ARYA VARASTE

۱۴۰۰/۰۹/۱۲

بازیابی هوشمند اطلاعات

تمرین دوم

a.varaste.n@gmail.com

ایجاد شاخص

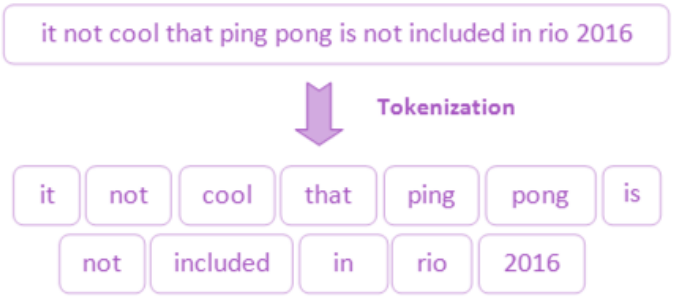
در هر زبانی، کلمات با توجه به نقشی که در جملات ایفا می کنند، به شکل‌های ظاهری متفاوتی خواهند بود. اما تمامی آن‌ها از یک ریشه ساخته می شوند. لذا در بسیاری از روش‌ها، ابتدا می‌بایست ریشه کلمات را پیدا کنیم. یکی از روش‌های متداول برای ریشه‌یابی کلمات، روش ریشه یابی Stemming است. الگوریتم‌های مختلفی جهت انجام عمل ریشه یابی وجود دارد که الگوریتم Porter از الگوریتم های معروف در زبان انگلیسی می باشد. این الگوریتم طبق یک سری قاعده‌ی منظم (مثلاً حذف حرف s در آخر کلماتِ جمع) می‌تواند ریشه‌ی کلمات را با دقتِ خوبی به دست آورد.

پس از نصب و راه‌اندازی گالاگو، به کمک دستورات گالاگو از روش Porter Stemmer برای ریشه‌یابی کلمات استفاده کردیم.

```
"stemmer" : ["porter"],
```

عمل Tokenization متن را مانند شکل زیر به توکن‌های تشکیل دهنده خود تبدیل می‌کند. این عمل را با دستورات زیر انجام می‌دهیم:

```
"tokenizer" : {
   "fields" : ["text","head"],
   "formats" : {
     "text"    : "string",
     "head"    : "string"
   }
 }
```



نهایتاً فرمت فایل که در ابتدا به صورت بدون فرمت بود را با کمک نرم افزار 7zip از سیستم‌عامل ویندوز اکسترکت کرده و به فایل txt. رسیدیم، سپس فایل حاصل شده را به فرمت trectext. که قالبی مشابه XML دارد بردیم. این تایپ از فایل‌ها برای اسناد و متونی مناسب است که لازم داریم قسمت‌های مختلف آن را بتوان جدا دید.

```
<DOC>
<DOCNO> AP890325-0001 </DOCNO>
<FILEID>AP-NR-03-25-89 0106EST</FILEID>
<FIRST>r a AM-People-Bridges 1stLd-Writethru a0733 03-25 0278</FIRST>
<SECOND>AM-People-Bridges, 1st Ld-Writethru, a0733,0282</SECOND>
<HEAD>Bridges Pleads Innocent To Attempted Murder Charge</HEAD>
<HEAD>Eds: SUBS lead to reflect that Bridges' role in the shooting has
not been established.</HEAD>
<DATELINE>LOS ANGELES (AP) </DATELINE>
<TEXT>
   Actor Todd Bridges, a star in the NBC comedy
``Diff'rent Strokes,'' pleaded innocent Friday to attempted murder
charges stemming from an incident in which he allegedly shot a
roommate.
   Bridges shot Kenneth Clay as many as eight times in a Feb. 2
argument that stemmed from Clay having borrowed Bridges' BMW
automobile, according to testimony in the case.
   Another man, Harvey Duckett, 30, who was with Bridges during the
episode, also faces the attempted murder charge. But Duckett, who
has pleaded no contest, is testifying in the case in exchange for
leniency from prosecutors.
   Bridges, 23, was bound over to Superior Court last week by
Municipal Judge David Horwitz, who also refused to reduce Bridges'
$2 million bail.
   On Friday, Superior Court Judge David Horowitz scheduled a
pretrial hearing, trial setting and bail motions for April 14.
   In addition to his role as Gary Coleman's protective older
brother on ``Diff'rent Strokes,'' Bridges has appeared in ABC's
``Fish'' and has been on various installment of ``Circus of the
Stars.''
   During a preliminary hearing earlier this month, Bridges'
attorney, Johnnie Cochran argued for a reduction in charges,
contending Bridges was too drugged to premeditate attempted
first-degree murder. Witnesses testified that the actor ingested
rock cocaine at least four times on the day of the shooting.
   Clay described Bridges as ``based out'' from freebasing or
smoking cocaine. ``It was the worst I seen him. He looked like his
eyes were about to jump out of his head,'' Clay testified.
</TEXT>
</DOC>
```

