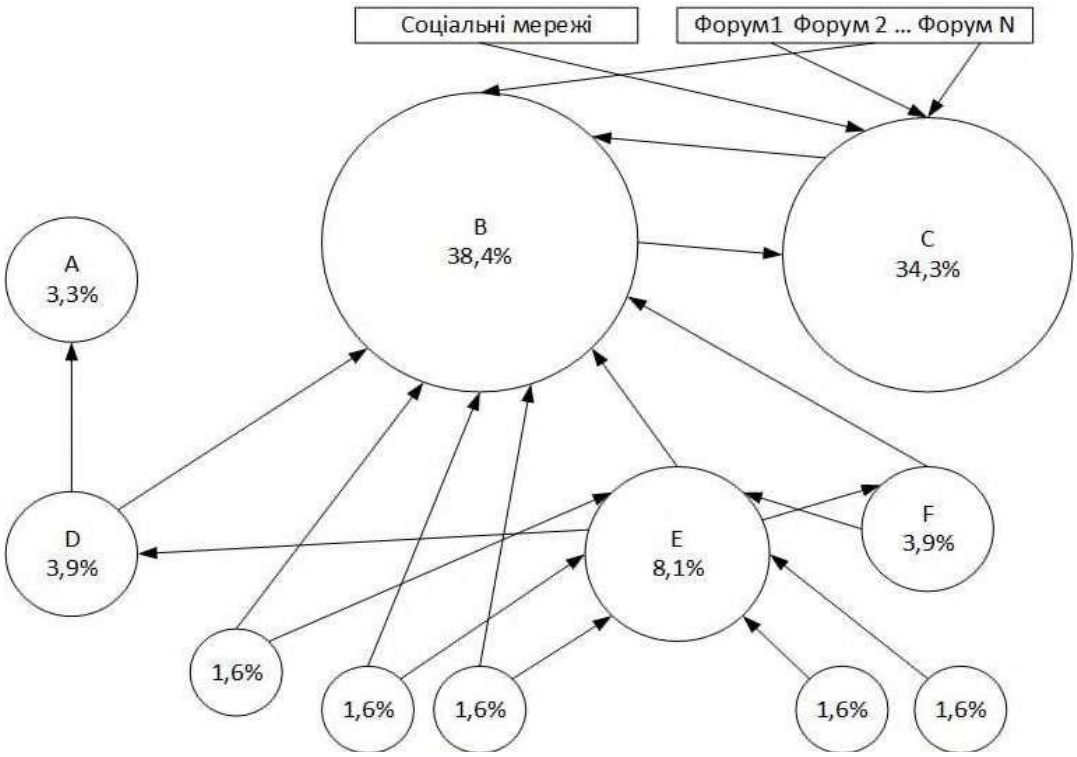


Рисунок 3.5 – Схема вираховування рейтингу ресурсу модифікованим алгоритмом

Як видно з рисунку 3.5, після впровадження коефіцієнта *s* реальний рейтинг ресурсу B, дещо впаде, оскільки він не настільки популярний на форумах та в соціальних групах як сайт C. До рейтингу сайту C додається 0,45:

$$s \times T(C_s) = 3 \times 0{,}15 = 0{,}45$$

Відповідно до загальної рейтингу ресурсу B додамо:

$$s \times T(C_s) = 1 \times 0{,}15 = 0{,}15$$

В такому випадку ресурси B та C стануть рівними за рейтингом, а отже, відповідно до розробленого модифікованого алгоритму, перевага у списку видачі результатів пошуку піде на ресурс C.

За такого розвитку подій, коли на форумах та соцмережах ресурс С рекомендують частіше та більше ніж ресурс В, він буде мати вищий рейтинг, та буде розташаваний вище у видачі сторінок у відповідь на запит користувача.

Модель функціонування алгоритму ранжування зображено на рисунку 3.6.

.

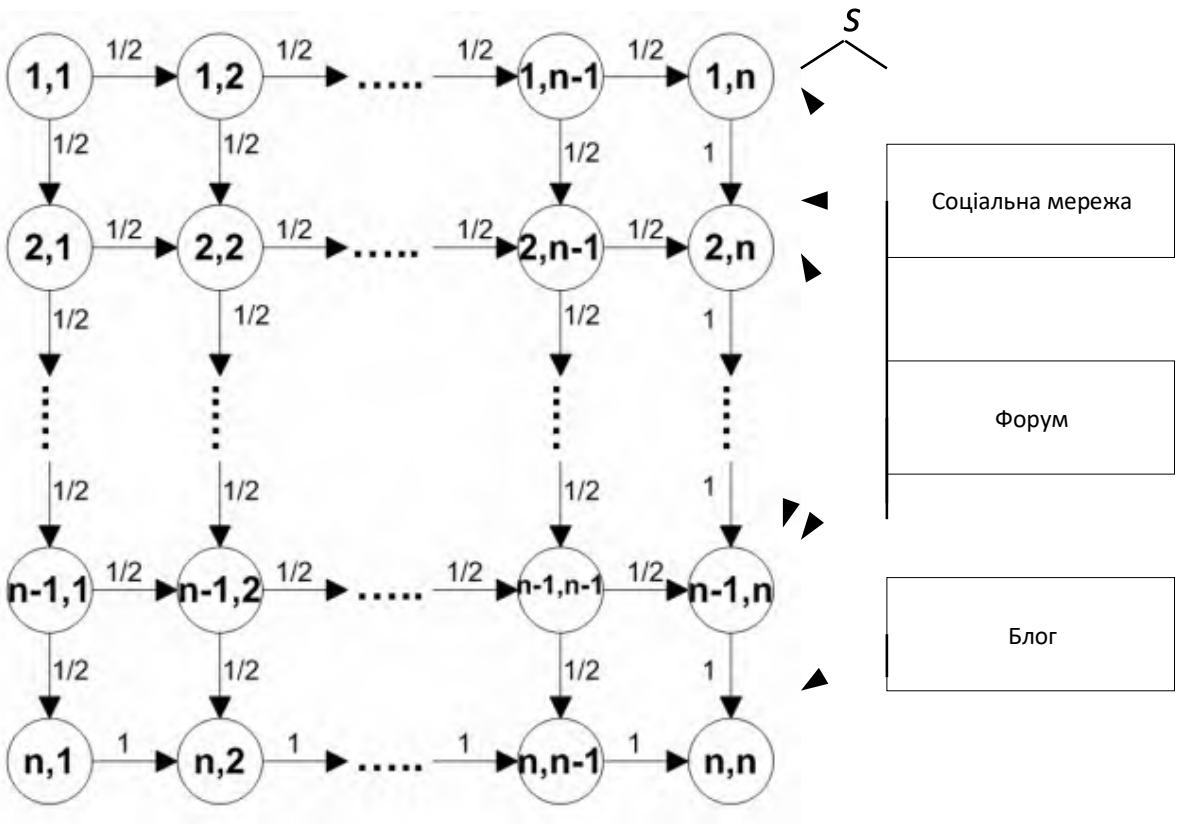


Рисунок 3.6 – Матрична модель алгоритму ранжування ModPR

Кількість вихідних посилань з батьківського ресурсу має велике значення для розрахунку рейтингу сайту потомка. Goocle допускає, що один вебресурс може передати максимум 85% свого вагового коефіцієнту, тоді як її рейтинг не зменшується.

Модифікований алгоритм також враховує рейтинг кожної сторінки, що отримала голос, адже рейтинги деяких сторінок є важливішими, і їх популярність в соціальних групах.

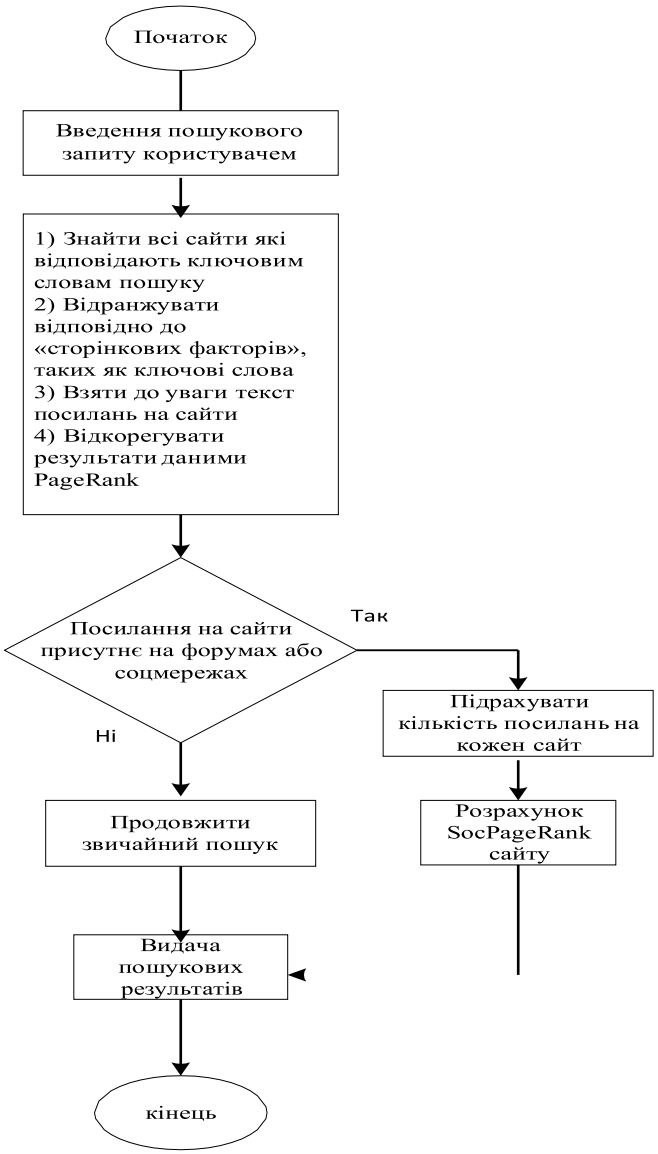


Рисунок 3.7 – Блок-схема алгоритму ModPR

Особливістю модифікованого алгоритму нова формула розрахунку ваги сайту - ModPR. Тут враховується додатковий ваговий коефіцієнт популярності сайту на форумах та в соціальних спільнотах.

Алгоритм отримання потрібних та корисних гіперпосилань з соціальних груп буде вирішено в наступному розділі.

3.3 Висновки

Традиційні способи пошуку відповідних сторінок у випадку односкладових запитів не дають задовільних результатів, оскільки популярні теми завжди знайдуть велику кількість сторінок з однаковою актуальністю. Для того, щоб якось

упорядкувати такі сторінки, пошукові системи вдаються до різних інструментів Для цього Google використовує PageRank, що дає приголомшливі результати.

Кількість вихідних посилань з батьківського ресурсу має велике значення для розрахунку рейтингу сайту потомка. Goocle допускає, що один вебресурс може передати максимум 85% свого вагового коефіцієнту, тоді як її рейтинг не зменшується.

У розділі розглянуто функціонування класичного PageRank запропоновано підхід до його модифікації.

За допомогою вдосконаленого алгоритму ранжування ModPR, корисні, актуальні та малопопулярні сайти які часто обговорюють корстувачі в тематичних соціальних групах матимуть змогу отримувати свій реальний рейтинг.

4 ЗАСТОСУВАННЯ МОДИФіКОВАНОГО МЕТОДУ ПОШУКУ MODPR

4.1 Алгоритми пошуку спільнот у соціальних мережах

Особливістю модифікованого алгоритму нова формула розрахунку ваги сайту - ModPR. Тут враховується додатковий ваговий коефіцієнт популярності сайту на форумах та соціальних мережах.

Пошук сторінок груп здійснюється інструментами соціальних мереж за назвами форумів або за їх коротким змістом. Але це завжди відповідає інформаційному наповненню цих документів.

Як результат, виникла проблема, пов’язана з пошуком необхідної інформації на сторінках груп у соціальних мережах, що значно ускладнюється необхідністю пошуку відповідно до змісту та релевантності сторінки обговорення з урахуванням функціонування сторінки для обговорення в соціальних мережах, а саме [24]:

- сторінки мають низький рейтинг в алгоритмах ранжування сторінок;
- велика кількість вебсторінок для обговорення не класифікуються глобальними пошуковими системами;
- взаємозв’язок вебсторінок для обговорення;
- збереження форумів неактуальної тематичної спрямованості.

Запит глобальної пошукової системи для пошуку відповідних обговорень складається з таких операцій:

- операція локалізації пошукового запиту;
- операція виявлення інформаційного вмісту;
- обмеження часу.

