**РАНЖУВАННЯ В ІНФОРМАЦІЙНО-ПОШУКОВИХ СИСТЕМАХ НА ОСНОВІ СОЦІАЛЬНИХ СЕРВІСІВ**

*Кучеренко А.В. ст.викл., Полтавский національний технічний*

*університет ім. Ю.Кондратюка*

*Носко Н.В., Полтавский національний технічний*

*університет ім. Ю.Кондратюка*

*Стаття присвячена задачі ранжуванню в інформаційно-пошукових системах на основі соціальних сервісів. Основною проблемою ранжування являється пошук потрібного в колекціях документів. Тобто те, які документи будуть видані користувачу у відповідь на конкретний запит, саме це і визначає функція ранжування, або функція схожості. Метою слугує – дослідження методів застосування статичного ранжування в інформаційно-пошукових системах, а також розробка моделі ранжування повідомлень блогів на основі соціальних сервісів для покращення результатів пошуку.*

**ВСТУП**

Пошук потрібного результату в безмежних колекціях документів є важливим завданням. Про це свідчить як велика кількість пошукових систем, так і їх постійний розвиток. Колекції документів можуть бути різних типів: блоги, новинні стрічки, наукові статті або безлічі веб-сторінок. Пошукові системи, такі як Google або Yahoо оперують з останнім типом колекцій.

Принцип роботи пошукової системи наступний: користувач вводить запит, після чого система повертає ті документи з колекції, які найкращим чином відповідають запиту. Як правило, в традиційних пошукових системах ранжування документів (визначення релевантності документа на вимогу) проводиться на основі статистичної інформації про безліч слів у запиті і в документі. На відміну від пошуку в базі даних, де результатом є безліч записів з ключами, що задовольняють логічній умові, результат пошуку в колекції документів не визначений точно. Тому вводяться параметри точності і повноти, що відображають якість пошуку.

Більшість користувачів пошукових систем ніколи не замислювалися про принцип роботи пошукових систем, про схему обробки запитів користувачів, про те, з чого ці системи складаються і як функціонують.

Пошукова система - це програмно-апаратний комплекс, призначений для здійснення пошуку в мережі Інтернет, вона реагує на запит користувача, що задається у вигляді текстової фрази (пошукового запиту), видачею списку посилань на джерела інформації, в порядку релевантності (відповідно до вимоги). Пошукова машина, керуючись алгоритмом, аналізує контент сайту, з’ясовує наявність у складі контенту ключової фрази, приймає рішення про те, наскільки сайт відповідає пошуковому запиту, і залежно від ступеня відповідності присвоює сайту ту чи іншу позицію у видачі – вище або нижче.Для кожної пошукової системи розробляються свої алгоритми.

**ОЦІНКА РЕЗУЛЬТАТІВ РОБОТИ СИСТЕМИ**

**ІНФОРМАЦІЙНОГО ПОШУКУ**

Маючи деяку модель ранжирування в інформаційно-пошуковій системі, необхідно вміти відповідати на питання, наскільки ефективно працює конкретна фукнція ранжування.

Для оцінки ефективності роботи пошукової системи потрібні три сутності:

1. Колекція документів

2. Колекція тестових запитів

3. Дані про те, які документи в колекції є релевантними для кожного тестового запиту

Стандартний підхід полягає в тому, що для пари документ-запит відомо, чи є він релевантним чи ні, і не більше того. Тестова колекція повинна мати розумний розмір, щоб середня величина ефективності мала сенс. За необхідний мінімум приймається кількість в 50 запитів. Для прикладу наведемо одну широко використовувану тестову колекцію TREC. Вона розробляється в Національному Інституті Стандартів і Технологій в США. Колекція має поділ за типами розв'язуваних завдань, при цьому всього в ній міститься 1.89 мільйонів документів (в основному, новинних статей) і 450 різних тестових запитів, а також оцінки релевантності кожної пари документ - запит.

Для оцінки результатів ранжированного пошуку в TREC часто використовується єдина величина, яка називається MAP (mean average precision). Сенс цієї величини в тому, що для кожного запиту підраховується середнє серед значень точності, отриманих після кожного нового повернутого системою документа. Це значення називається середня точність (AP, average precision). Для отримання MAP, AP береться середнє по всіх тестових запитам.



де  , - підмножина  від документа  до документа . У даній роботі MAP буде використовуватися для оцінки результатів роботи системи.

**ДОСЛІДЖЕННЯ ТА ПОБУДОВА РОЗВ´ЯЗКУ ЗАДАЧІ**

Основним завданням є поліпшення якості пошуку за рахунок використання лічильників соціальних сервісів в якості статичної ознаки для ранжування.