دستورات مربوط به تنظیماتی که در بالا بدان‌ها اشاره شد را در فایل indexSettings.json قرار داده و دستور زیر را در ترمینال اجرا می‌کنیم:

```
Galago/galago-3.16/core/target/appassembler/bin/galago build  /home/arya/Desktop/CA1-Resources/indexSettings.json
```

پس از گذشت حدوداً یک ساعت از اجرای دستور Build عمل شاخص‌گذاری با موفقیت به اتمام رسید.

```
Done Indexing.
  - 0.92 Hours
  - 55.18 Minutes
  - 3310.79 Seconds
Documents Indexed: 163912.
```

سوال ۱: بررسی روش‌های هموارسازی

ابزار گالاگو، به صورت پیش فرض بازیابی را به روش Query-Likelihood انجام می‌دهد. با توجه به کتاب مرجع اول درس، Query-Likelihood اینطور فرض شده که برای نمونه اگر کاربری سند d را دوست داشته باشد، چقدر احتمال دارد که برای بازیابی کردن آن کوئری q را وارد کند؟[1] در این تمرین با از روش بازیابی Query-Likelihood استفاده کرده و با متدهای گوناگونی اقدام به هموارسازی می‌کنیم تا احتمال رخداد کلمه‌های دیده نشده را با مقادیری بجز صفر تخمین بزنیم. در این راستا روش‌های خواسته شده در تمرین را به ترتیب اعمال کرده و تاثیر پارامترهای این روش‌ها را بر نتایج حاصل از بازیابی که با پارامترهای MAP و P@20 ارزیابی شده‌اند را بررسی کنیم و مقادیر بهینه برای این پارامترها را جهت بیشینه کردن معیارهای ارزیابی کشف کنیم.

روش اول: JM[2] با پارامتر λ

در روش JM یک اینترپولیشن خطی بین MLE[3] و CLM[4] انجام می‌دهیم که به وسیله **پارامتر لاندا** کنترل می‌شود. $\lambda \in [0, 1]$ لذا در این روش هرچه مقدار لاندا بزرگتر تنظیم شود نگاه به مجموعه مدل زبان یا **Background** پررنگ‌تر و هرچه که مقدار لاندا کوچکتر باشد اهمیت **سند فعلی** بیشتر می‌شود. پس با اختلاط دو توزیع با یکدیگر، به هدف اختصاص دادن احتمال غیرصفر به کلمات نادیده در سندی که در حال حاضر امتیاز می‌دهیم، می‌رسیم.

در زیر و در فرمول روش JM می‌توان جزئیات نحوه اینترپولیشن خطی را دید:

$$p(w|d) = (1-\lambda)\frac{c(w,d)}{|d|} + \lambda\, p(w|C)$$

حال با روش JM بازیابی را ابتدا با مقدار پیش‌فرض λ روی مجموعه پرس‌وجو ۵۱ الی ۱۰۰ با مقدار ۱۰۰۰ برای تعداد Requested، انجام داده و مقادیر MAP و P@20 را بدست می‌آوریم. در ادامه λ را به صورت آزمون و خطا ابتدا با گام‌های بلند تغییر داده و هر نوبت مقادیر را یاداشت می‌کنیم و در صورتی که شاهد بهبود نسبت به حالت پیش‌فرض بودیم، مقادیر نزدیک را با گام‌های کوچک‌تری امتحان می‌کنیم تا به مقدار بهینه برسیم.