Загальна структура запиту наведена на рис. 4.1

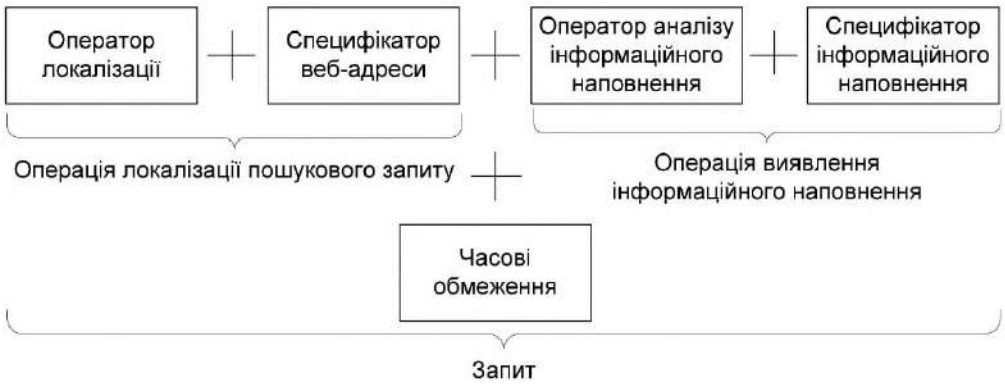


Рисунок 4.1 - Формалізований запит для виявлення повязаних дискусій вебсторінок

Операція локалізації пошукового запиту дозволяє обмежити пошук ресурсом або доменом. Ця операція складається із команди локалізації та специфікатора адреси ресурсу.

Оператор аналізу вмісту - оператори послуг глобальної пошукової системи, які аналізують вміст вебсторінки за заданою темою. Ми будемо розглядати слово чи фразу, що містяться в тілі вебсторінки, як інформаційний зміст сторінки.

Специфікатор вмісту - це функція, яка характеризує об’єкт пошуку. Ознака інформаційного змісту - це слово або фраза в інформаційному вмісті веб-сторінки.

Соціальна мережа Facebook дозволяє користувачам створювати власні спільноти та групові сторінки, в яких вони можуть об’єднуватися відповідно до інтересів. Група - це спільнота, в якій люди мають можливість спілкуватися між собою. Однією з головних особливостей спільнот є те, що вони ***не індексуються*** глобальними пошуковими системами. На відміну від спільнот, сторінки індексуються глобальними пошуковими системами.

Отже, пошук спільнот ведемо за допомогою пошуку сторінок соціальній мережі «Facebook», використовуючи формалізоване звернення (рис. 3.1), наведений на рис. 3.2, з подальшим аналізом HTML-коду сторінки.

Рисунок 4.2 - Формалізований запит для виявлення груп у соціальній мережі «Facebook»

Формалізатор сторінки «спільнота» вноситься до формалізованого запиту на пошук сторінок у соціальній мережі «Facebook» в операції пошуку інформаційного контенту. Пошук здійснюється за специфікатором інформаційного змісту (Ключові слова) - набором ключових слів. Сторінки у "Facebook" чітко структуровані, і для визначення присутності відповідних обговорень ми аналізуємо HTML-код на наявність пункту меню "групи" (рис. 4.3).

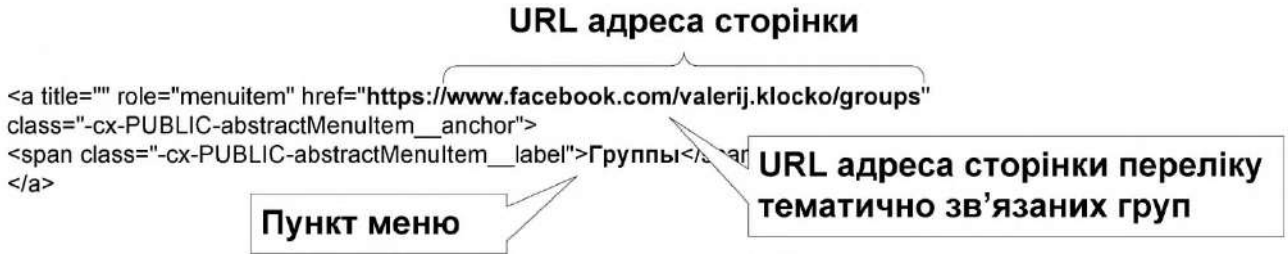


Рисунок 4.3 - Дослідження HTML-коду сайту для виявлення URL-адреси ресурсу переліку тематично зв'язаних груп

За результатами аналізу формуємо URL-адресу тематично зв'язаних спільнот.

Схематичне зображення алгоритму пошуку відображено на рис. 3.4.

Рисунок 4.4 - Алгоритми пошуку спільнот у соціальних мережах

Розглянутий алгоритм (рис. 4.4 використовує формалізовані запити глобальної пошукової системи Google та аналіз HTML-коду сторінок обговорень у соціальних мережах Це дозволяє шукати їх відповідно до їх змісту.

Потреба в поглибленому пошуку зумовлена як технічними, так і організаційними причинами, а саме: велика кількість сторінок для обговорень у соціальних мережах не індексуються глобальними пошуковими системами, є певні особливості сторінок для обговорень у соціальних мережах, тобто взаємозв'язок сторінки обговорення.

Поглиблений пошук реалізується шляхом розробки пошукової системи, входом якої є аналіз списку контактів у структурі сторінки обговорення та гіперпосилань в інформаційному змісті обговорень, знайдених глобальною пошуковою системою Google за допомогою API-методів запитів соціальних мереж [25].

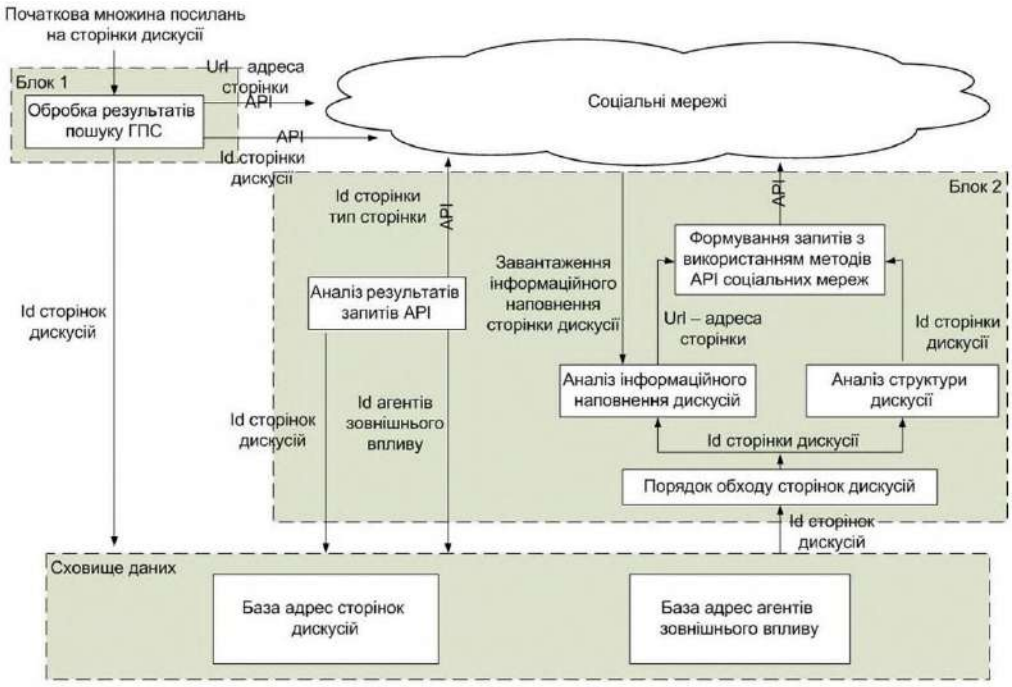


Рисунок 4.5 - Структурна схема функціонування поглибленого пошуку

Робота пошукової системи розділена на два блоки:

- ідентифікація (унікальних ідентифікаторів) сторінок обговорень, знайдених глобальною пошуковою системою Google за допомогою API-методів запитів соціальних мереж;

- аналіз переліку контактів у структурі сторінки соціальної групи та гіперпосилань в інформаційному змісті обговорень з використанням запитів API-

методів соціальних мереж.

Блок 1. Оскільки, коли ми шукаємо сторінки обговорень за допомогою глобальної пошукової системи Google, ми отримуємо коротку адресу, яку адміністратор групи може змінити, ми формуємо запит за допомогою API-методів соціальних мереж для визначення унікального коду обговорення. Унікальний код обговорення - це унікальний ідентифікатор групи, який присвоюється при створенні групи і не може бути змінений.

Блок 2. Для аналізу структури дискусії щодо наявності гіперпосилань використовуємо запит API-методів соціальних мереж, в результаті чого отримуємо унікальний код сторінки та тип сторінки.

Результатами роботи глибинного пошуку є уточнення переліку груп, які пов’язані з тематикою, що відповідає пошуковому запиту, та формування переліку зовнішніх гіперпосилань, для зміни коефінієнтів ранжування відповідних сторінок.

4.2 Проєктування програмного продукту

Сервіс містить графічні інструменти для оновлення, зберігання та видалення інформації в пошуковій системі, за допомогою яких розробник пошукових служб зможе працювати з власною пошуковою службою.

Існує два різні способи налаштування інформації.

Перший спосіб - це налаштування за допомогою інструментів графічної обгортки служби, а саме вибір інформаційного ключа, спосіб вирівнювання інформації та вибір головного ключа, за допомогою якого буде здійснюватися пошук інформації.

Другий спосіб - використовувати програмний код JS та розробити власну бібліотеку, яка забезпечить методи роботи з інформацією у створеній службі пошуку.

Для опису можливостей, що надаються розробленою системою, була розроблена схема прецедентів.

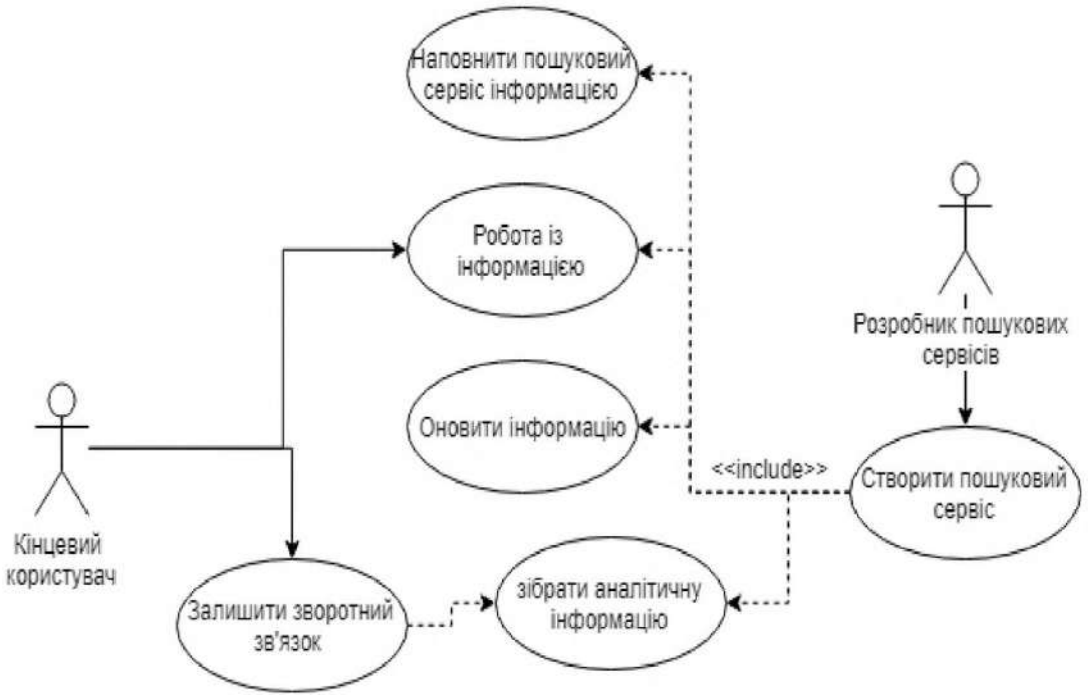


Рисунок 4.6 - Діаграма прецедентів системи пошуку

Діаграма прецедентів - це граф, який складається з набору акторів, прецедентів (використання), взаємозв'язку між акторами та прецедентами.

Клієнт-серверна архітектура є одним із архітектурних шаблонів дизайну програмного забезпечення і є домінуючою технологією у створенні розподілених мережевих застосунків та передбачає взаємодію та обмін даними між ними. Він включає такі основні компоненти:

- серверна частина;

- частина клієнта;

- інтерфейс прикладного програмного забезпечення [24].

Серверна частина складається з таких елементів, як Server, який написаний мовою програмування JavaScript та MongoDB - нереляційну базу даних.Серверна частина реагує на дані з API та вводить нові дані до бази даних, також залежить від запиту API додатка, отримує та перевіряє дані з бази даних та передає їх до відповідного запиту API застосунку.

API надає розробнику пошукових систем інструменти для швидкої розробки власної служби пошуку.

