https://medium.com/@williamscott701/introduction-to-information-retrieval-series-436082826197

✅Мета

Основною ціллю галузі пошуку інформації - це задовольнити знайденими даними ввід користувача.

Тобто те, що робить система пошуку інформації - приймає ввід користувача, аналізує його, шукає по даних та відсилає результати знайдених документів.

Чому не можна використати ctrl+f? Якщо шукати потрібну інформацію таким чином, то у відповідь ми отримаємо інформацію про те, чи є таке слово, чи його немає, а також окремі слова в невпорядкованому вигляді. Проте потрібно також враховувати семантику, тобто шукати синоніми слова. Також слово може бути гомографом (слово з багатьма значеннями).

✅ Основні поняття та техніки організації пошуку інформації

https://towardsdatascience.com/tf-idf-for-document-ranking-from-scratch-in-python-on-real-world-dataset-796d339a4089

TF-DF(Term Frequency — Inverse Document Frequency) - техніка, яка рахує повторюваність слова в документі, визначаючи його вагу, яка відповідає за важливість слова в документі.

Наприклад, речення «Споруда дуже висока» має сприйматися комп'ютером як дані з числовим знченням, тому потрібно векторизувати текст, щоб комп'ютер міг краще зрозуміти запит. Векторизуючи документ, ми можемо знаходити потрібні документи, групувати їх за типом та рейтингом. Так само й відбувається пошук у Google, який сприймає веб-сторінки як документи, а текст, який вводить користувач, називається запитом. Коли приходить запит, google знайде найбільш доречні до запиту документи, впорядкує їх за доречністю і покаже користувачу у вигляді іншої веб-сторінки з посиланнями на знайдені результати. Цей процес виконується завдяки векторизації документів, хоча, звичайно, поверх нього розроблені ефективні алгоритми, які роблять такий пошук швидшим.

TF визначає частоту слова в документі, яка залежить від розміру документа та вживаності слова. Наприклад, слово «був» може з'являтися багато разів у документі, проте якщо ми візьмемо документ на 100 слів та документ на 10000 слів, то, звичайно, є велика ймовірність, що слово «був» частіше трапиться у більшому документі. Через

те, що важливість документа не залежить від його довжини, ми маємо провести нормалізацію щодо частоти, а саме поділити її на загальну кількість слів у документі.

Векторизуючи документи, важливо розуміти, що значення TF буде коливатися від 0 до 1 включно:

TF = кількість слова, яке ввів користувач, у документі/кількість слів у документі

Вирахувавши TF здається, що пошукова система має добре працювати. Проте якщо користувач введе занадто вживані слова, які обов'язково трапляться у будь-якому документі, то результат пошуку буде неефективним. Для цього потрібно ще порахувати частоту появи слова у множині документів, тобто DF - кількість документів, у яких слово присутнє. Щоб нормалізувати DF, потрібно поділити його на загальну кількість документів у множині(нехай ця множина буде A).

Також існує таке поняття, як Inverse Document Frequency(IDF), яке вимірює інформативність введеного слова. Наприклад, слово «буде» є у кожному документі і тому значення IDF = A/DF буде дуже малим. Якщо A має, наприклад, 10000 документів, то IDF буде дуже великим. Тому беруть логарифм від значення. Також якщо слово не трапляєтсья в жодному документі, то DF = 0, а щоб не відбувалося ділення на нуль, отримаємо формулу:

IDF = log(A/(DF + 1))

Перемноживши TF та DF, отримаємо оцінку доречності документа відповідно до вводу і далі за нею можемо проводити сортування:

TF-IDF(t, d) = TF(t, d) * log(N/(DF + 1))

Автор цих двох дописів спеціалізується в нейронних мережах, пошуку інформації, науці про дані та штучному інтелекті.

http://www.ijcncs.org/published/volume3/issue9/p3_3-9.pdf

Є два основні фактори для того, щоб визначити якість пошуку інформації:
Точність (відсоток доречних документів, знайдені за короткий період часу)
Кількість запитів (відсоток доречних документів, які були насправді знайдені та проаналізовані)

Є три основні процеси під час пошуку інформації: знайдені дані, ввід користувача та порівняння двох вищезгаданих.

Спочатку потрібно взяти інформацію від користувача, а потім, базуючись на вводі, почати пошук. Тут пропонується індексація для пошуку та представлення інформації. Потім проводиться порівняння знайденого та отриманого та виводиться результат пошуку користувачу.

✅Моделі пошуку даних:

1️⃣Булева модель

У цій моделі документ асоціюється з множиною ключових слів та зберігається як доречний або не доречний. Пошук керується трьома булевими операторами: AND, OR або NOT.

