پاسخ تمرین سری اول مبانی بازیابی اطلاعات و جستجوی وب

دکتر هدی مشایخی

مصطفی فضلی شهری – ۹۸۲۲۸۰۳

۱۰ آبان ۱۴۰۰

Q1

ایندکس معکوس (inverted index)و ماتریس رخداد (incidence matrix) را برای مستندات زیر رسم کنید .

سند1 : من درس بازیابی اطلاعات را در دی ماه پاس میکنم .

سند2 : یکی از دروس مهندسی کامپیوتر بازیابی اطلاعات است .

سند3 : من این ترم درس بازیابی اطلاعات دارم .

سند4 : این ترم بیست واحد دارم.

Q1.1 Incidence Matrix

	سند 1	سند 2	سند 3	سند 4
من	1	0	1	0
درس	1	0	0	0
بازیابی	1	1	1	0
اطلاعات	1	1	1	0
را	1	0	0	0
در	1	0	0	0
دی	1	0	0	0
ماه	1	0	0	0
پاس	1	0	0	0
می کنم	1	0	0	0
یکی	0	1	0	0
از	0	1	0	0
دروس	0	1	0	0
مهندسی کامپیوتر	0	1	0	0
است	0	1	0	0
این	0	0	1	1
ترم	0	0	1	1
دارم	0	0	1	1
بیست	0	0	0	1
واحد	0	0	0	1

Q1.2 Inverted Index

من	1 , 3
درس	1
بازیابی	1 , 2 , 3
اطلاعات	1 , 2 , 3
را	1
در	1
دی	1
ماه	1
پاس	1
می کنم	1
یکی	2
از	2
دروس	2
مهندسی کامپیوتر	2
است	2
این	3 , 4
ترم	3 , 4
دارم	3 , 4
بیست	4
واحد	4

Q2.1 Incidence Matrix

Retrieval AND Information
Doc1, Doc2, Doc3

Lesson OR Semester
Doc1, Doc 3, Doc4

Unit AND(~Have)
(Nothing)

Computer Engineering AND
~(Information AND Retrieval)
(Nothing)

	سند 1	سند 2	سند 3	سند 4
من	1	0	1	0
درس	1	0	0	0
بازیابی	1	1	1	0
اطلاعات	1	1	1	0
را	1	0	0	0
در	1	0	0	0
دی	1	0	0	0
ماه	1	0	0	0
پاس	1	0	0	0
می کنم	1	0	0	0
یکی	0	1	0	0
از	0	1	0	0
دروس	0	1	0	0
مهندسی کامپیوتر	0	1	0	0
است	0	1	0	0
این	0	0	1	1
ترم	0	0	1	1
دارم	0	0	1	1
بیست	0	0	0	1
واحد	0	0	0	1

Q2.2 Inverted Index

Retrieval AND Information
Doc1, Doc2, Doc3

Lesson OR Semester
Doc1, Doc 3, Doc4

Unit AND(~Have)
(Nothing)

Computer Engineering AND
~(Information AND Retrieval)
(Nothing)

من	1 , 3
درس	1
بازیابی	1 , 2 , 3
اطلاعات	1 , 2 , 3
را	1
در	1
دی	1
ماه	1
پاس	1
می کنم	1
یکی	2
از	2
دروس	2
مهندسی کامپیوتر	2
است	2
این	3 , 4
ترم	3 , 4
دارم	3 , 4
بیست	4
واحد	4

Q3

از سیستم های بازیابی اطلاعات می توان به Google, Yahoo, Bing, AOL, Ask.com, AltaVista اشاره کرد.
در ادامه به بررسی موتورجستجوگر ASK میپردازیم.
برای اینکار دو چک لیست را مورد بررسی قرار می‌دهیم،یکی برای تشخیص ساختمان و ویژگی های محتوایی و دیگری برای بررسی رکورد های Recallو Precision.
برای بررسی سرعت یک جستجوگر، سه مجموعه که برر روی 20 کوئری انتخاب شده آزمایش و اجرا شده است را بررسی می کنیم:

1) Question Search (QS)
2) Phrase Search (PS)
3) Keyword Search (KS)

فرمول های محاسباتی برای به دست اوردن درصد Precision و Recall بدین صورت به دست می آیند:

$$\textbf{Precision (percent)} = \frac{\text{Total percentage of relevant retrieved contents}}{T=(NR+RR)}$$

PS = Phrase Search
KS= Keyword Search
QS=Question Search

$$\textbf{Recall (percent)} = \frac{\text{Total percentage of relevant retrieved contents}}{(KS+PS+QS)-\Sigma R.R}$$

NR=Non-repetitive Record
RR = Repetitive Records
T = Total count of the relevant retrieved records based on KS

Q3

در بحث بررسی ساختمانی جستجو به اجزای مختلف نظیر قابلیت های جستجوی پیشرفته، جستجوی تصویر، جستجوی فیلم، اخبارو... ،
در بخش نمایش، نمایش اجزای سایت نظیر لینک سایت، تیتر مطلب، برجسته کردن عبارات جستجو شده و... ،
در بحث بحث کاربرپسند بودن، تم های جستجوگر، آسانی استفاده، بررسی و تعمیر لینک ها و اجزای بدون استفاده یا حذف شده و...، مورد بررسی قرار گرفته می شود که همه این ها با یکدیگر می توانند ساختار یک جستجوگر را تشکیل دهند.