Для того, щоб використовувати інтерфейс прикладного програмування для розробки служби пошуку, вам потрібно знати мову програмування JavaScript і використовувати мою письмову бібліотеку для роботи з інтерфейсом прикладного програмування.

Розроблена бібліотека дозволить робити вкраплення програмного коду розробника пошукових служб на клієнтській частині кінцевих користувачів.

Логіка додатку розділена між серверною та клієнтською частинами, дані зберігаються на сервері, інформація обмінюється через мережу, яка працює через інтерфейс прикладного програмування. Однією з переваг цього підходу є той факт, що користувачі не залежать від конкретної операційної системи.

Nodejs використовує менеджер пакетів вузлів (менеджер пакетів вузлів)для роботи з іншими програмними модулями. Ви можете використовувати для встановлення локальних або глобальних пакетів.

У локальному режимі пакети встановлюються в каталог./node_modules батьківського каталогу. Власником каталогу є поточний користувач. Глобальні пакети встановлюються у каталог, власником якого є root в операційній системі. Також при створенні нового проекту та додавання до нього інших модулів створюється файл package.json

```
  "license": "MIT",
  "private": true,
  "dependencies": {
    "@babel/cli": "^7.1.2",
    "@babel/core": "^7.1.2",
    "@babel/node": "^7.0.0",
    "@babel/preset-env": "^7.1.0",
    "babel-preset-minify": "^0.5.0",
    "express": "^4.16.4",
    "global": "^4.3.2",
    "lodash": "^4.17.11",
    "luxon": "^1.5.0",
    "nodemon": "^1.18.5"
  },
  "scripts": {
    "start": "nodemon --exec babel-node src/index.js",
    "build": "babel src/index.js --out-file dist/server.js"
  },
  "devDependencies": {}
}
```

Рисунок 4.7 – Фрагмент серверної частини

Серверна частина виконує такі операції, як: прийняти запит від інтерфейсу

програми, перевірити та записати дані в базу даних, отримати дані з бази даних, перевірити та надіслати їх на запит з інтерфейсу програми.

Для роботи сервера з базою даних MongoDB вибрано модуль Mongoose.

MongoDB - це орієнтована на документи система управління базами даних із відкритим кодом (СУБД), яка не вимагає опису схеми таблиці [43]. MongoDB займає проміжок між швидкими та масштабованими системами, які працюють з даними у форматі ключ / значення, та реляційними базами даних, функціональними та простими для генерації запитів.

MongoDB - написана на мові C ++ документоорієнтованих NoSQL система управління базами даних з відкритим вихідним кодом, що не потребує опису схеми таблиць (schemaless).

MongoDB має такі рівні представлення даних:

1. Документ - JSON-об'єкт має довільне число полів. Поля могуть зберігати як простве значення, так до Вожен об'єкти і масиви.

2. Колекція (таблиця) - однотипні документи зберігаються в окремій колекції. Документи в колекції могуть бути проіндекіровани. Доступ до докумeніту можливий як по ключу, так і за значенням полів.

3. База даних - набір колекцій.

Підтримка MongoDB реалізована для більшості мов програмування: C, C++, C#, Java, Node.js, Perl, PHP, Python, Ruby, Scala.

Відмінності MongoDB від реляційних баз даних:

- Не підтримуються транзакції. Атомарність гарантується тільки на рівні всього документа, тобто часткового оновлення документа статися не може.
- Відсутність механізму «ізоляції». Будь-які дані, що зчитуються одним клієнтом, можуть паралельно модифікуватися іншим клієнтом.

Переваги MongoDB перед реляційними базами:

- Підтримка горизонтального масштабування з реплікацією даних. Дані можуть зберігатися на довільному числі серверів. Реплікація забезпечує відмовостійкість системи з підтримкою функціонала при виході вузлів з ладу.

- Формат зберігання даних (документ) близький до формату представлення даних в мовах програмування (об'єктів) не потрібно складних і дорогих запитів для отримання потрібного об'єкта.
- Підтримка операцій MapReduce для масової паралельної обробки даних.

Mongoose пропонує просте рішення на основі схеми для моделювання даних додатків. Він включає перевірку, функції зворотного виклику бізнес-логіки, побудову запитів, та багато іншого.

Mongoose дає величезну множину функціональних можливостей для створення та роботи з таблицями. Наразі Mongoose має вісім, які може мати властивість, яка зберігається до MongoDB [32].

Для кожного типу даних ви можете:

• вказати значення, встановивши

• вказати функцію користувача для перевірки даних

• вказати, що поле необхідно заповнити

• вказати функцію get (getter), яка дозволяє маніпулювати даними перед тим, як повернути їх як об’єкт

• вказати функцію набору (* setter), яка дозволяє маніпулювати даними перед збереженням їх у базі даних

• визначити індекси для швидшого пошуку даних

Змішаний тип даних використовується для перетворення властивості в «нечитабельне» поле (поле, в якому допустимий будь-який тип даних). Подібно до того, як багато розробників використовують MongoDB для різних цілей, це поле може зберігати дані різних типів, оскільки немає визначеної структури. . Використовуйте цей тип даних з обережністю, оскільки він обмежує можливості Mongoose, такі як перевірка даних та відстеження змін сутності для автоматичного оновлення властивостей під час зберігання.

Тип даних ObjectId зазвичай використовується для ідентифікації посилання на якийсь документ у вашій базі даних. Наприклад, якщо у вас є колекція книг та авторів, документ книги може містити властивість ObjectId, яка посилається на

конкретного автора документа.

Тип даних Array дозволяє зберігати схожі на JavaScript масиви. За допомогою цього типу даних ви можете виконувати типові операції JavaScript над масивами, такі як push, pop, shift, slice тощо.

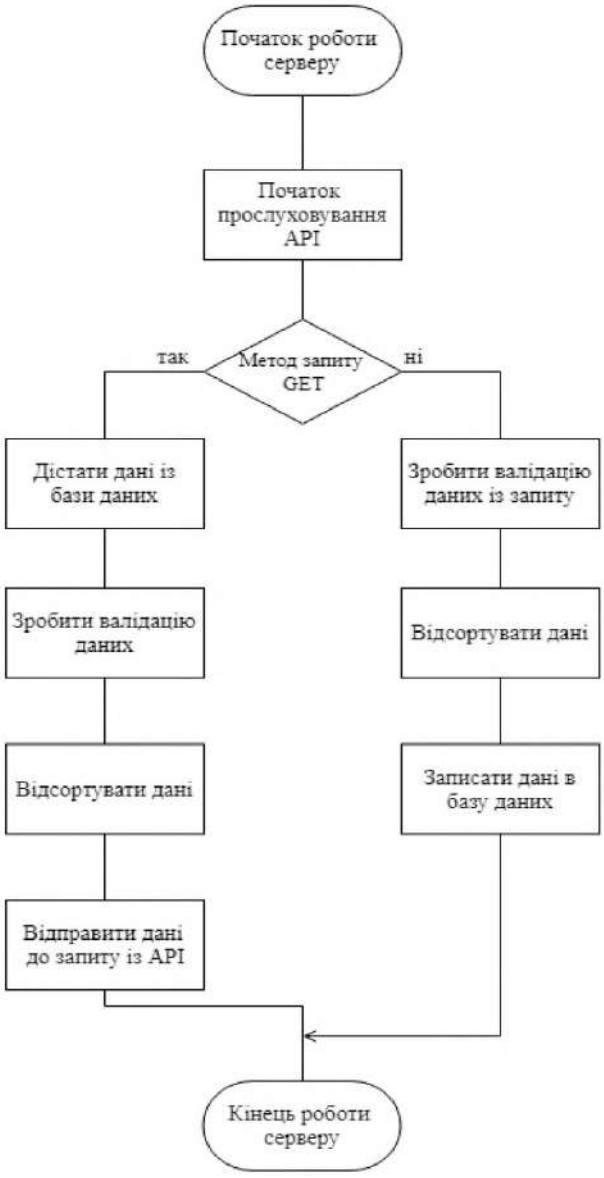


Рисунок 4.8 – Блок-схема алгоритму роботи серверної частини

Для розробки клієнтської частини використано такі технології Polyfill для підтримки програмного коду на різних браузерах babel/core, lingui/cli для керування програмною утилітою за допомогою терміналу, lingui/react, фреймворк react.

Поліфіл (англ. "Polyfill") або поліфілер (англ. "Polyfiller") - це фрагмент коду (або плагін), що надає функціонал необхідної технології, яка (ви, як розробник сподіваєтеся) буде нативним чином представлена браузером. Іншими словами, код буде працювати в точності так, як технологія, яку він, власне, і призначений представляти. Підробка архітектури API інтерфейсу, якщо вам завгодно.

Слід розуміти, що з технічних причин поліфіл не завжди може повною мірою відтворити можливості, що відсутні у старих переглядачах. Деякі функції можна відтворити лише частково, а для деяких створення запасного варіанту є геть неможливим.

Як свідчить офіційний слоган, React - це бібліотека для створення користувацьких інтерфейсів. React не є фреймворком - він навіть не розрахований виключно для web. Він використовується для візуалізації і в зв'язці з іншими бібліотеками. Наприклад, React Native можна використовувати для створення мобільних додатків; React 360 можна використовувати для створення додатків віртуальної реальності; крім того є й інші варіанти .

Для створення вебдодатків розробники використовують React в тандемі з ReactDOM . React and ReactDOM часто обговорюються в тому ж просторі і використовуються для вирішення тих же проблем, що й інші справжні фреймворки для веб-розробки. Коли ми посилаємося на React як на «фреймворк», ми маємо на увазі це розмовне розуміння.

Основна мета React - мінімізувати помилки, що виникають при розробці користувальницьких інтерфейсів. Це досягається за рахунок використання компонентів - автономних логічних фрагментів коду, які описують частина призначеного для користувача інтерфейсу. А вже ці компоненти об'єднуються для створення повноцінного користувальницького інтерфейсу. React абстрагує більшу частину роботи по візуалізації, залишаючи вам можливість зосередитися на дизайні.

На відміну від інших платформ, React не зобов'язує до суворих правил щодо угод про код або організації файлів. Це дозволяє командам домовлятися, що для

них більш підходить, і структурувати React проект відповідним чином. React може відповідати за одну кнопку, кілька частин або ж весь призначений для користувача інтерфейс програми.

Крім того, такі переваги React-додатки, як написання інтерфейсів за допомогою JSX, вимагають процесу компіляції. Додавання на сайт компілятора Babel призводить до більш повільного виконання коду, тому такі інструменти зазвичай настроюються для процесу складання. Так, можливо, у React є серйозні вимоги до інструментарію, але цього можна освоїти.

4.3 Оцінка впливу груп в соціальних мереж на пертинентність результатів пошуку

Припустимо, що малодосвідчений користувач хоче замовити гаджет компанії «Apple». Для цього в пошуковій системі Google він введе запит «Apple».

Рисунок 4.9 – Результати пошукук по запиту

Як видно з рисунку 4.9 – друге та третє посилання серед результатів пошуку є непотрібними, оскільки користувач хоче замовити товар цієї компанії.

При ранжування отриманих результатів пошуку за допомогою алгоритму ModPR. Тепер, завдяки роботі блоку поглибленого пошуку (рис 4.5) можна використати допоміжну БД із специфікою груп відповідно до пошукового запиту та тематики.

При ранжуванні із ModPR, вебсторінка, яка 4 в рейтингу підніметься на другу позицію. Аналогічно відбудеться і з іншими корисними та малорейтинговими сторінками, про які є багато згадок в групах соціальних мереж. Такий порядок буде ефективнішим для користувача, оскільки він швидко знайде необхідні йому веб-сайте, а найголовніше – корисні, якісні джерела, що містять задовільняючий вміст.

Отже при ранжуванні ModPR задіює власну БД для формування рейтингу ресурса у відповідності до потреби користувача, тобто пертинентності. Це відбувається наступним чином:

Відранжовані результати пошуку надаються користувачеві в вигляді сторінки видачі пошукових результатів. Проведення відбору зображено на рисунку 4.10.

