

Ранжування текстових документів

Тарас Шевченко

Rails Reactor / Giphy

Компанії, які займаються пошуком

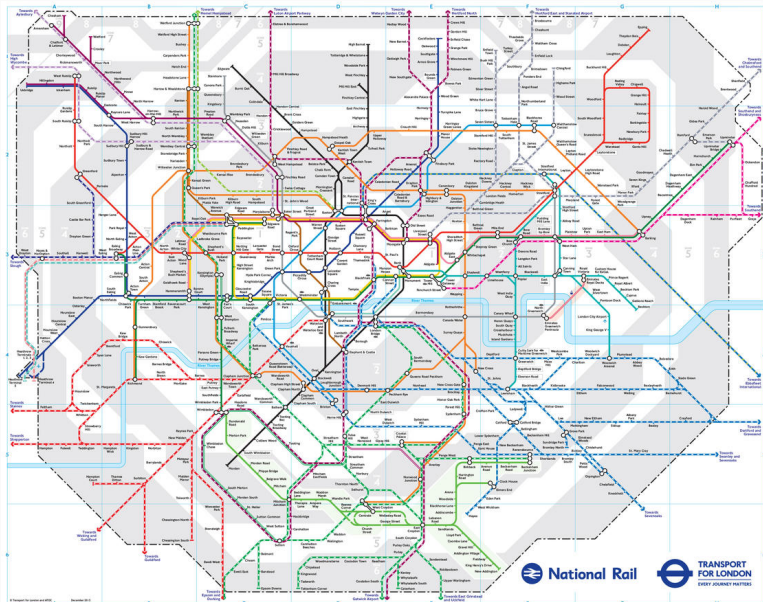


DuckDuckGo.

amazon




Як ознайомлювати себе із задачею/ковою базою




Як ознайомлювати себе із задачею/ковою базою


1. Пошук у ширину з випадкового місця.
2. Пошук у глибину з випадкового місця.
3. Декомпонувати на компоненти та опанувати кожен окремо, а потім зрозуміти як вони зв'язані.
4. Прослідкувати за формуванням результату.


Ранжування. Спискова видача. Google





cppcon 2020





 All

 Videos

 Images

 News

 Maps

 More

Settings

Tools

About 198,000 results (0.62 seconds)

cppcon.org › timeline2020 ▾

[CppCon 2020 Timeline | CppCon](#)

CppCon 2020 Timeline. 18-May, Presentation submission deadline. 23-Jun, Volunteer Grant Application Deadline. 29-Jun, Early bird registration ends. Call for ...

cppcon.org ▾

[CppCon | The C++ Conference](#)

CppCon represents an unparalleled opportunity for C++ authors to engage with ... The first of the these conferences will be **CppCon 2020** at the Gaylord Rockies ...

[CppCon 2019 Timeline](#) · [Registration](#) · [Classes](#) · [CppCon 2019 Program](#)

cppcon.org › call-for-proposals-for-cppcon-2020-classes ▾

[Call for Proposals for CppCon 2020 Classes | CppCon](#)

Nov 19, 2019 - The conference is asking for instructors to submit proposals for pre- and post-conferences classes and/or workshops to be taught in conjunction ...

isocpp.org › wiki › faq › conferences-worldwide ▾

[Conferences Worldwide, C++ FAQ - Standard C++](#)

NDC TechTown, 2020, Aug. ... Meeting C++, 2020, November 12-14, Berlin, Germany ...

CppCon, 2019, September 15-20, Aurora, Colorado, USA, Schedule ...

Ранжування. Спискова видача. Youtube

☰

YouTube UA

cppcon 2020

🔍

📺

📺

🔔

🏠
Home

🔥
Trending

📺
Subscriptions

📺
Library

CppCon
69.7K subscribers • 1,003 videos
Visit cppcon.org for details on next year's conference. CppCon sponsors have made it possible to record and freely distribute

SUBSCRIBED 🔔

🔧 FILTER

Latest from CppCon

CppCon 2019: Jim Radigan C++ Sanitizers and Fuzzing for the Windows Platform Using New Compilers...
CppCon • 8.6K views • 4 months ago
<http://CppCon.org> — Discussion & Comments: <https://www.reddit.com/r/cpp/> — Presentation Slides, PDFs, Source Code and other ...