[1] if a user likes document *d*, how likely would the user enter query *q* in order to retrieve document *d*?
[2] Linear interpolation, Jelinek-Mercer
[3] Maximum Likelihood Estimate
[4] Collection Language Model

در جدول زیر نتایج بدست آمده قابل مشاهده است:

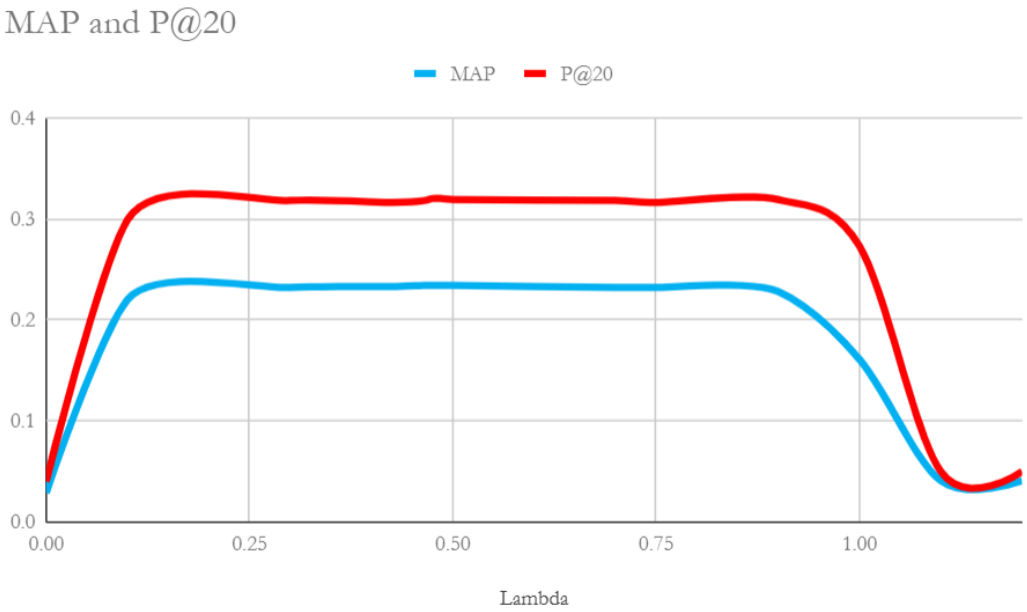
روش JM	۱۰۰ - ۵۱		
	λ	MAP	P@20
	۰	۰.۰۲۹	۰.۰۴
	۰.۱	۰.۲۲	۰.۲۹۹
	۰.۳	۰.۲۳۲	۰.۳۱۸
	۰.۴۲۵	۰.۲۳۳	۰.۳۱۶
	۰.۴۶۵	۰.۲۳۴	۰.۳۱۸
	۰.۴۷۵	۰.۲۳۴	۰.۳۲
	۰.۴۸۵	۰.۲۳۴	۰.۳۲
	۰.۵	۰.۲۳۴	۰.۳۱۹
	۰.۷	۰.۲۳۲	۰.۳۱۸
	۰.۷۵	۰.۲۳۲	۰.۳۱۶
	۰.۹	۰.۲۲۸	۰.۳۱۹
	۱	۰.۱۶۱	۰.۲۷۳
	۱.۱	۰.۰۴۱	۰.۰۵
	۱.۲	۰.۰۴۱	۰.۰۵

بازه مقادیر مناسب برای پارامتر λ بین صفر و یک بوده و مقدار پیش‌فرض آن ۰.۵ می‌باشد. در این بخش ما بر این اساس به تست کردن مقادیر در این نواحی و نیز حاشیه‌ای از بالا و پایین این نواحی ابتدا با گام‌های بلند و سپس با گام‌های کوتاه پرداختیم و مقادیر بهینه برای پرسوجوهای ۵۱ الی ۱۰۰ برای پارامتر λ ۰.۴۷۵ و ۰.۴۸۵ دیده شد.