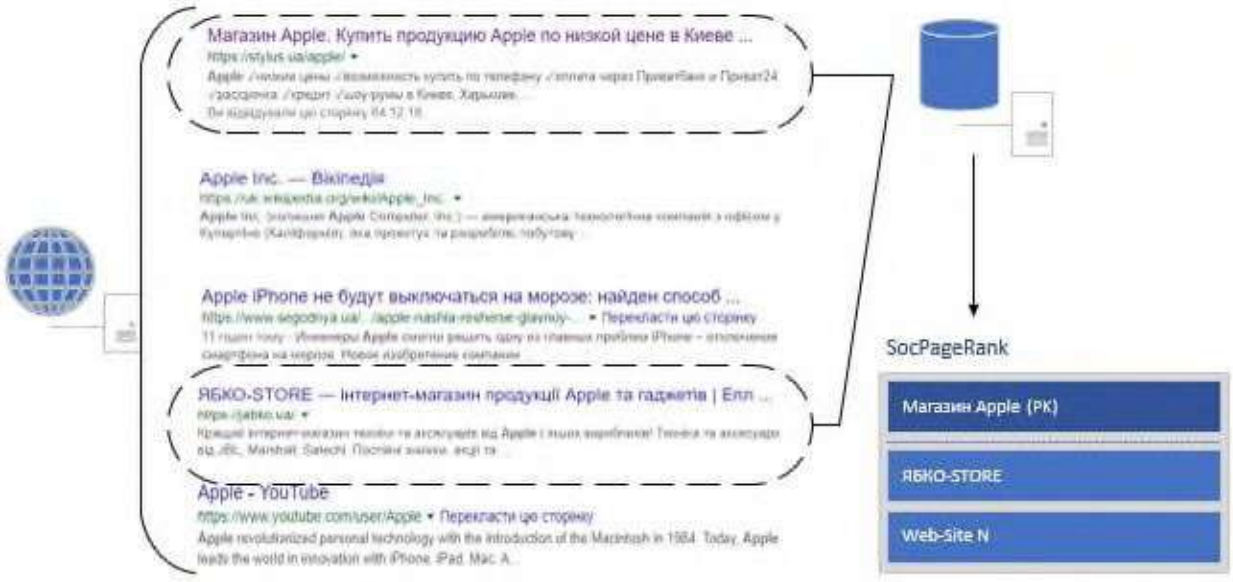


Рисунок 4.10 – Схема процесу відбору вебресурсів із допоміжної бази даних

Знайдені та відранжовані результати пошуку будуть виведені користувачеві у вигляді SERP сторінки.

Рисунок 4.11 – SERP, яку видає запропонований метод ModPR

На рисунку 4.11 видно, що такий список результатів є більш оптимальним від попереднього (рисунок 4.9).

Порівняння отриманих результатів ваги сайту за умови звичайного алгоритму ранжування та реальної ваги сайту після модифікації доводить, що запропонований метод є більш ефективним для збільшення пертинентності пошуку.

4.4 Висновки

У цьому розділі описується процес розробки програмного забезпечення для створення системи пошуку інформації, яка дозволить вам розробляти нові служби

пошуку для певної категорії та наповнювати їх інформацією.

Описано вимоги користувача, функціональні вимоги та можливості розробленої системи, описано обґрунтування вибору стеку технологій.

Проаналізовано сучасні мови програмування, середовища розробки та бібліотеки. Для реалізації було обрано найбільш підходящі інструменти для виконання завдання.

Для розробки клієнтської частини використано такі технології Polyfill для підтримки програмного коду на різних браузерах babel/core, lingui/cli для управління програмною утилітоюза допомогою терміналу, lingui/react, бібліотека react.

Для роботи цієї модифікації в пошуковій системі повинна бути присутня не тільки основна а й допоміжга база даних ModPR.

Запропоновано алгоритм глибинного пошуку для виявлення інформаційного впливу соціальних мереж на індексацію сторінки та формування переліку зовнішніх гіперпосилань, дя зміни коефінієнтів ранжування відповідних сторінок.

Цей алгоритм підніме маловідомі корисні джерела на вищі позиції, конкретизує та звужує пошук до обсягу теми, щодо якої було зроблено запит. Його ефективно використовувати у конкретних галузях та напрямках пошуку, оскільки він призначений для специфіки пошукової сфери.

ВИСНОВКИ

У даній магістерській роботі виділено недоліки сучасних інформаційно-пошукових систем, які використовуються для пошуку та роботи з інформацією у мережі Інтернет. В роботі вирішено наукове завдання – розроблено наукові основи моделювання роботи інформаційно-пошукових систем. Мета магістерського дослідження полягає в підвищені пертинентності результатів пошуку за рахунок модифікації алгоритму ранжування Google – PageRank.

За допомогою вдосконаленого алгоритму ранжування ModPR, корисні, актуальні та малопопулярні сайти які часто обговорюють користувачі в тематичних соціальних групах матимуть змогу отримувати свій реальний рейтинг.

Основні результати магістерської роботи є такими:

1. Проаналізовано та досліджено принципи пошуку інформації в Google.
2. Розглянуто математичну модель алгоритму існуючого алгоритму ранжування.
3. Вдосконалено математичну модель ранжування сайтів пошукової системи Google, за рахунок враховування популярності сторінки в тематичних соціальних спільнотах.
4. Розроблено метод підвищення пертинентності результату пошуку за рахунок вдосконалення алгоритму ранжування та індексації сайтів/
5. Розроблено алгоритми пошуку сторінок груп у соціальних мережах з застосуванням розширених можливостей глобальних пошукових машин та запитів API-методів, які дають змогу виявити сторінки груп у соціальних мережах відповідно їх інформаційного наповнення.
6. Вдосконалено метод виявлення інформаційного впливу соціальних мереж на індексацію сторінки та формування переліку зовнішніх гіперпосилань, для зміни коефіцієнтів ранжування відповідних сторінок.
7. Кількість пертинентних посилань серед перших десяти, одержаних унаслідок пошуку, збільшилась в середньому на 2.

8. Проведено практичне дослідження роботи методу.

9. Описано вимоги користувача, функціональні вимоги та можливості розробленої системи, описано обґрунтування вибору стеку технологій.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1 Ашманов И. С. Продвижение сайта в поисковых системах / И. С. Ашманов, А. А. Иванов. - М.: Вильямс, 2007. - 304 с.

2 Багузин С. Алгоритм ссылочного ранжирования: PageRank и линейная алгебра [Електронний ресурс] / С.Багузин // Baguzin. – Режим доступу: http://baguzin.ru/wp/algoritm-ssylochnogo-ranzhirovaniya-pagerank-i/ (дата звернення 17.09.2020).

3 Види пошукових запитів [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – mylink.org.ua – Режим доступу: http://mylink.org.ua/vydy-poshukovyh-zapytiv/ (дата звернення 6.00.2020).

4 Грайворонська А.М., Ланде Д.В. Дослідження інформаційних потоків, як динамічних мультиагентних систем // Системный анализ и информационные технологии: материалы 17-й Международной научнотехнической конференции SAIT 2015, Киев, 22 -25 июня 2015 г. / - К.: УНК "ИПСА" НТУУ "КПИ", 2015. - С. 62-63.

5 Дорнфест Р. Секреты Google. Трюки и тонкая настройка / Р. Дорнфест, Р. Бош, Т. Калишейн. - М.: Русская редакция, 2008. - 510 с.

6 Загальний огляд Інтернет [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – ua-referat.com – Режим доступу: http://ua-referat.com/Загальний_огляд_Інтернет (дата звернення 19.09.2020).

7 Как работает Google поиск, основные алгоритмы обновлений [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – habr.com – Режим доступу: https://habr.com/company/ua-hosting/blog/277819/ (дата звернення 17.09.2020).

8 Класифікація пошукових запитів [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – igroup.com.ua – Режим доступу: http://igroup.com.ua/seo-articles/klasyfikatsiya-poshukovyh-zapytiv/ (дата звернення 16.08.2020).

9 Ландэ Д.В. Корпоративная система мониторинга сетевых информационных ресурсов на основе мультиагентного подхода / Д.В. Ландэ, В.А. Додонов, Т.В. Коваленко // Інформаційні технології та безпека, 2016. - Т. 2. - №

2(52). - С. 80-87.

10 Методи пошуку інформації в інтернеті [Електронний ресурс]. – Режим доступу: http://ua-referat.com/Методи_пошуку_інформації_в_Інтернеті (дата звернення 19.04.2020).

11 Мурах Б.Р. Підвищення пертинентності результатів пошуку за рахунок модифікації алгоритму ранжування Google /Б.Р. Мурах, І.В. Муляр // Збірник наукових праць за матеріалами XII всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2020». Хмельницький – 2020. – С. 188-193.

12 Мурах Б.Р. Дослідження інформаційного пошуку / О.В. Огнєвий,Є.С. Лєнков, І.В.Гурман, Б.Р. Мурах // Тези доповідей XVI Міжнародної науково-практичної конференції "Військова освіта і наука: сьогодення та майбутнє" Том 2 [Текст] / за заг. редакцією Ігоря Толока. – К. : ВІКНУ, 2020. – С. 54

13 Основні фактори ранжування сайтів [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – lemarbet.com – Режим доступу: https://lemarbet.com/ua/razvitie-internet-magazina/faktory-ranzhirovaniya/ (дата звернення 16.08.2020).

14 Основи розробки веб-додатків. Навчальний посібник / В.В. Осадчий, В.С. Круглик - Мелітополь: ТОВ «Видавничий будинок ММД», 2012. - 540 с.

15 Поведінкові фактори ранжирування: все, що вам потрібно про них знати [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – lemarbet.com – Режим доступу: https://lemarbet.com/ua/otkrytie-internet-magazina/povedencheskie-faktory-ranzhirovaniya-vse-chto-vam-nuzhno-o-nih-znat/ (дата звернення 15.08.2020).

16 Пріоритетні наукові напрями та найважливіші проблеми: від теорії до практики. Матеріали міжнародної науково-практичної конференцій. - Одеса: ГО «Інститут інноваційної освіти», 2017. - 164 с.

17 Растолкованный PageRank, или Все, что вы всегда хотели знать о PageRank [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – digits.ru – Режим доступу: http://digits.ru/articles/promotion/pagerank.html (дата звернення 27.05.2020).

18 Технологія пошуку інформації засобами мережі Інтернет: основні способи пошуку інформації в Інтернеті [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – disted.edu.vn.ua – Режим доступу: https://disted.edu.vn.ua/courses/learn/3121 (дата звернення 16.09.2020).

19 Трофименко Є. PageRank, початки аналізу [Електронний ресурс] / Є.Трофименко // Ua-referat. – Режим доступу: http://ua-referat.com/PageRank_початки_аналізу

20 Что такое Docker и технология контейнеров Linux [Електронний ресурс] - Режим доступу до ресурсу: https://vps.ua/blog/docker-and-linux-containers. (дата звернення 17.06.2020).

21 Цынгуев Б.Т. Математические модели ранжирования вершин в графах коммуникационных сетей [Електронний ресурс] / Б.Т.Цыгунев // Petrsu. – Режим доступу: https://petrsu.ru/files/user/beb228124c2cc2305a9ec67be48c9b98/diss_cinguev.pdf (дата звернення 03.09.2020).

22 Як працює пошук: користувач у центрі уваги [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – google.com. – Режим доступу: https://www.google.com/intl/uk/search/howsearchworks/mission/web-users/ (дата 27.05.2018).

23 Bille P, Gørtz IL, Vildhøj HW, Wind DK (2012) String matching with variable length gaps. Theoret Comput Sci 443(1):25–34

24 Cole R, Hariharan R (20182) Verifying candidate matches in sparse and wildcard matching. In: Proceedings of the 34th annual ACM symposium on theory of computing, May 2018, pp 592–601

25 «Github» [Електронний ресурс]: [Веб-сайт]. – Режим доступу: https://github.com/ (дата звернення 05.03.2019)

26 Guo D, Hong XL, HuX G, Gao J, Liu YL, Wu GQ, Wu XD (2011) A bit-parallel algorithm for sequential pattern matching with wildcards. Cybernet Syst 42(6):382–401

27 Gonzalo N, Mathieu R (2007) Flexible pattern matching in strings: practical on-line search algorithms for texts and biological sequences. Publishing House of Electronics Industry, Beijing

28 «IntelliJ IDEA» [Електронний ресурс]: [Веб-сайт]. – Режим доступу: https://www.jetbrains.com/idea/ (дата звернення 05.03.2019)

29 «Java» [Електронний ресурс]: [Веб-сайт]. – Режим доступу: https://www.java.com/en/ (дата звернення 05.03.2019)

30 Liu, N., Xie, F. & Wu, X. Pattern Anal Applic (2018) 21: 1151. https://doi.org/10.1007/s10044-018-0733-0

31 PageRank [Електронний ресурс] : [Веб-сайт]. – Режим доступу: https://uk.wikipedia.org/wiki/PageRank

32 Segev E. Google and the Digital Divide: The Bias of Online Knowledge / Elad Segev. - Oxford: Chandos Publishing, 2010. - 221 c.

33 Rakesh V., Singh D., Vinzamuri B., Reddy C.K. Personalized Recommendation of Twitter Lists Using Content and Network Information // Association for the Advancement of Artificial Intelligence (Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media, 2014.

34 Rahman MS, Iliopoulos CS, Lee I et al (2016) Finding patterns with variable length gaps and don’t cares. In: Proceedings of the 12th annual international computing and combinatorics conference, vol 8, pp 146–155

35 Woo J., Chen H. Epidemic model for information diffusion in web forums: experiments in marketing exchange and political dialog // Springerplus, 2016. - № 22. - pp. 5-66.