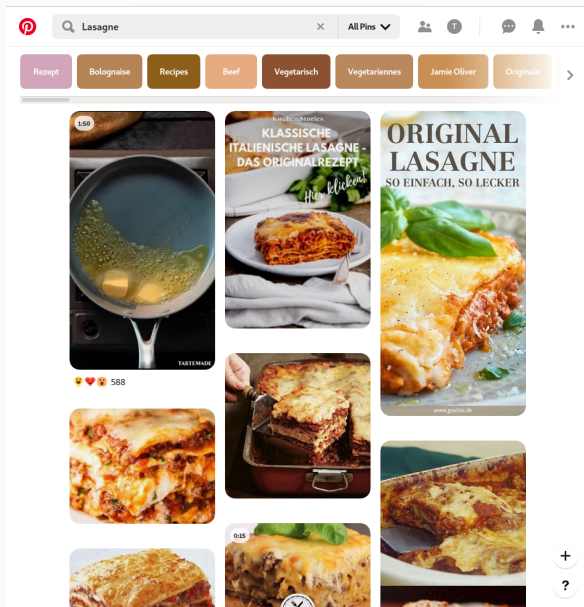
CppCon 2019: Jussi Pakkanen "Let's cmakeify the C++ standard library"
CppCon • 7.4K views • 4 months ago
<http://CppCon.org> — Discussion & Comments: <https://www.reddit.com/r/cpp/> — Presentation Slides, PDFs, Source Code and other ...

➦ MORE

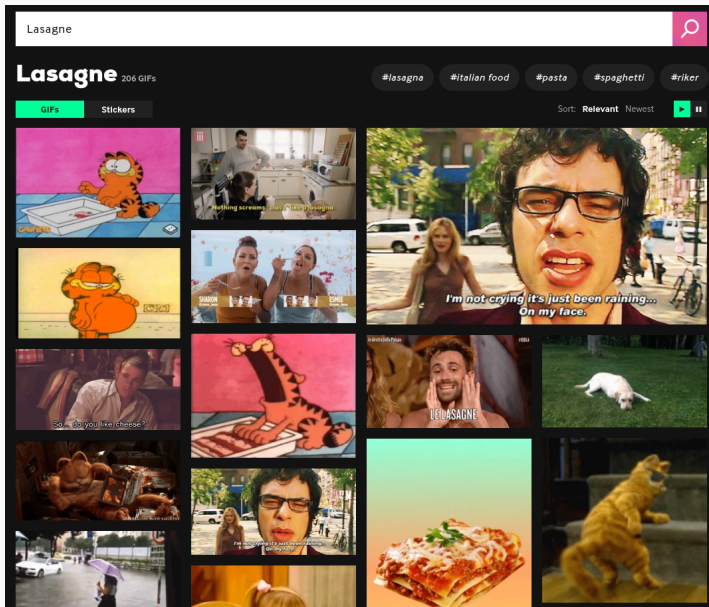
CppCon 2019: Bjarne Stroustrup "C++20: C++ at 40"
CppCon • 113K views • 5 months ago
<http://CppCon.org> — Discussion & Comments: <https://www.reddit.com/r/cpp/> — Presentation Slides, PDFs, Source Code and other ...