نتایج حاصل از این مقادیر بهینه (رنگ سبز) به نسبت نتایج بدست آمده از مقدار پیش‌فرض (رنگ آبی) مقدار بیشتر داشت که نشان‌دهنده این است که بهینه‌سازی این پارامترها در کسب نتایج بهتر موفق بوده است.

علیرغم این که طبق تئوری مقدار بالای یک برای پارامتر λ بی‌معنا است ما آن‌ها را هم تست کردیم که نتایج عملی بدست آمده هم در تائید تئوری بودند.

در شکل زیر نمودار MAP و P@20 حاصل از تست مقادیر مختلف برای پارامتر λ قابل مشاهده است.



روش دوم: Dirichlet با پارامتر μ

در این روش نیز درست مانند هموارسازی Jelinek-Mercer، ما از مدل زبان مجموعه استفاده خواهیم کرد، اما در این مورد می‌خواهیم آن را با تخمین MLE به روشی متفاوت ترکیب کنیم. فرمول ابتدا می‌تواند به عنوان یک اینترپولیشن از احتمال MLE و مدل زبان مجموعه مانند قبل دیده شود. در عوض، α_d صرفاً یک λ ثابت نیست، بلکه یک ضریب پویا است که $\mu > 0$ را به عنوان یک پارامتر می گیرد. اگر μ را روی یک ثابت قرار دهیم، نتیجه این است که یک سند طولانی در واقع ضریب کوچکتری در اینجا به دست می‌آورد. بنابراین، یک سند طولانی همانطور که ما انتظار داریم، هموارسازی کمتری خواهد داشت، بنابراین به نظر می رسد این امر منطقی تر از هموارسازی با ضریب ثابت باشد. در این روش مجموع ضرایب همچنان جمع برابر با یک دارند که مورد انتظار است. در زیر فرمول روش دریکله آورده شده است:

$$p(w|d) = \frac{c(w,d) + \mu p(w|C)}{|d| + \mu} = \frac{|d|}{|d| + \mu} \frac{c(w,d)}{|d|} + \frac{\mu}{|d| + \mu} p(w|C) \qquad \mu \in [0, +\infty)$$

حال با روش Dirichlet بازیابی را ابتدا با مقدار پیش‌فرض μ روی مجموعه پرس‌وجو ۵۱ الی ۱۰۰ با مقدار ۱۰۰۰ برای تعداد Requested، انجام داده و مقادیر MAP و P@20 را بدست می‌آوریم. در ادامه μ را به صورت آزمون و خطا ما بین بازه ۰ الی ۳۰۰۰۰ ابتدا با گام‌های بلند تغییر داده و هر نوبت مقادیر را یاداشت می‌کنیم و در صورتی که شاهد بهبود نسبت به حالت پیش‌فرض بودیم، مقادیر نزدیک را با گام‌های کوچک‌تری امتحان می‌کنیم تا به مقدار بهینه برسیم.