36 Zhang H, Chow TW, Wu QM (2016) Organizing books and authors by multilayer SOM. IEEE Trans Neural Netw Learn Syst 27(12):2537

37 Zhou Z, Zhang T, Chow SSM, Zhang Y, Zhang K (2016) Efficient authenticated multi-pattern matching. In: Presented at the 11th ACM, ACM Press, New York, USA, pp 593–604.

ДОДАТОК А

(обов’язковий)
Фрагмент програмного коду алгоритму

```
 def ModPR(G, alpha=0.85, personalization=None,
         max_iter=100, tol=1.0e-6, nstart=None, weight='weight',
         dangling=None):
"""Return the PageRank of the nodes in the graph.

PageRank computes a ranking of the nodes in the graph G based on
the structure of the incoming links. It was originally designed as
an algorithm to rank web pages.

Parameters
-------------
G : graph
  A NetworkX graph.  Undirected graphs will be converted to a directed
  graph with two directed edges for each undirected edge.


alpha : float, optional
  Damping parameter for PageRank, default=0.85.


personalization: dict, optional
  The "personalization vector" consisting of a dictionary with a
  key for every graph node and nonzero personalization value for each node.
  By default, a uniform distribution is used.


max_iter : integer, optional
  Maximum number of iterations in power method eigenvalue solver.


tol : float, optional
  Error tolerance used to check convergence in power method solver.


nstart : dictionary, optional
  Starting value of PageRank iteration for each node.


weight : key, optional
  Edge data key to use as weight.  If None weights are set to 1.


dangling: dict, optional
  The outedges to be assigned to any "dangling" nodes, i.e., nodes without
  any outedges. The dict key is the node the outedge points to and the dict
  value is the weight of that outedge. By default, dangling nodes are given
  outedges according to the personalization vector (uniform if not
  specified). This must be selected to result in an irreducible transition
  matrix (see notes under google_matrix). It may be common to have the
  dangling dict to be the same as the personalization dict.


Returns
----------
pagerank : dictionary
   Dictionary of nodes with PageRank as value


Notes
-------
The eigenvector calculation is done by the power iteration method
and has no guarantee of convergence.  The iteration will stop
after max_iter iterations or an error tolerance of
```

```
number_of_nodes(G)*tol has been reached.

The PageRank algorithm was designed for directed graphs but this
algorithm does not check if the input graph is directed and will
execute on undirected graphs by converting each edge in the
directed graph to two edges.


"""
if len(G) == 0:
    return {}

if not G.is_directed():
    D = G.to_directed()
else:
    D = G

# Create a copy in (right) stochastic form
W = nx.stochastic_graph(D, weight=weight)
N = W.number_of_nodes()

# Choose fixed starting vector if not given
if nstart is None:
    x = dict.fromkeys(W, 1.0 / N)
else:
    # Normalized nstart vector
    s = float(sum(nstart.values()))
    x = dict((k, v / s) for k, v in nstart.items())

if personalization is None:

    # Assign uniform personalization vector if not given
    p = dict.fromkeys(W, 1.0 / N)
else:
    missing = set(G) - set(personalization)
    if missing:
        raise NetworkXError('Personalization dictionary '
                            'must have a value for every node. '
                            'Missing nodes %s' % missing)
    s = float(sum(personalization.values()))
    p = dict((k, v / s) for k, v in personalization.items())

if dangling is None:

    # Use personalization vector if dangling vector not specified
    dangling_weights = p
else:
    missing = set(G) - set(dangling)
    if missing:
        raise NetworkXError('Dangling node dictionary '
                            'must have a value for every node. '
                            'Missing nodes %s' % missing)
    s = float(sum(dangling.values()))
    dangling_weights = dict((k, v/s) for k, v in dangling.items())
dangling_nodes = [n for n in W if W.out_degree(n, weight=weight) == 0.0]

# power iteration: make up to max_iter iterations
for _ in range(max_iter):
    xlast = x
    x = dict.fromkeys(xlast.keys(), 0)
    danglesum = alpha * sum(xlast[n] for n in dangling_nodes)
```

```
    for n in x:

        # this matrix multiply looks odd because it is
        # doing a left multiply x^T=xlast^T*W
        for nbr in W[n]:
            x[nbr] += alpha * xlast[n] * W[n][nbr][weight]
        x[n] += danglesum * dangling_weights[n] + (1.0 - alpha) *  p[n]

    # check convergence, l1 norm
    err = sum([abs(x[n] - xlast[n]) for n in x])
    if err < N*tol:
        return x
raise NetworkXError('pagerank: power iteration failed to converge '
                    'in %d iterations.' % max_iter)
```

ДОДАТОК Б

(Обов'язковий)

Копії наукових праць

к.т.н., доц. Огнєвий О.В. (ХмНУ)
к.т.н. Лєнков Є.С. (ЦНДІ ЗСУ)
к.т.н., доц. Гурман І.В. (ХмНУ)
Мурах Б.Р (ХмНУ)

Дослідження інформаційного пошуку

У час глобального розвитку мережі Інтернет ефективне використання можливостей, що вона відкриває, може стати вирішальним чинником успішності більшості починань. Більшість українців очікують, що зможуть отримати необхідну їм інформацію в он-лайн режимі.

Завдання ранжирування документів, згідно з деякими заздалегідь визначеними критеріями, підпадає під відповідальність алгоритмів ранжирування. Алгоритм ранжирування є однією з найважливіших складових будь-якої пошукової системи і зазвичай вимагає великої уваги під час розробки двигуна. Різні пошукові системи використовують різні класи алгоритмів ранжування з різним ступенем ефективності та ефективності. Інтуїтивно зрозуміло, що хороша система пошуку інформації повинна

представляти відповідні документи вище за рейтингом, а менш релевантні документи слід за ними. Незважаючи на те, що алгоритми ранжирування, слідуючи процесу пошуку, прагнуть до досягнення цієї мети, часто зустрічається багато нерелевантного серед відповідної запитуваної інформації. Цей неоптимальний результат призвів до кількох досліджень у галузі алгоритмів ранжування пошукових систем.

У цій роботі проводиться дослідження основних алгоритмах пошуку інформації в пошукових системах, їх аналізу та порівнянні. Автором пропонується алгоритм ранжування та індексації вебсайтів, який призначений для підвищення ефективності пошуку інформації з врахуванням їхньою популярності в соціальних мережах та форумах.

Мета дослідження полягає в підвищені пертинентності результатів пошуку за рахунок модифікації алгоритму ранжування Google – PageRan.

Для досягнення цієї мети в роботі необхідно вирішити наступні завдання:

1. Провести огляд та аналіз публікацій в сфері інформаційного пошуку.
2. Проаналізувати фактори, що впливають на ефективне функціонування пошукових систем.
3. Розглянути існуючі алгоритми пошуку інформації.
4. Дослідити алгоритм ранжування пошукової системи Google.
5. Розробити модифікований алгоритм ModPR.
6. Дослідити розроблений алгоритм та оцінити його ефектиність.

УДК 004.522

Муляр І. В, Мурах Б. Р.

Хмельницький національний університет

ПІДВИЩЕННЯ ПЕРТИНЕНТНОСТІ РЕЗУЛЬТАТІВ ПОШУКУ ЗА РАХУНОК МОДИФІКАЦІЇ АЛГОРИТМУ РАНЖУВАННЯ GOOGLE

У роботі поставлено наукове завдання – досліджено основи моделювання роботи інформаційно-пошукових систем. Мета магістерського дослідження полягає в підвищенні пертинентності результатів пошуку за рахунок модифікації алгоритму ранжування Google – PageRank. Для цього розроблено алгоритми пошуку сторінок груп у соціальних мережах з використанням розширених можливостей глобальних пошукових систем та запитів API-методів, які дають змогу вишукати сторінки груп у соціальних мережах відповідно їх інформаційного наповнення.

За допомогою вдосконаленого алгоритму ранжування ModPR, корисні, актуальні та малопопулярні сайти які часто обговорюють користувачі в тематичних соціальних групах матимуть змогу отримувати свій реальний рейтинг.

The scientific problem is solved in the work - the basics of modeling the work of information retrieval systems are investigated. The purpose of the master's study is to increase the pertinence of search results by modifying Google's ranking algorithm – PageRank. To do this, algorithms have been developed to search for group pages on social networks using the advanced capabilities of global search engines and queries of API-methods, which allow to find group pages on social networks according to their content.

With ModPR's advanced ranking algorithm, useful, relevant and unpopular sites that are often discussed by users in thematic social groups will be able to get their real ranking.

Вступ. Традиційні способи пошуку відповідних сторінок у випадку односкладових запитів не дають задовільних результатів, оскільки популярні теми завжди знайдуть велику кількість сторінок з однаковою актуальністю. Для того, щоб якось упорядкувати такі сторінки, пошукові системи вдаються до різних інструментів Для цього Google використовує PageRank, що дає приголомшливі результати [1].

Кількість вихідних посилань з батьківського ресурсу має велике значення для розрахунку рейтингу сайту потомка. Google допускає, що один вебресурс може передати максимум 85% свого вагового коефіцієнту, тоді як її рейтинг не зменшується [2].

Постановка задачі. В ході дослідження потрібно модифікувати алгоритм ранжування Google – PageRank для підвищення пертинентності результатів пошуку. Для цього необхідно розробити алгоритм пошуку сторінок груп у соціальних мережах з використанням розширених можливостей глобальних пошукових систем

та запитів API-методів, які дають змогу виявляти сторінки груп у соціальних мережах відповідно їх інформаційного наповнення.

Основна частина. Модифікований алгоритм також враховує значимість кожної сторінки, що отримала голос, адже голоси деяких сторінок є важливішими, і їх популярність в соціальних групах.

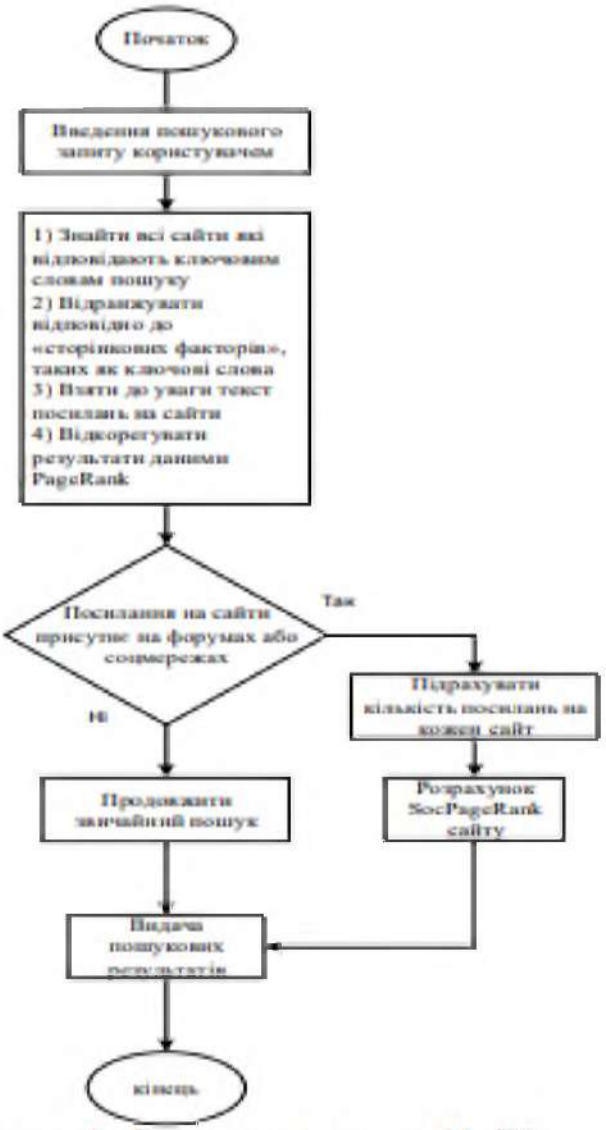


Рисунок 1 – Блок-схема алгоритму ModPR

За допомогою вдосконаленого алгоритму ранжування ModPR, корисні, актуальні та малопопулярні сайти які часто обговорюють корстувачі в тематичних соціальних групах матимуть змогу отримувати свій реальний рейтинг.

Особливістю модифікованого алгоритму нова формула розрахунку ваги сайту – ModPR. Тут враховується додатковий ваговий коефіцієнт популярності сайту на форумах та соціальних мережах.

Пошук сторінок груп здійснюється інструментами соціальних мереж за назвами форумів або за їх коротким змістом. Але це завжди відповідає інформаційному наповненню цих документів.

Як результат, виникла проблема, пов'язана з пошуком необхідної інформації на сторінках груп у соціальних мережах, що значно ускладнюється необхідністю пошуку відповідно до змісту та релевантності сторінки обговорення з урахуванням функціонування сторінки для обговорення в соціальних мережах, а саме [3]:

- сторінки мають низький рейтинг в алгоритмах ранжування сторінок;
- велика кількість вебсторінок для обговорення не класифікуються глобальними пошуковими системами;
- взаємозв'язок вебсторінок для обговорення;
- збереження форумів неактуальної тематичної спрямованості.