5

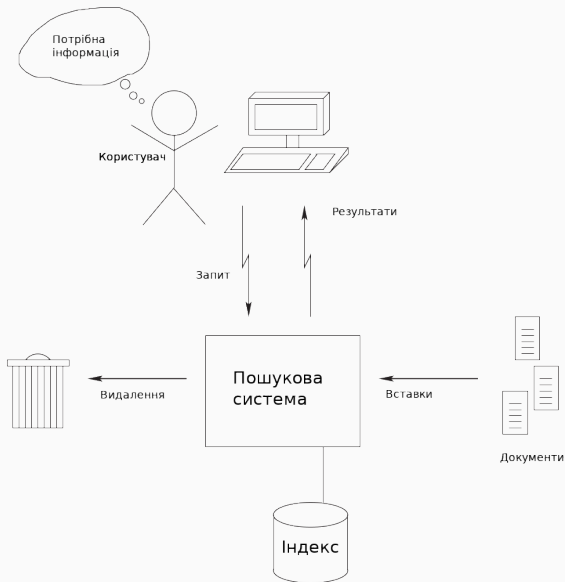
Ранжування. Динамічна сітка. Pinterest



Ранжування. Динамічна сітка. Giphy

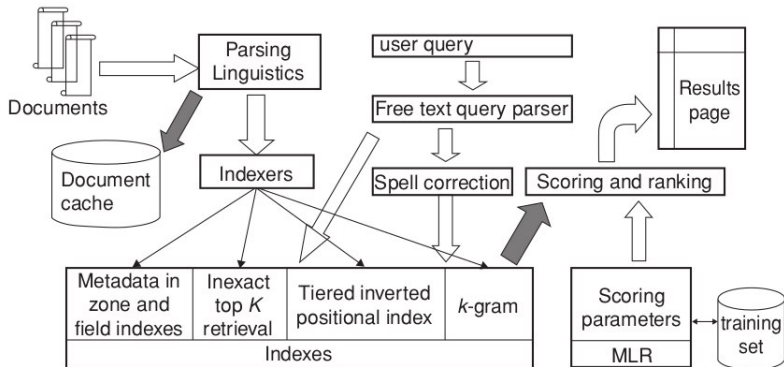


Спрощений погляд на пошук



1. Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008.
2. Christopher Manning, Hinrich Schütze, Foundations of Statistical Natural Language Processing, The MIT Press
3. Stefan Büttcher, Charles L. A. Clarke, Gordon V. Cormack, Information Retrieval: Implementing and Evaluating Search Engines, The MIT Press
4. Hang Li, A Short Introduction to Learning to Rank.

7.2 Components of an information retrieval system



Приклад результатів ранжування

Doc ID	Score
12	153.3
23	135.2
31	93.12
41	80.12
54	40.12
61	30.12
27	25.12
38	25.12
99	25.12

Найпростіша формула ранжування, яка може дати прийнятний результат

$$BM25(D, Q) = \sum_{i=1}^n \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} * \frac{f(q_i, D)(k_1 + 1)}{f(q_i, D) + k_1(1 - b + b \frac{|D|}{avgdL})}$$

Символ	Пояснення
N	кількість документів у колекції
n	кількість слів у запиті
q_i	i -те слово запиту
$n(q_i)$	кількість документів, в яких зустрічається слово q_i .
$f(q_i, D)$	скільки разів зустрічається слово q_i у документі.
$ D $	довжина документу D
$avgdL$	середня довжина документу
k_i	[1.2, 2.0]
b	0.75

1. не враховує позицію в документі;
2. не враховує особливостей мови

Okapi



Структура логу записів

Doc ID	Score	Query
12	153.3	чай
23	135.2	чай
31	93.12	чай
41	80.12	чай
54	40.12	чай
61	30.12	чай
27	25.12	чай
38	25.12	чай
99	25.12	чай

Структура логу записів

ts	DID	S	Query	QL	RID	SID	MID	A	UID	IP	CC
13	12	95	чай	UK	e1q	fbde	1	V	13	-	UA
13	23	94	чай	UK	e1q	fbde	1	V	13	-	UA
13	31	93	чай	UK	e1q	fbde	1	V	13	-	UA
13	41	80	чай	UK	e1q	fbde	1	V	13	-	UA
13	54	40	чай	UK	e1q	fbde	1	V	13	-	UA
13	61	30	чай	UK	e1q	fbde	1	V	13	-	UA
13	27	25	чай	UK	e1q	fbde	1	V	13	-	UA
13	38	24	чай	UK	e1q	fbde	1	V	13	-	UA
13	99	23	чай	UK	e1q	fbde	1	V	13	-	UA

Вибір ключу для сортування логу запитів

Ключ	Переваги
<ts, sid, rid>	Швидкість запису
<A, ts, sid, rid>	Відсутність необхідності сканувати дані для дій, які не беруть участь у запиті.
<A, ql, ts, sid, rid>	Дозволяє ефективніше програховувати запити різних мов.