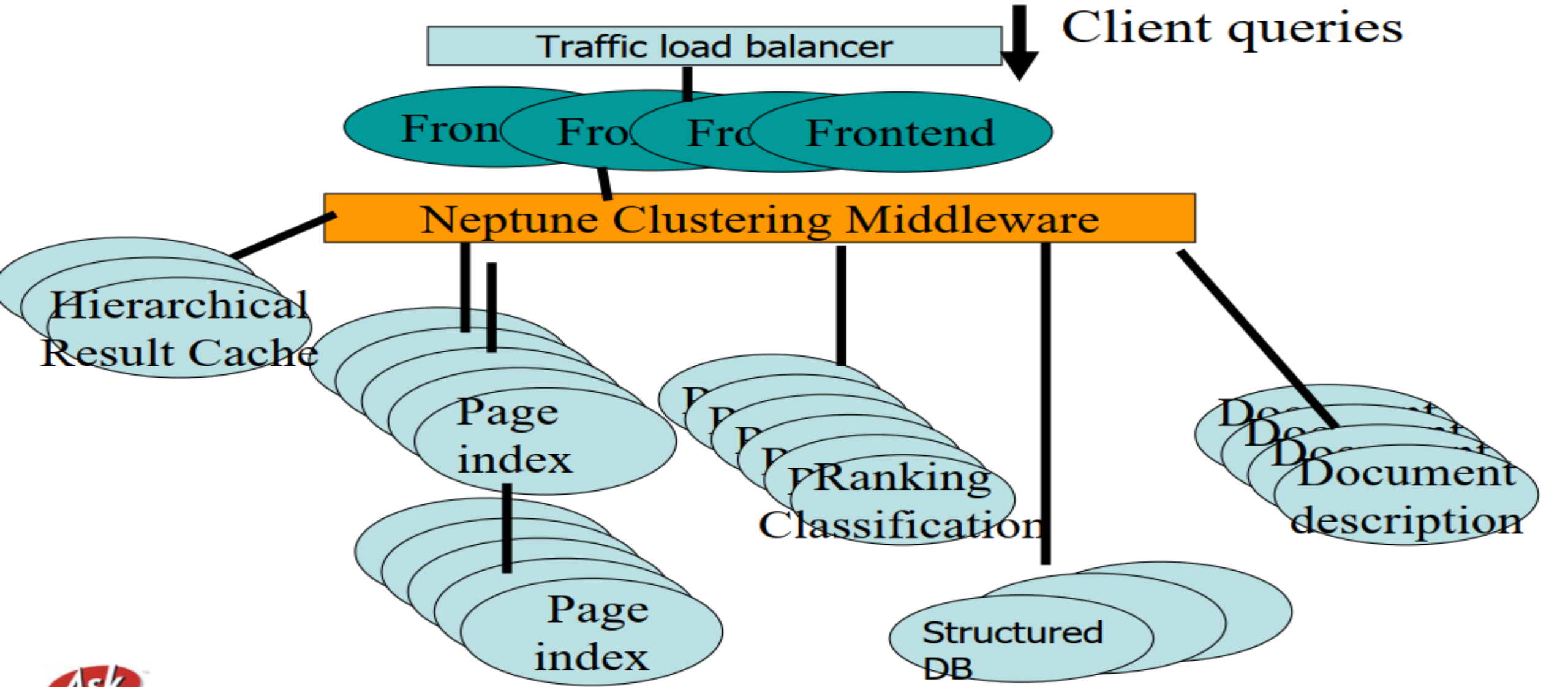
در قسمت پردازش و سرعت، کلیدواژه ها، کلیدواژه های مترادف، عبارات اصلی و سوالات مورد بررسی قرار میگیرند و با ترکیب کردن ان ها نسبت های Recallو Precision را محاسبه می کنند.

Steps of ExpertRank at Ask.com

1 Search the index for a query
2 Clustering for subject communities for matched results
3 local subject-specific mining
4 Ranking with knowledge and classification

ASK.COM Manual

Engine Architecture



Q4

(tangerine OR trees) AND (marmalade OR skies) AND (kaleidoscope OR eyes)
(87009 + 213312) ^ (107913 + 271658) ^ (46653 + 316812)
(300,321) ^ (379,571) ^ (360,165)
(1) ^ (3) ^ (2)

Q5

در ابتدا لیست هایی که سایز کمتری دارند را با همدیگر AND می کنیم.
اما باز هم بستگی به نوع توزیع posting لیست ها دارد، با اینکه سایز (tangerine OR trees) از (marmalade OR skies) کمتر است، اما ممکن است فاصله (پراکندگی) آن ها بسیار بیشترباشد و باید مراحل بیشتری را طی نماید که وجود OR را بررسی کند.
با این حال میتوانیم با استفاده از الگوریتم های مختلف مرتب سازی انجام داده و فاصله ابتدا و انتهای Index ها را بررسی کنیم و اگر بهینه بود از این روش استفاده کنیم.