Запит глобальної пошукової системи для пошуку відповідних обговорень складається з таких операцій:

- операція локалізації пошукового запиту;
- операція виявлення інформаційного вмісту;
- обмеження часу.

Загальна структура запиту наведена на рис. 2 [4]

Рисунок 2 – Формалізований запит для виявлення релевантних дискусій вебсторінок

Операція локалізації пошукового запиту дозволяє обмежити пошук вебсайтом або доменом. Функціонально ця операція складається з оператора локалізації та специфікатора вебадреси.

Оператор аналізу вмісту - оператори послуг глобальної пошукової системи, які аналізують вміст вебсторінки за заданою темою. Ми будемо розглядати слово чи фразу, що містяться в тілі веб-сторінки, як інформаційний зміст сторінки.

Специфікатор вмісту - це функція, яка характеризує об'єкт пошуку. Ознака інформаційного змісту - це слово або фраза в інформаційному вмісті веб-сторінки.

Соціальна мережа Facebook дозволяє користувачам створювати власні групи та групові сторінки, в яких вони можуть об'єднуватися відповідно до інтересів. Група - це спільнота, в якій люди можуть спілкуватися між собою. Однією з головних особливостей груп є те, що вони ***не індексуються*** глобальними пошуковими системами. На відміну від груп, сторінки індексуються глобальними пошуковими системами [5].

Отже, пошук груп ведемо за допомогою пошуку сторінок у «Facebook», використовуючи формалізований запит наведений на рис. 3, з подальшим аналізом HTML-коду сторінки.

Рисунок 3 – Формалізований запит для виявлення груп у соціальній мережі «Facebook»

Формалізатор сторінки «спільнота» вноситься до формалізованого запиту на пошук сторінок у соціальній мережі «Facebook» в операції пошуку інформаційного контенту. Пошук здійснюється за специфікатором інформаційного змісту (Ключові слова) - набором ключових слів. Сторінки у "Facebook" чітко структуровані, і для визначення присутності відповідних обговорень ми аналізуємо HTML-код на наявність пункту меню "групи" (рис. 4).

Рисунок 4 – Дослідження HTML-коду сторінки для виявлення URL-адреси сторінки переліку тематично зв'язаних груп

За результатами аналізу формуємо URL-адресу тематично зв'язаних груп.

Висновки. В рамках цієї роботи запропоновано алгоритм глибинного пошуку для виявлення інформаційного впливу соціальних мереж на індексацію сторінки та формування переліку зовнішніх гіперпосилань, для зміни коефіцієнтів ранжування відповідних сторінок.

Цей алгоритм підніме маловідомі корисні джерела на вищі позиції, конкретизує та звужує пошук до обсягу теми, щодо якої було зроблено запит. Його ефективно використовувати у конкретних галузях та напрямках пошуку, оскільки він призначений для специфіки пошукової сфери

Перелік посилань

1. Багузин С. Алгоритм ссылочного ранжирования: PageRank и линейная алгебра [Електронний ресурс] / С.Багузин // Baguzin. – Режим доступу: http://baguzin.ru/wp/algoritm-ssylochnogo-ranzhirovaniya-pagerank-i/
2. Как работает Google поиск, основные алгоритмы обновлений [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – habr.com – Режим доступу: https://habr.com/company/ua-hosting/blog/277819/
3. Основні фактори ранжування сайтів [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – lemarbet.com – Режим доступу: https://lemarbet.com/ua/razvitie-internet-magazina/faktory-ranzhirovaniya/
4. Segev E. Google and the Digital Divide: The Bias of Online Knowledge / Elad Segev. - Oxford: Chandos Publishing, 2010. - 221 c.
5. Види пошукових запитів [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – mylink.org.ua – Режим доступу: http://mylink.org.ua/vydy-poshukovyh-zapytiv/

ДОДАТОК В

(Обов’язковий)

Презентація

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Мурах Богдан

МЕТОД ПІДВИЩЕННЯ ПЕРТИНЕНТНОСТІ РЕЗУЛЬТАТУ ПОШУКУ ЗА РАХУНОК ВДОСКОНАЛЕННЯ АЛГОРИТМУ РАНЖУВАННЯ ТА ІНДЕКСАЦІЇ САЙТІВ

Науковий керівник
к.т.н., доц. Муляр І.В.

Тема: Метод підвищення пертинентності результату пошуку за рахунок вдосконалення алгоритму ранжування та індексації сайтів

Мета магістерської роботи полягає в полягає в підвищені пертинентності результатів пошуку за рахунок модифікації алгоритму ранжування Google – PageRank.

Об'єкт дослідження: є процес пошуку та індексації сайтів для формування їх рейтингу пошуковою системою Google.
Предмет дослідження: є методи і алгоритми роботи пошукових систем.

Задачі досліджень у роботі формулюються наступним чином:

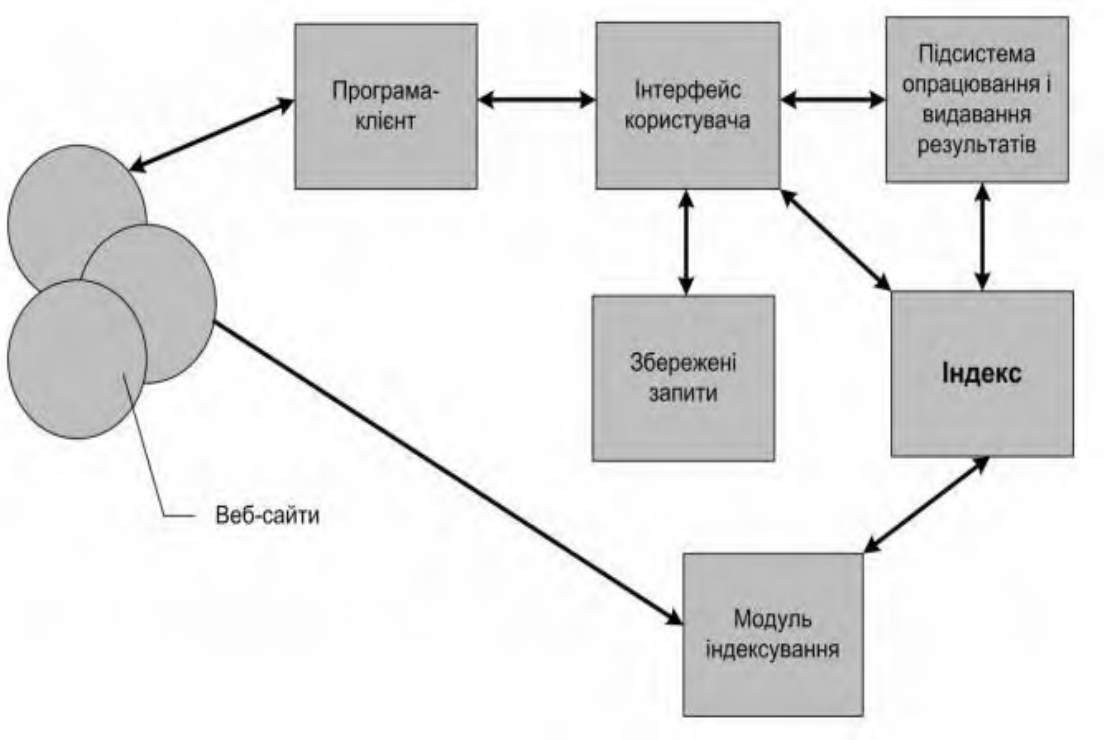
1. Проаналізувати та детально дослідити принципи пошуку інформації в Google.
2. Розглянути математичну модель існуючого алгоритму та вдосконалити її.
3. Вдосконалити алгоритм ранжування сайтів пошукової системи Google, за рахунок враховування популярності сторінки в тематичних соціальних спільнотах.
4. Розробити метод виявлення інформаційного впливу соціальних мереж на індексацію сторінки та формування переліку зовнішніх гіперпосилань, для зміни коефіцієнтів ранжування відповідних сторінок.
5. Проведено дослідження розробленого методу, та оцінено його ефективність

Наукова новизна

1. Вдосконалено алгоритм ранжування сайтів пошукової системи Google, який враховує популярність сторінки в тематичних соціальних спільнотах.
2. Розроблено метод виявлення інформаційного впливу соціальних мереж на індексацію сторінки та формування переліку зовнішніх гіперпосилань, для зміни коефіцієнтів ранжування відповідних сторінок.

Практична цінність. Модифікований алгоритм формування рейтингу вебресурсів ModPR доцільно використати для пошуку інформації яка найбільш часто обговорюється на форумах та в соціальних групах
Апробація роботи. Наукові результати і основні положення магістерської роботи доповідались і обговорювались на всеукраїнських та міжнародних науково-технічних конференціях,

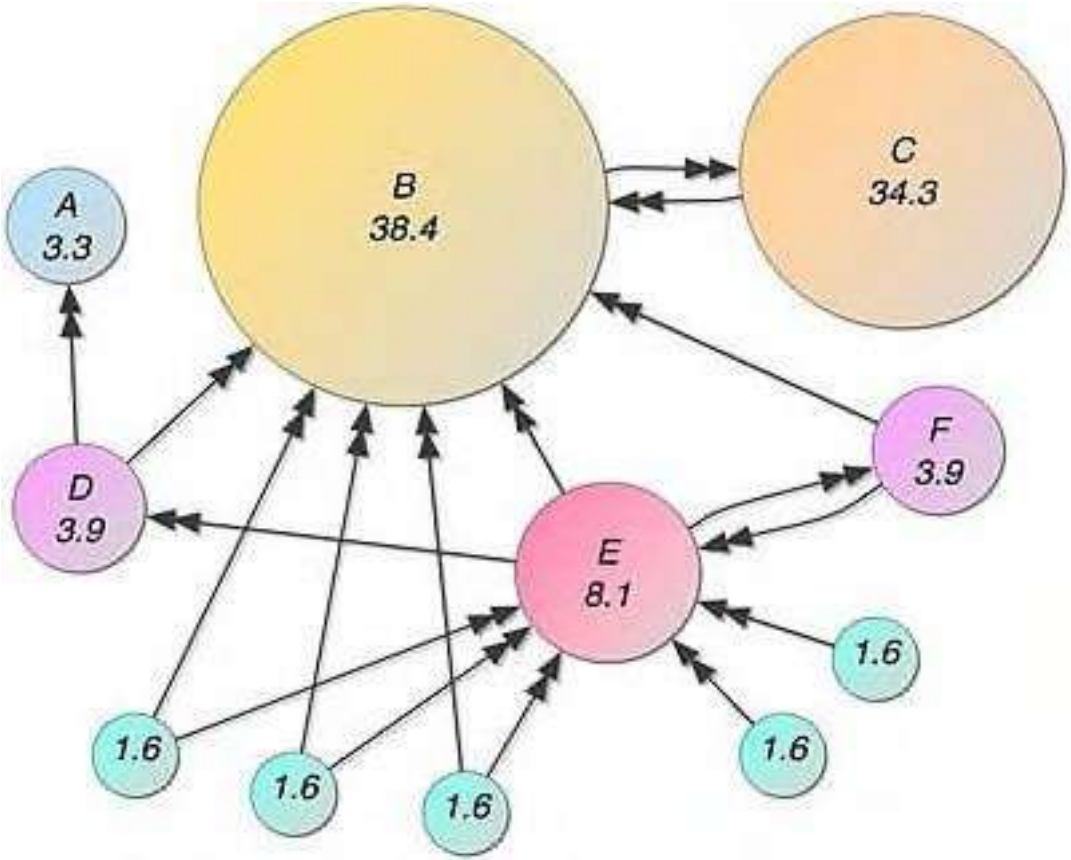
Типова структура пошукової системи для WWW



Складові структури

1) Модуль індексування - він служить для постійного сканування мережі Інтернет та підтримування бази даних індексу в актуальному стані.
2) Індекс пошукової системи (index database) - це база даних, яка зберігається на пошуковому сервері.
3) Підсистема опрацювання та видавання результатів (Search Engine and Results Engine).
4) Інтерфейс користувача (user interface).
5) Збережені запити (saved queries) - це запити, які надходять до пошукової системи від користувачів.
6) Програма-клієнт (програма перегляду) (client).
7) Веб-сайти (WWW sites).