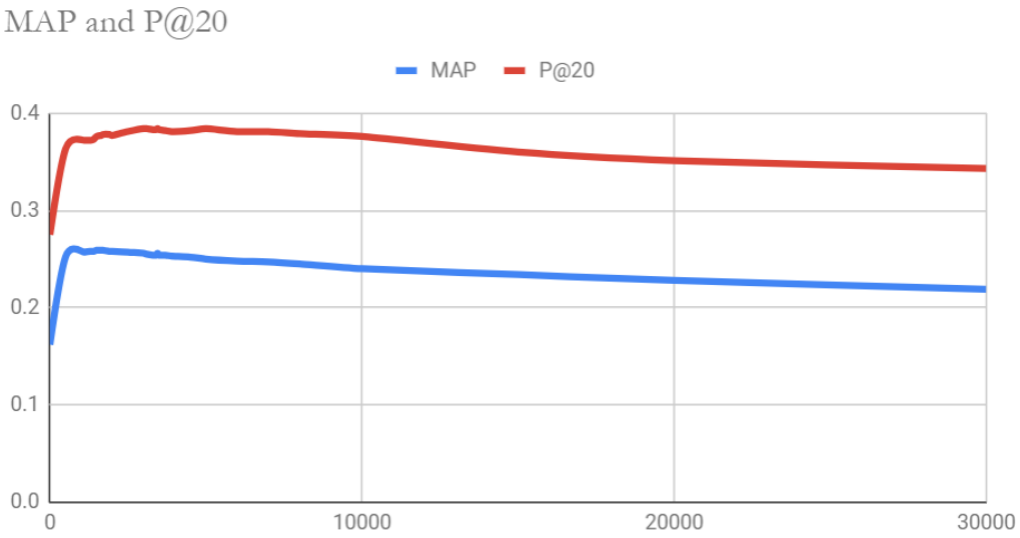
در جدول زیر نتایج بدست آمده قابل مشاهده است:

	۵۱ – ۱۰۰		
	μ	MAP	P@20
روش Dirichlet	۰	۰.۱۶۲	۰.۲۷۵
	۵۰۰	۰.۲۵۲	۰.۳۶۳
	۱۱۰۰	۰.۲۵۷	۰.۳۷۲
	۱۳۰۰	۰.۲۵۸	۰.۳۷۲
	۱۴۰۰	۰.۲۵۸	۰.۳۷۳
	۱۵۰۰	۰.۲۵۹	۰.۳۷۶
	۱۶۰۰	۰.۲۵۹	۰.۳۷۷
	۱۶۵۰	۰.۲۵۹	۰.۳۷۷
	۱۷۰۰	۰.۲۵۹	۰.۳۷۸
	۱۹۰۰	۰.۲۵۸	۰.۳۷۸
	۲۰۰۰	۰.۲۵۸	۰.۳۷۷
	۲۵۰۰	۰.۲۵۷	۰.۳۸۱
	۳۰۰۰	۰.۲۵۶	۰.۳۸۴
	۳۱۰۰	۰.۲۵۵	۰.۳۸۴
	۳۳۰۰	۰.۲۵۴	۰.۳۸۳
	۳۴۰۰	۰.۲۵۴	۰.۳۸۳
	۳۴۵۰	۰.۲۵۶	۰.۳۸۴
	۳۵۰۰	۰.۲۵۴	۰.۳۸۳
	۳۷۰۰	۰.۲۵۴	۰.۳۸۲
	۳۹۰۰	۰.۲۵۳	۰.۳۸۱
	۴۵۰۰	۰.۲۵۲	۰.۳۸۲
	۵۰۰۰	۰.۲۵	۰.۳۸۴
	۶۰۰۰	۰.۲۴۸	۰.۳۸۱
	۷۰۰۰	۰.۲۴۷	۰.۳۸۱
	۸۰۰۰	۰.۲۴۵	۰.۳۷۹
	۱۰۰۰۰	۰.۲۴	۰.۳۷۶
	۱۵۰۰۰	۰.۲۳۴	۰.۳۶
	۲۰۰۰۰	۰.۲۲۸	۰.۳۵۱

	۳۰۰۰۰	۰.۲۱۹	۰.۳۴۳

مقدار پیش‌فرض پارامتر μ، ۱۵۰۰ می‌باشد. در این بخش ما بر این اساس به تست کردن مقادیر در این نواحی و نیز حاشیه‌ای از بالا و پایین این نواحی ابتدا با گام‌های بلند و سپس با گام‌های کوتاه پرداختیم و مقدار بهینه برای پرسوجوهای ۵۱ الی ۱۰۰ برای پارامتر μ ، ۳۴۵۰ دیده شد. البته در این مقدار اندازه معیار ارزیابی MAP نسبت به حالت پیش‌فرض کاهش کوچکی داشت اما معیار ارزیابی P@20 بهبود قابل ملاحظه‌ای پیدا کرد.