The Search Result



I'd love to see the great views on Naboo again.

The Query: {search_query}

1. Please rate the relevance of the Pin to: {search_query}.

Make your judgement based primarily on the image, using the description below the image only as complementary information.

- ☐ Very Relevant
- ☐ Relevant
- ☐ Not relevant

2. Please check all boxes that apply.

- ☐ The Pin is missing
- ☐ The Pin contains adult or offensive content
- ☐ The Pin (image or text below) is in a foreign language

Note: If you're stuck on a certain Pin, please use [the training guide](#) to review a few nuanced cases.

Поради стосовно розмітки даних

1. випадково обирайте запити для тренувальної вибірки, враховуючи частість запитів;
2. обирайте документи так, щоб покрити якомога ширший діапазон значень ваших ознак, які будете використовувати;
3. власноруч розмітьте кілька сотень пар;
4. обирайте від трьох до 5-ти мір релевантності;
5. враховуйте специфіку вашої предметної області (можливо, варто внести варіант виду "аксесуар до пристрою, який згадувався у запиті").
6. застосовуйте статистичні методи оцінки узгодженості результатів

Поради стосовно розмітки даних

1. правильно підбирайте тестові запитання;
2. слідкуйте за процесом розмітки;
3. вимагайте як мінімум 3 голоси для оцінки якості пари;
4. обирайте не менше десяти документів для 1-го запиту.

Оцінка поточного алгоритму сортування

№	Doc ID	Query	Score	Rel
1	12	чай	153.3	1
2	23	чай	135.2	2
3	31	чай	93.12	3
4	41	чай	80.12	4

№	Doc ID	Query	Score	Rel
1	41	чай	80.12	4
2	31	чай	93.12	3
3	23	чай	135.2	2
4	12	чай	153.3	1

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{r_i} = \frac{1}{1} * \left(\frac{1}{3}\right)$$

$$nDCG_p = \frac{DCG_p}{IDCG_p} = \frac{\sum_{i=1}^p \frac{rel_i}{\log_2(i+1)}}{\sum_{i=1}^p \frac{rel_{x_i}}{\log_2(i+1)}} = \frac{\frac{1}{\log_2 2} + \frac{2}{\log_2 3} + \frac{3}{\log_2 4} + \frac{4}{\log_2 5}}{\frac{4}{\log_2 2} + \frac{3}{\log_2 3} + \frac{2}{\log_2 4} + \frac{1}{\log_2 5}} \approx \frac{5.48}{7.32} \approx 0.75$$

$$PairAccuracy = \frac{1}{|\{(i,j) | rel_i \prec rel_j\}|} \sum_{\{(i,j) | rel_i \prec rel_j\}} [score_i \prec score_j] =$$

$$= ([153.3 < 135.2] + [153.3 < 93.12] + [153.3 < 80.12] +$$

$$+ [135.2 < 93.12] + [135.2 < 80.12] + [93.12 < 80.12])/6 = 0.0$$

Оцінка поточного алгоритму сортування. Інші методи.

Точність $Precision = \frac{|relevant\ retrieved\ documents|}{|retrieved\ documents|} = \frac{tp}{tp+fp}$

Повнота $Recall = \frac{|relevant\ retrieved\ documents|}{|relevant\ documents|} = \frac{tp}{tp+fn}$

F-score $F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha) \frac{1}{R}}$

MAP $MAP = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} Precision(R_{jk})$

Ранжування. Формулювання задачі

- D - колекція документів
- Q - множина пошукових запитів
- $D_q \subseteq D$ - множина документів, знайдених за запитом q
- $X = Q \times D$ - об'єкти початкової вибірки (пари <запит, документ>)
- Y - множина оцінок пар
- $y : X \rightarrow R$ - оцінки релевантності, $y(q, d)$ - оцінка релевантності документу d для запиту q
- $x(q, d) = \{f_1(q, d), \dots, f_n(q, d)\}$
- $(q, d_i) \prec (q_j)$