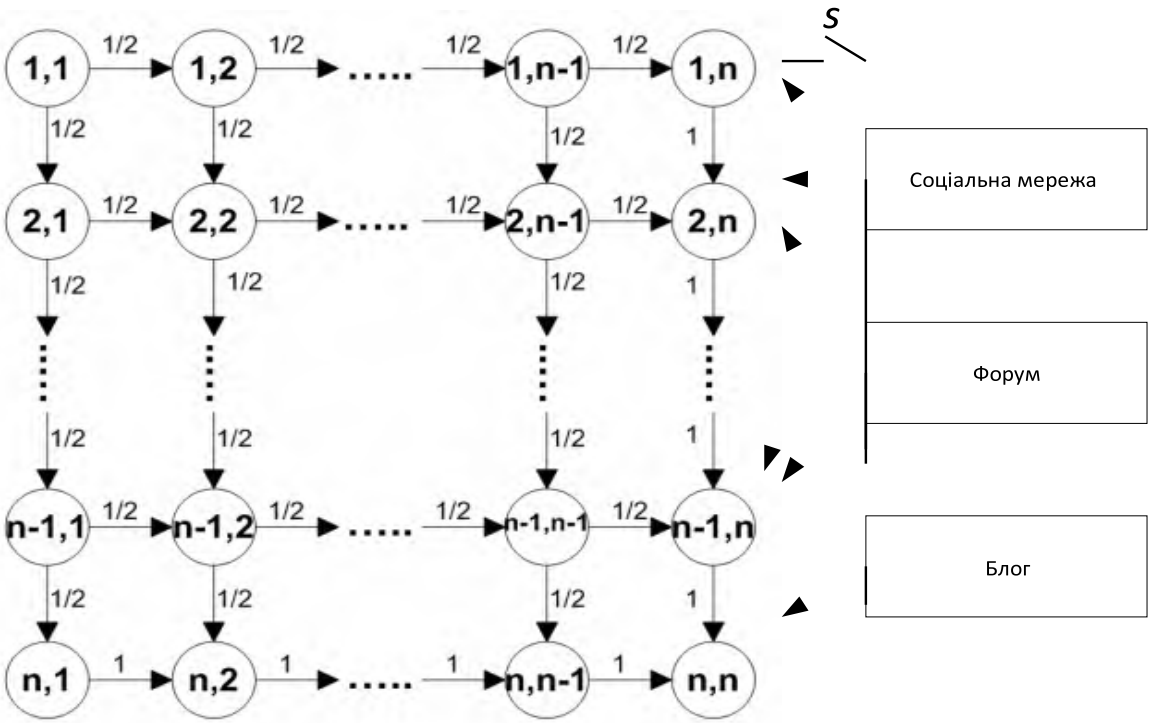
Загальна модель яка описує функціонування алгоритму PageRank та формування рейтингу сайтів



Як видно із моделі – рейтинг будь-якого сайту пов'язаний та формується на основі інших сайтів, навіть якщо вони з ним не зв’язані на пряму. Найбільший рейтинг, тобто ранг має вебсайт B, оскільки на нього посилається багато другорядних сайтів.

Також на сайт може бути лише одне посилання, але воно може бути більш вагомішим ніж десятки інших посилань на інший сайт.

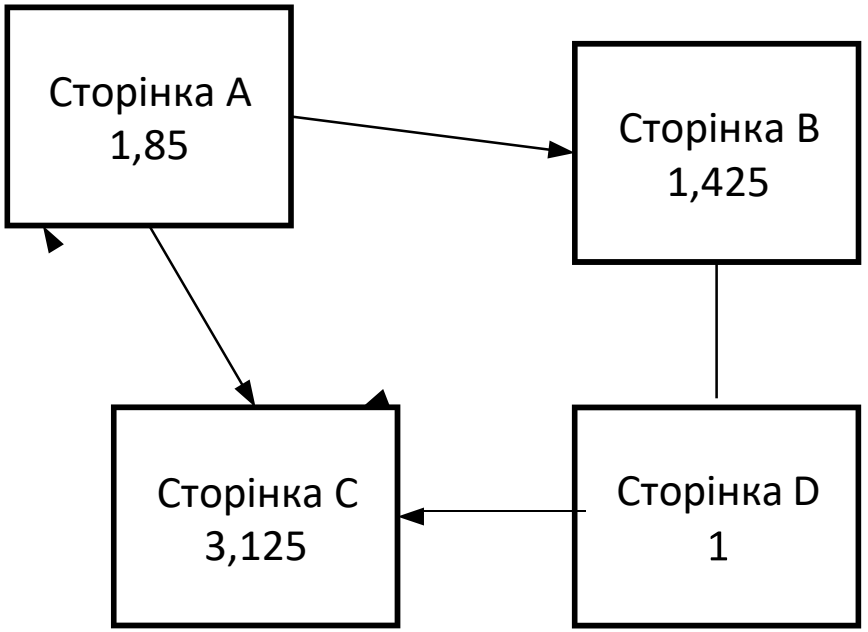
Матрична модель модифікованого алгоритму в мережі з імовірностями переходів



У кожної вебсторінки є вхідні і вихідні посилання. При цьому з кожної вебсторінки можна рівноймовірно перейти на будь-яку з вебсторінок, на які веде посилання. Тобто, якщо зі сторінки *i* одна з посилань веде на сторінку *j*, то ймовірність перейти зі сторінки *i* на *j* дорівнює $p_{ij} = \frac{1}{n_j}$ $p_{ij} = \frac{1}{n_i}$, де n_j – кількість посилань зі сторінки *i*.

В модифікованій моделі до уваги береться коефіцієнт *s*, який рівний 0,15, та враховується кількість посилань із форумів та соціальних мереж на певний сайт.

Математична модель алгоритму PageRank



$$PR(A) = (1-d) + d\left(\frac{PR(T_1)}{C(T_1)} + \frac{PR(T_n)}{C(T_n)}\right)$$

де $:PR(A)$ - ваговий коефіцієнт сторінки A,

D - коефіцієнт затухання який Google пропонує встановити рівним $0{,}85$,

$PR(T_1)$ - ваговий коефіцієнт PageRank сторінки, що посилається на сторінку A ,

$C(T_1)$ - кількість посилань із цього ресурсу,

$\frac{PR(T_1)}{C(T_n)}$ - для кожного ресурсу який вказує на ресурс A.

Подивимося на сторінку A. Її поточна вага PageRank дорівнює 1,85. Величина PageRank, доступна для передачі, після застосування затухання (0.85) складає:

$$1{,}85 \times 0{,}85 = 1.5725$$

Є два посилання зі сторінки, тому по завершенню ітерації ми додамо 0,78625 до ваги PageRank сторінки B і вазі PageRank сторінки C. Перейдемо до сторінки B. У неї є тільки одне посилання. Отже, вона передасть:

$$1{,}425 \times 0{,}85 = 1{,}21125$$

сторінці C, коли ми завершимо всі обчислення з посиланнями.

Сторінка C також має одне посилання, але при цьому володіє великою вагою 3,125 PageRank. Тому вона передасть:

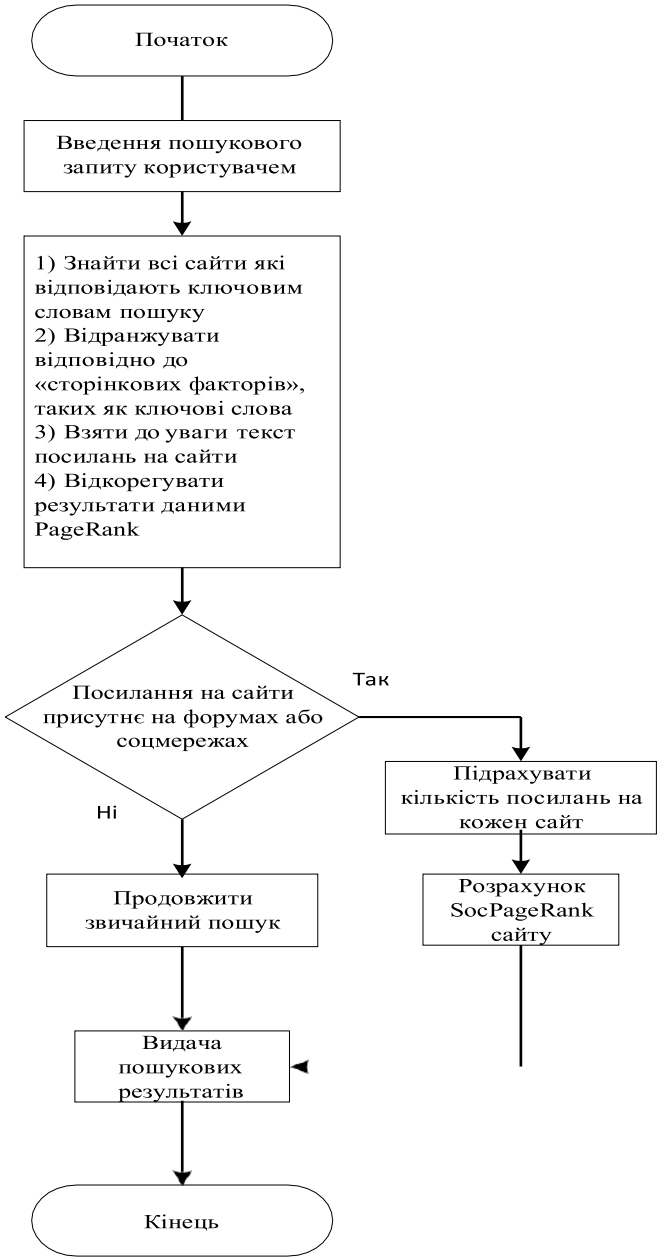
$$3{,}125 \times 0{,}85 = 2{,}65625$$

сторінці A.

Сторінка D має одне посилання, тому вона передає *0,85* сторінці C.

Отже рейтинги будуть наступні: A = 4,50625; B = 2,21125; C = 5,9725; D = 1.

Модифікований алгоритм та формула розрахунку ModPR



Модифікована формула алгоритму PageRank

$$PR(A) = (1-d) + d\left(\frac{PR(T_1)}{C(T_1)} + \frac{PR(T_n)}{C(T_n)}\right) + s \times T(C_s) \qquad (1)$$

де :$PR(A)$ - вага сторінки A,

D - коефіцієнт затухання який зазвичай встановлюють рівним 0,85,

$PR(T_1)$ - вага PageRank сторінки, що посилається на сторінку A,

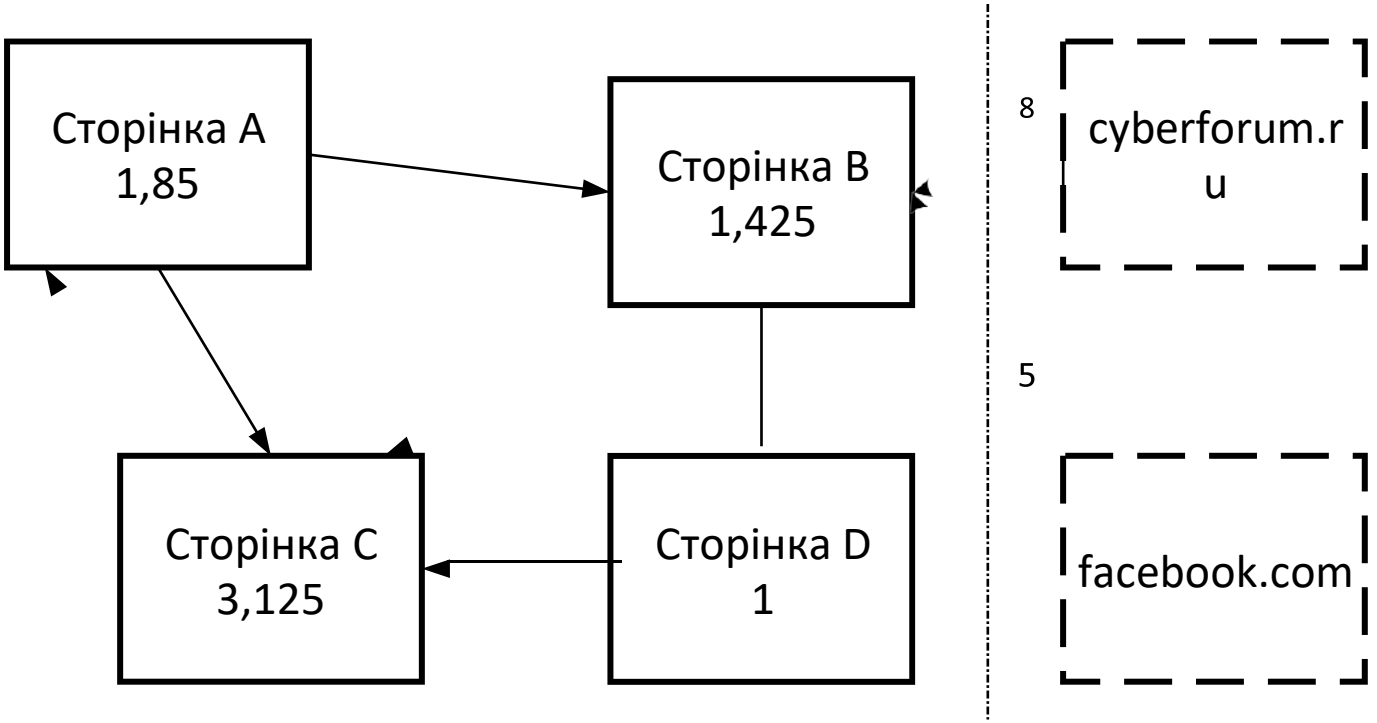
$C(T_1)$ кількість посилань із цієї сторінки,

$\frac{PR(T_1)}{C(T_n)}$ - для кожної сторінки яка вказує на сторінку A.