Як вже зазначалося, соціальні сервіси надають кнопки, які поміщаються на сторінку поруч з текстом повідомлення блогу і дозволяють користувачеві єдину дію - натиснути на цю кнопку, причому лише один раз. І мета, яку переслідує дана дія, у всіх випадках однакова: зазначити, що дане повідомлення сподобалося, було корисно користувачеві і що він має намір поділитися посиланням на нього з друзями - користувачами обраного сервісу. Digg, facebook, twitter - всі вони надають зазначену функціональність. Окрім іншого, кнопки різних сервісів, як правило, знаходяться поруч один з одним і мають однаковий розмір. Ці міркування дозволяють зробити висновок про те, що принципової різниці між тим, який сервіс використовувати, немає. Тому порівняємо такі характеристики: зручність зовнішніх інтерфейсів (API) і кількість користувачів сервісу. Результати представлені в таблиці.

|  |  |  |
| --- | --- | --- |
| Назва сервісу | Приблизна кількість користувачів | API надає спеціальний запит, який повертає необхідні дані |
| Facebook | 1,32 мільярда | так |
| Twitter | 140 мільйонів | ні |
| Digg | 4 мільйона | ні |

Критерії порівняння обрані за такими принципами: кількість користувачів визначає масштаб впливу кожного сервісу, а відсутність в API спеціального запиту, що містить безпосередньо необхідні дані, збільшує складність реалізації відповідного модуля системи. Під API-запитом, про який йде мова в третьому стовпці таблиці, розуміється можливість отримувати потрібні дані не методом підрахунку, виконуючи декілька запитів і отримуючи зайві дані (наприклад, про ідентифікатори користувачів, що натиснули потрібну кнопку), а виконуючи єдиний запит. Виходячи з наведеного порівняння, можна зробити висновок, що використання сервісу facebook найбільш вигідно. Так як безпосередньо сервіс facebook, що надає потрібні нам дані називається facebook like button, а саме значення цього лічильника - likes count.

**МОЖЛИВА РЕАЛІЗАЦІЯ МЕТОДУ**

Розробка підсистеми з розширенням існуючої системи опису джерел інформації Blognoon.

Вибір програмного засобу був зроблений з урахуванням наступних вимог:

1. Підсистема повинна бути легко інтегрована в систему Blognoon.

2. Підсистема повинна працювати на різних платформах.

3. Реалізація повинна бути гнучкою і розширюваної.

Для реалізації було обрано такі інструменти і бібліотеки:

1. Java - об'єктно-орієнтована мова програмування

2. MySQL - реляційна система управління базами даних.

3. Hibernate - бібліотека, що надає функції об'єктно-реляційного проектування.

4. Spring MVC - бібліотека, що надає каркас для веб-додатків.

Частина компонентів розробленої підсистеми



**СХЕМА РОБОТИ**

На малюнку стрілками вказано, в якому напрямку і між якими компонентами передаються дані. Опишемо по кроках цикл обробки одного пошукового запиту.



1. Веб-сторінка формує get-запит, що містить текст запиту, заданого

користувачем системи.

2. Контролер (Controller) розбиває запит на терміни, формує пошукову

команду - об'єкт, що містить список термінів і параметри пошуку.

3. ExtendedSearcher отримує від системи повнотекстового пошуку Apache Lucene ранжований список повідомлень, що задовольняють запит.

4. ExtendedSearcher отримує від Сховища (Storage) дані про кожне повідомленні з списку, що фігурує на попередньому кроці. Ці дані містять і значення лічильника Facebook-likes.

5. ExtendedSearcher змінює порядок ранжирування повідомлень відповідно до

одним з алгоритмів комбінування різних ознак ранжирування.

6. ExtendedSearcher повертає список результатів Контролеру.

7. Контролер поміщає результати на сторінку і повертає браузеру користувача

HTML-код.

**ВИСНОВКИ**

У даній роботі розглянута ​​задача підвищення якості пошуку в системі Blognoon за допомогою розробки методу ранжування на основі соціальних сервісів. У процесі вирішення даного завдання були досягнуті наступні результати:

1. Досліджено методи ранжування в інформаційно-пошукових системах.

2. Розроблено модель статичного ранжування з використанням соціального сервісу facebook.

3. Реалізована та інтегрована в систему Blognoon підсистема, що відповідає за пошук відповідно до розробленої функції ранжування, а також збір і зберігання необхідних даних.

Порівнюючи з вихідною системою, було отримано поліпшення якості пошуку на 7% на тестовому множині.

**ДЖЕРЕЛЛА**

1) http://internetdevels.ua/blog/google-ranking-factors-internetdevels

2) Justin Zobel, Alistair Moffat. Exploring the Similarity Space // SIGIR forum 32(1). 1998. P. 18-34.

3) Chris Burges, Tal Shaked. Learning to rank using gradient descent // ICML '05 Proceedings of the 22nd international conference on Machine learning. 2005. P. 89-96.

4) Yoav Freund, Raj Iyer, Robert E. Schapire. An efficient boosting algorithm for combining preferences // The Journal of Machine Learning Research. 4. P. 933-969.

5) Ralf Herbrich, Thore Graepel, Klaus Obermayer. Large Margin Rank Boundaries for Ordinal Regression. Cumberland, England: MIT Press, 2000. P 115-132.