در شکل زیر نمودار MAP و P@20 حاصل از تست مقادیر مختلف برای پارامتر μ را مشاهده می‌کنید.



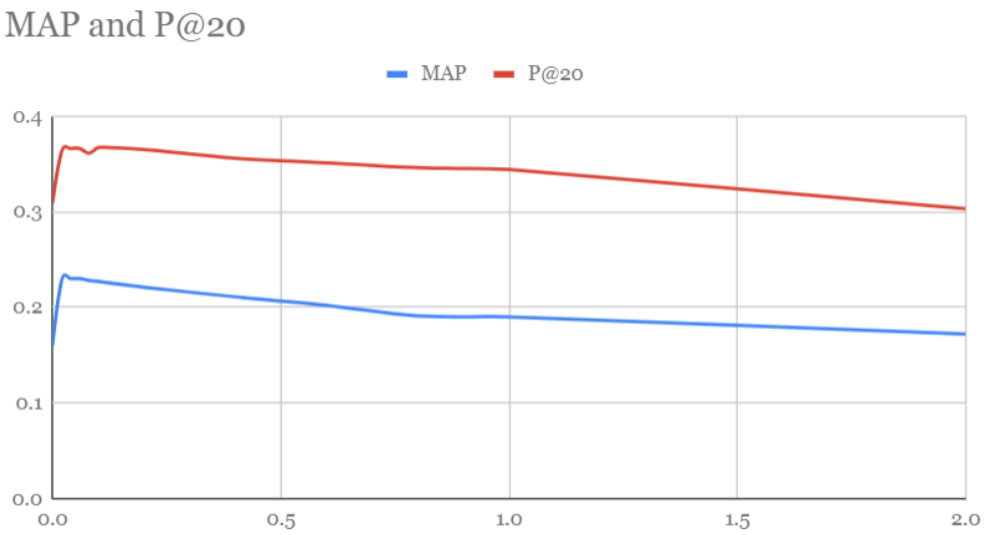
روش سوم: Additive Smoothing

در این قسمت با روش Additive Smoothing بازیابی را ابتدا با مقدار پیش‌فرض تتا روی مجموعه پرس‌وجو ۵۱ الی ۱۰۰ با مقدار ۱۰۰۰ برای تعداد Requested، انجام داده و مقادیر MAP و P@20 را بدست می‌آوریم. در ادامه تتا را به صورت آزمون و خطا ما بین بازه ۰ الی ۱۰ ابتدا با گام‌های بلند تغییر داده و هر نوبت مقادیر را یاداشت می‌کنیم و در صورتی که شاهد بهبود نسبت به حالت پیش‌فرض بودیم، مقادیر نزدیک را با گام‌های کوچک‌تری امتحان می‌کنیم تا به مقدار بهینه برسیم.

در جدول زیر نتایج بدست آمده قابل مشاهده است:

روش Additive Smoothing	θ	MAP	P@20
	۰	۰.۱۶	۰.۳۰۹
	۰.۰۲	۰.۲۲۸	۰.۳۶۳
	۰.۰۴	۰.۲۳	۰.۳۶۶
	۰.۰۶	۰.۲۳	۰.۳۶۶
	۰.۰۸	۰.۲۲۸	۰.۳۶۱
	۰.۱	۰.۲۲۷	۰.۳۶۷
	۰.۲	۰.۲۲۱	۰.۳۶۵
	۰.۴	۰.۲۱۱	۰.۳۵۶
	۰.۶	۰.۲۰۲	۰.۳۵۱
	۰.۸	۰.۱۹۱	۰.۳۴۶
	۱	۰.۱۹	۰.۳۴۴
	۲	۰.۱۷۲	۰.۳۰۳
	۴	۰.۱۵۳	۰.۲۷۲
	۱۰	۰.۱۳۱	۰.۲۲۵

در شکل زیر نمودار MAP و P@20 حاصل از تست مقادیر مختلف برای پارامتر تتا را مشاهده می‌کنید.