$T(C_S)$ кількість посилань із форума або соцмережі

s - коефіцієнт ModPR, рівний 0,15

Розрахований рейтинг PageRank веб-сайтів за моделлю після модифікації



Як видно із моделі вище, на сайт B йдуть 5 посилань із соцмережі та 8 посилань із форуму. Отже використовуючи модифіковану формулу отримаємо реальний рейтинг:

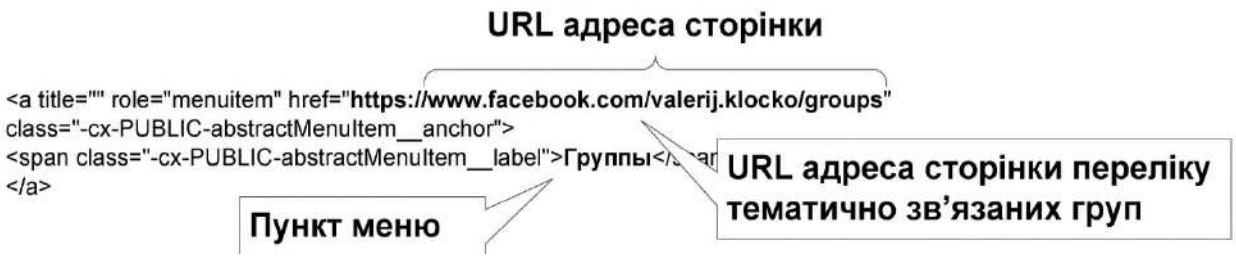
$$B = 2{,}21125 + 0{,}15 \times 13 = 4{,}16125$$

Після використання модифікованого алгоритму рейтинг сайту B значно підвищиться, звичайно він не витіснить сайти A та C, але це дасть йому змогу піднятись вище у списку видачі результатів.

АЛГОРИТМИ ПОШУКУ ГРУП У СОЦІАЛЬНИХ МЕРЕЖАХ

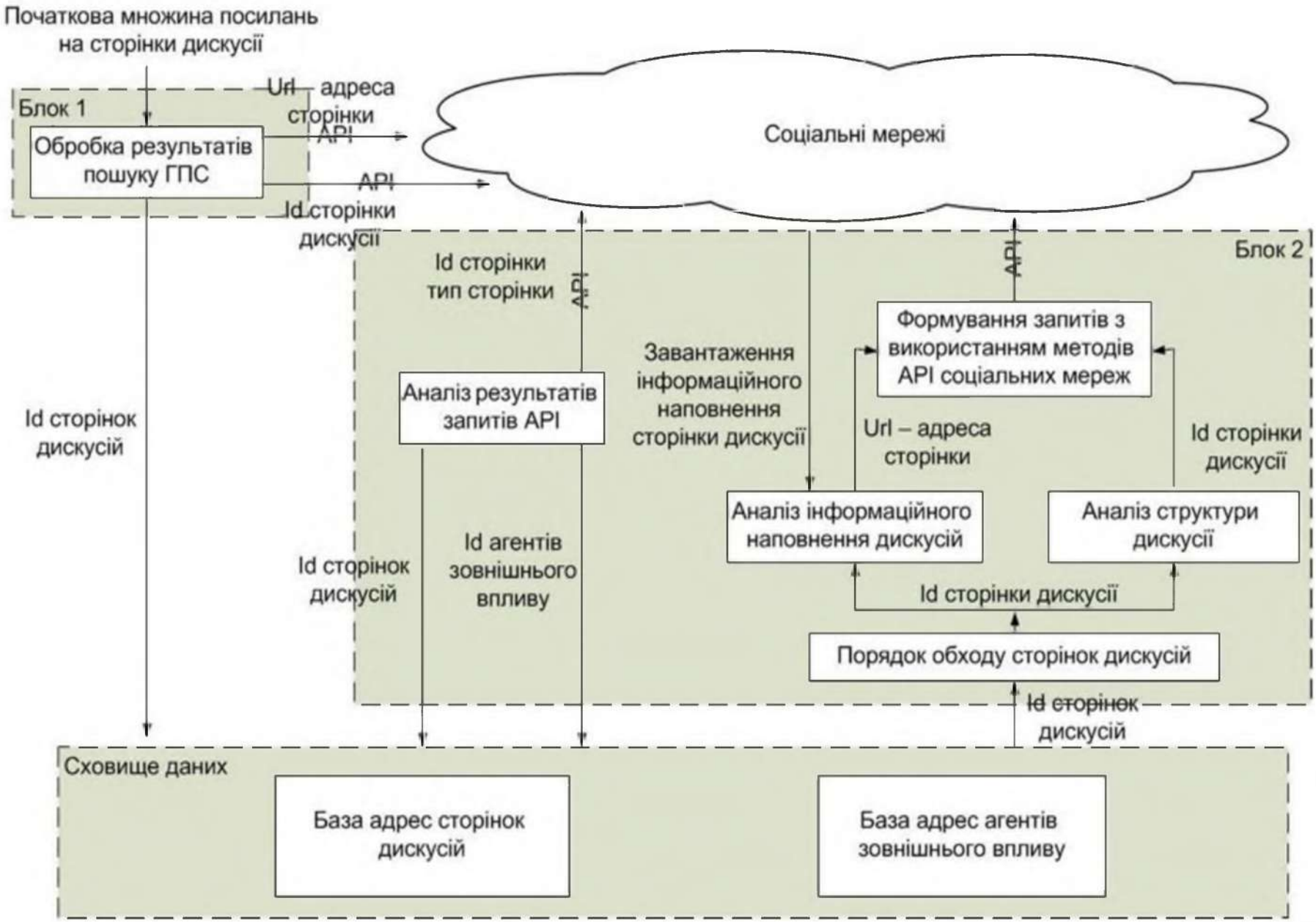
Формалізований запит для виявлення сторінок у соціальній мережі «Facebook»

Аналіз HTML-коду сторінки для виявлення URL-адреси сторінки переліку тематично зв'язаних груп



Схематичне зображення алгоритму пошуку

ГЛИБИННИЙ ПОШУК ДИСКУСІЙ У СОЦІАЛЬНИХ МЕРЕЖАХ



ВИСНОВКИ

У даній магістерській роботі виділено недоліки сучасних інформаційно-пошукових систем, які використовуються для пошуку та роботи з інформацією у мережі Інтернет. В роботі вирішено наукове завдання – розроблено наукові основи моделювання роботи інформаційно-пошукових систем. Мета магістерського дослідження полягає в підвищені пертинентності результатів пошуку за рахунок модифікації алгоритму ранжування Google – PageRank. За допомогою вдосконаленого алгоритму ранжування ModPR, корисні, актуальні та малопопулярні сайти які часто обговорюють користувачі в тематичних соціальних групах матимуть змогу отримувати свій реальний рейтинг.

Основні результати магістерської роботи є такими:

1. Проаналізовано та досліджено принципи пошуку інформації в Google.
2. Розглянуто математичну модель алгоритму існуючого алгоритму ранжування.
3. Вдосконалено математичну модель ранжування сайтів пошукової системи Google, за рахунок враховування популярності сторінки в тематичних соціальних спільнотах.
4. Розроблено метод підвищення пертинентності результату пошуку за рахунок вдосконалення алгоритму ранжування та індексації сайтів.
5. Розроблено алгоритми пошуку сторінок груп у соціальних мережах з використанням розширених можливостей глобальних пошукових систем та запитів API-методів, які дають змогу виявити сторінки груп у соціальних мережах відповідно їх інформаційного наповнення.
6. Вдосконалено метод виявлення інформаційного впливу соціальних мереж на індексацію сторінки та формування переліку зовнішніх гіперпосилань, для зміни коефіцієнтів ранжування відповідних сторінок.
7. Кількість пертинентних посилань серед перших десяти, одержаних унаслідок пошуку, збільшилась в середньому на 2.
8. Проведено практичне дослідження роботи методу.
9. Описано вимоги користувача, функціональні вимоги та можливості розробленої системи, описано обґрунтування вибору стеку технологій.

UNICHECK

Ім'я користувача:
Kafedra TMIT KhNU

ID перевірки:
1005403220

Дата перевірки:
08.12.2020 16:33:20 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
08.12.2020 16:38:03 EET

ID користувача:
100005657

Назва документа: Мурах

Кількість сторінок: 80 Кількість слів: 15226 Кількість символів: 115616 Розмір файлу: 2.59 MB ID файлу: 1005695091

246 слів позначені як "вилучені" та не враховуються у підрахунку слів

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

0.6%
Схожість

Найбільша схожість: 0.21% з джерелом з Бібліотеки (ID файлу: 1005683463)

0.21% Джерела з Інтернету 14 ..... Сторінка 82

0.21% Джерела з Бібліотеки 1 ..... Сторінка 82

0% Цитат

Цитати 1 ..... Сторінка 83

Посилання 1 ..... Сторінка 83

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 3

Підозріле форматування 13 сторінок

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 2.0%

Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 10%**

	Документ		Суммарное совпадение по Базе Данных	
ID: 80947 Название: Метод підвищення пертинентності результату пошуку за рахунок вдосконалення алгоритму ранжування та індексації сайтів Добавлено в БД: 2020-11-23 Авторы: Мурах Богдан Ростиславович Руководители: Муляр Ігор Володимирович Консультанты: Оппоненты:	Символы	Лексемы	Символы	Лексемы
	103181	829	4188 (4%)	33 (4%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

Завідувачу кафедри ТМІТ
д-р.техн.наук Підченку С.К.

Мурза Богдан Ростиславович
ПІБ здобувача вищої освіти

ФПКТС, 2 курсу, групи ПМм-19-1

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіатоповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів(Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

09.12.2020
дата

підпис

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА ДИПЛОМНУ РОБОТУ

Магістр Мурах Богдан Ростиславович

Тема Метод підвищення пертинентності результату пошуку за рахунок вдосконалення алгоритму ранжування та індексації сайтів

Спеціальність 113 – Прикладна математика

Обсяг дипломної роботи:

Кількість листів креслень 11; кількість сторінок записки 82

1.Короткий зміст ДР та прийнятих рішень У даній магістерській роботі виділено недоліки сучасних інформаційно-пошукових систем, які використовуються для пошуку та роботи з інформацією у мережі Інтернет. В роботі вирішено наукове завдання – розроблено наукові основи моделювання роботи інформаційно-пошукових систем. Мета магістерського дослідження полягає в підвищені пертинентності результатів пошуку за рахунок модифікації алгоритму ранжування Google – PageRank.

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

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У вступі висвітлюється актуальність теми роботи, дається аналіз досліджуваної проблеми і обгрунтовується застосовуваний підхід до її вирішення, формулюються цілі і завдання дослідження, описується наукова новизна і практична значимість отриманих результатів. У першому розділі розглядаються питання аналіз існуючих підходів до пошуку інформації. Наступні розділи присвячені розробці методу Метод підвищення пертинентності результату пошуку за рахунок вдосконалення алгоритму ранжування та індексації сайтів

4. Позитивні сторони роботи _Дипломна робота містить ряд інноваційних рішень, зокрема , була проведена робота над удосконалення алгоритму ранжування , робота якого полягає в підвищенні пертинетності результатів пошуку. Розроблено методи виявлення впливу соціальних мереж на індексацію сторінок на формувань переліку зовнішніх гіперпосилань_

5. Негативні сторони роботи Полягають в небажанні і ліні клієнтів спілкуватись на форумах соціальних мережах. Часто сайти обговорюються в живому спілкуванні але немає цифрового сліду який ми б могли б відслідковувати і на ньому бувати рейтинги сайтів . Аналіз аудіо та відео повідомлень які б могли збільшувати рейтинг сайтів.

6. Оцінка графічного оформлення та пояснювальної записки роботи _Графічне оформлення повністю виконане відповідно теми дипломної роботи з дотриманням стандартів і всіх вимог. Пояснювальна записка відповідає нормам для її оформлення

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

8. Інші зауваження

9. Оцінка дипломної роботи Розглянувши позитивні та негативні сторони представленої дипломної роботи, можна зробити висновок, робота дає вирішення поставленим задачам і вона заслуговує оцінку «добре».

РЕЦЕНЗЕНТ (прізвище, ім'я, по-батькові, посада, місце роботи) _____
к.т.н.доц.кафедри кібербезпеки та комп'ютерних систем і мереж Кльоц Ю.П

"9" 12 2020 р. _____
(підпис)