2️⃣Векторна просторова модель

У цій моделі документи сортуються залежно від їх подібності та вводу користувача. Документи та ввід представляються як вектори та для визначення оцінки його доречності, рахується кут між цими векторами.

3️⃣Ймовірнісна модель

4️⃣Inference Network Model

Працює подібно до векторної моделі, адже вираховує вагу запиту в документі та порівнює ваги доречних документів.

✅Алгоритми пошуку:

1️⃣Лінійний пошук (проходиться по даних по порядку)

2️⃣«Brute force» пошук систематично генерує всіх можливих кандитатів і перевіряє, чи цей документ відповідає запиту. Цей пошук простий та завжди знайде відповідні дані, якщо такі існують

3️⃣Бінарний пошук, який полягає у повторному діленні на половину посортованих ключів даних та переході до меншої або до більшої половини даних.

Автори цього допису є іранськими професорами у галузі комп’ютерних наук в одному з університетів. Спеціалізуються не лише в галузі пошуку інформації, а й у штучному інтелекті, нейронних мережах та ін.

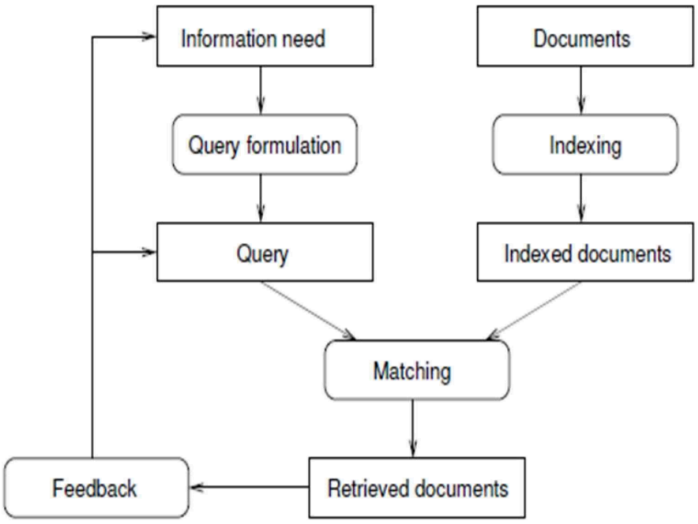

Fig 1. Information retrieval processes

https://www.tutorialspoint.com/natural_language_processing/natural_language_processing_information_retrieval.htm

Типи моделей інформаційного пошуку:

Types of Information Retrieval (IR) Model

An information model (IR) model can be classified into the following three models −

Classical IR Model

It is the simplest and easy to implement IR model. This model is based on mathematical knowledge that was easily recognized and understood as well. Boolean, Vector and Probabilistic are the three classical IR models.

Non-Classical IR Model

It is completely opposite to classical IR model. Such kind of IR models are based on principles other than similarity, probability, Boolean operations. Information logic model, situation theory model and interaction models are the examples of non-classical IR model.

Alternative IR Model

It is the enhancement of classical IR model making use of some specific techniques from some other fields. Cluster model, fuzzy model and latent semantic indexing (LSI) models are the example of alternative IR model.

(https://www.researchgate.net/publication/242403883_Information_Retrieval_Techniques)

Оскільки кількість даних зростає з кожним днем, важливо розуміти яка техніка підійде для їх пошуку. Існує кілька методів, які допомагають у цьому.

1. Аналіз синтаксису. Полягає у тому, що залежно від вводу користувача, знаходяться та індексуються ключові слова. Ця техніка не завжди дає доречні результати або ж їх багато через часту вживаність введених слів.

2. Пошук по метаданих. Тобто цей пошук здійснюється по опису, який представляється кожним документом.

3. Концептуальні графи. Представлення інформації за допомогою графів, в порівнянні з пошуком по метаданих, працює більш точно.

4. Індексація. Полягає в організації даних по важливості перед тим, як виводити її користувачу.

Автори допису є професорами Хорватського університету, які спеціалізуються у науці про дані та дотичних галузях.

Моделі пошуку інформації

1. Булева.

Основні поняття у галузі пошуку інформації

!!!! Wiki !!!!

Пошукова система є посередником між запитом та даними, які мають бути організовані відповідно до того, щоб пошук був швидким. За допомогою індексації шукаються відповідні дані та представляються користувачу у порядку спадання, від найбільш доречних до найменш доречних.

Для пошуку інформації проводиться підрахунок того, як часто слово з’являється у документі і порівнюється з результатами в інших документах

Автор допису є спеціалістом з декількох галузей таких, як інформаційна архітектура,
дизайн, пошуковий інтерфейс
https://www.analyticsvidhya.com/blog/2015/04/information-retrieval-system-explained/