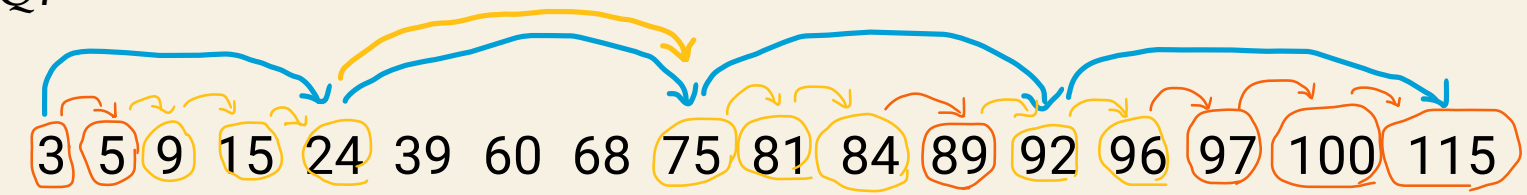
Term	Postings size
eyes	316812
kaleidoscope	46653
marmalade	107913
skies	271658
tangerine	87009
trees	213312

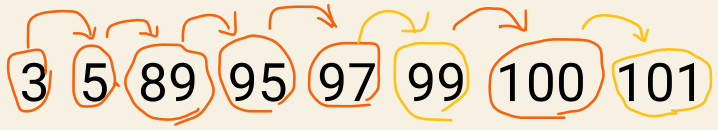
Q6

درستی و نادرستی عبارات زیر را مشخص کنید :
الف) stemmingسبب افزایش اندازه واژگان (vocabulary) می شود .
ب) Stemming باید در هنگام نمایه سازی Indexing فراخوانی شود نه در هنگام پردازش یک پرس وجو .
ج) در سیستم بازیابی بولین، stemmingهرگز مقدار Recallرا کاهش نمیدهد .

الف) نادرست
ب) نادرست
ج) درست

Q7





الف) با توجه به محل قرار گرفتن Skip pointer ها، تنها یکبار می توان از از skip استفاده کرد و از عبارت 24 به 75 پرش کرد.
ب) حداقل 14 مقایسه توسط این الگوریتم انجام می شود تا بتوان اشتراک های این دو لیست را به دست آورد.
ج) اگر بدون استفاده از اشاره گر های پرش این مقایسه را انجام می دادیم، نیاز داشتیم تا به اندازه m بار یعنی 17 بار مقایسه انجام می شد که 3 بار بیشتر از حالت همراه پرش است.

Q8

Doc1: I am a student, and I currently take IR course.
Doc2: I was a student; I have taken IR course.

I, 3:
<
<1:1, 6>;
<1, 5>;
>

Student, 2:
<
<3>;
<4>;
>

“I student” -> nothing
“student I” -> doc2

Q9

چگونه یک سیستم IRامی تواند استفاده از positional index و استفاده از لیست واژه ها stop words را با هم ترکیب کند؟ مشکل احتمالی در این فرآیند چیست و چگونه می توان آن را مدیریت کرد؟

کلمات ایست، کلمات بسیار پرتکرار هستند که ارزش ناچیزی در کمک به انتخاب اسناد مرتبط با نیاز کاربرد دارند.
برای ترکیب می توان کلمات ایست را هم به صورت positional index لیست کرد ولی این کار مشکلات متعددی را بوجود می آورد، مثلا گاهی کاربر از کلمات ایست متفاوتی برای جستجو استفاده می کند یا گاهی ممکن است کلمه ای که خود معنی بخصوص دارد به صورت کلمه ایست در نظر گرفته شود.
برای حل این مشکل می توانی ترکیب های دوتایی یا چندتایی BiWord کلمات را لیست کرد تا از این مشکلات جلوگیری شود و جستجوی کاربر با توجه به ترکیب های مختلف جستجو بررسی شود.

Q10

برای پرسوجوهای زیر آیا میتوانیم عملیات اشتراک را در زمان O(x+y) با در نظر گرفتن اینکه x و y طول postings lists برای Brutus و وCaesar اجرا کنیم؟ اگر اینطور نیست پس چگونه ما میتوانیم آن را محاسبه کنیم ؟

1. Brutus AND NOT Caesar

در اینجا ما به وارون Caesar نیاز داریم، پس y’ را باید محاسبه کنیم.
در واقع ما به به همه ی x+y’ موجود هم نیاز نداریم و با استفاده از الگوریتم های مختلف مانند پرش و... میتوانیم تا حد زیادی این پیمایش را بهبود ببخشیم، از طرفی لیست با سایز کمتر در AND بین دو لیست کافی است.

2. Brutus OR NOT Caesar

باز هم به y’ نیاز داریم ولی اینبار باید همه عناصر x و y’ را پیمایش کنیم تا بتوانیم هر دو لیست را با یکدیگر OR کنیم.