Ознаки ранжування

Категорія	Категорія
Для документа	Квантилі CTR, Індекс якості документу, кількість посилань, довжина документу, авторитетність автора
Для запиту	Квантилі IDF, популярність запиту, сумарна частота слів
Для запиту та документа	Кількість кліків для запиту, LCS, LCCS, BM25, LCS-BM25, wclccs, LMIR, MinHitPos, MinBestSpanPos

- Поточковий - застосування алгоритмів регресії та класифікація на парах запит-документ
- Попарний - алгоритм фокусується не на апроксимації релевантності, а на попарних відношеннях релевантностей.
- Списковий - алгоритм фокусується на визначенні міри релевантності для документів, які видані для певного запиту.

Приклади алгоритмів ранжування

Підхід	Алгоритми
Поточковий	Лінійна регресія, преспетрон, косинусна функція врат, TFRank
Попарний	RankSVM, SortNet, RankNet, FRank, RankBoost, GBRank, MHR, LambdaRank
Списковий	SoftRank, Smooth Rank, AdaRank, ListNet, ListMLE, BoltzRank

Швидке нагадування про SVM

$$\sum_{i=1}^m (1 - y_i(\langle \vec{x}_i, \vec{w} \rangle - w_0))_+ + \frac{1}{2C} \|\vec{w}\|^2 \approx \quad (1)$$

$$\approx \sum_{i=1}^m \log_2 (1 + e^{1 - y_i(\langle \vec{x}_i, \vec{w} \rangle - w_0)}) + \frac{1}{2C} \|\vec{w}\|^2 \rightarrow \min_{w, w_0} \quad (2)$$

$$\vec{w} \in R^n - \text{вектор ваг} \quad (3)$$

$$w_0 \in R - \text{зміщення} \quad (4)$$

$$\vec{x}_i \in R^n - \text{вектори ознак} \quad (5)$$

$$y_i \in \{-1, 1\} - \text{мітки} \quad (6)$$

$$(\vec{x}_i, \vec{w}) - w_0 > 0 - \text{функція прийняття рішень} \quad (7)$$

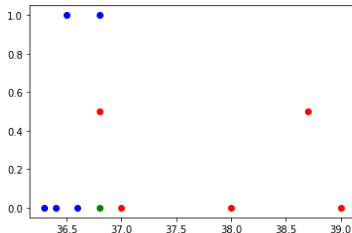
Іграшковий приклад

```
|: x = np.array([[36.3], [36.4], [36.6], [37.0], [38.0], [39.0]])
y = np.array([0, 0, 0, 1, 1, 1])
svc = LinearSVC(dual=False, C=2.0**31).fit(x, y)
assert np.all(svc.predict(x) == y)

plt.plot(x[y == 0], np.zeros(np.sum(y == 0)), "bo") # negative
plt.plot(x[y == 1], np.zeros(np.sum(y == 1)), "ro") # positive
middle = (-svc.intercept_[0] / svc.coef_[0])
assert 36.8 - 1.0e-6 < middle and middle < 36.8 + 1.0e-6
plt.plot(middle, [0], "go") # "hyperplane"

x_test = np.array([[36.5], middle - 1.0e-6, middle + 1.0e-6, [38.7]])
y_test = svc.predict(x_test)

plt.plot(x_test[y_test == 0], np.repeat(1, np.sum(y_test == 0)), "bo") # negative
plt.plot(x_test[y_test == 1], np.repeat(0.5, np.sum(y_test == 1)), "ro") # positive
```



Попарний підхід. Ranking SVM

Постановки задач SVM для попарного підходу

$$Q(a) = \frac{1}{2} \|\vec{w}\|^2 + C \sum_{i \prec j} (1 - \langle w, x_j - x_i \rangle)_+ \quad (8)$$

Еквівалентна постановка задачі у термінах квадратичного програмування:

$$\begin{cases} \frac{1}{2} \|\vec{w}\|^2 + C \sum_{i \prec j} \xi_{ij} \rightarrow \min_{\vec{w}, \xi}; \\ \langle w, x_j - x_i \rangle \geq 1 - \xi_{ij}, \quad i \prec j; \\ \xi_{ij} \geq 0, \quad i \prec j \end{cases}$$