سوال ۲: هموارسازی دو مرحله‌ای

در این قسمت به ترکیب دو روش از روش‌های استفاده شده در بخش قبل پرداخته و هموارسازی دو مرحله‌ای را شکل می‌دهیم و به بررسی نتایج حاصله از آن و ارزیابی نتایج با معیارهای کارایی MAP و P@20 بدست آورده و به تحلیل آن‌ها می‌پردازیم.

حال با روش هموارسازی دومرحله‌ای بازیابی را با مقادیر μ و λ روی مجموعه پرس‌وجو ۵۱ الی ۱۰۰ با مقدار ۱۰۰۰ برای تعداد Requested، انجام داده و مقادیر MAP و P@20 را بدست می‌آوریم. در ادامه μ و λ را به صورت آزمون و خطا به ترتیب ما بین بازه‌های ۱۰۰۰ الی ۴۰۰۰ و ۰ الی ۰.۵ ابتدا با گام‌های بلند تغییر داده و هر نوبت مقادیر را یاداشت می‌کنیم و در صورتی که شاهد بهبود نسبت به حالت پیش‌فرض بودیم، مقادیر نزدیک را با گام‌های کوچک‌تری امتحان می‌کنیم تا به مقدار بهینه برسیم.

در جدول زیر نتایج بدست آمده معیار ارزیابی MAP برای روش هموارسازی دومرحله‌ای قابل مشاهده است:

روش هموارسازی دومرحله‌ای	μ	λ	MAP
	۱۰۰۰	۰	۰.۲۵۷
	۱۵۰۰		۰.۲۵۹
	۱۷۰۰		۰.۲۵۹
	۲۰۰۰		۰.۲۵۸
	۳۰۰۰		۰.۲۵۶
	۴۰۰۰		۰.۲۵۳
	۱۰۰۰	۰.۱	۰.۲۵۷
	۱۵۰۰		۰.۲۵۹
	۱۷۰۰		۰.۲۵۹
	۲۰۰۰		۰.۲۵۸
	۳۰۰۰		۰.۲۵۴
	۴۰۰۰		۰.۲۵۱
	۱۰۰۰	۰.۲	۰.۲۵۸
	۱۵۰۰		۰.۲۵۸
	۱۷۰۰		۰.۲۵۸
	۲۰۰۰		۰.۲۵۸
	۳۰۰۰		۰.۲۵۳

	۴۰۰۰		۰.۲۵
	۱۰۰۰	۰.۳	۰.۲۵۸
	۱۵۰۰		۰.۲۵۷
	۱۷۰۰		۰.۲۵۸
	۲۰۰۰		۰.۲۵۶
	۳۰۰۰		۰.۲۵۱
	۴۰۰۰		۰.۲۴۸
	۱۰۰۰	۰.۴	۰.۲۵۸
	۱۵۰۰		۰.۲۵۷
	۱۷۰۰		۰.۲۵۶
	۲۰۰۰		۰.۲۵۴
	۳۰۰۰		۰.۲۴۹
	۴۰۰۰		۰.۲۴۷
	۱۰۰۰	۰.۵	۰.۲۵۷
	۱۵۰۰		۰.۲۵۵
	۱۷۰۰		۰.۲۵۴
	۲۰۰۰		۰.۲۵۲
	۳۰۰۰		۰.۲۴۸
	۴۰۰۰		۰.۲۴۴

در جدول زیر نتایج بدست آمده معیار ارزیابی P@20 برای روش هموارسازی دومرحله‌ای قابل مشاهده است:

روش هموارسازی دو مرحله‌ای	μ	λ	P@20
	۱۰۰۰	۰	۰.۳۷۳
	۱۵۰۰		۰.۳۷۶
	۱۷۰۰		۰.۳۷۸
	۲۰۰۰		۰.۳۷۷
	۳۰۰۰		۰.۳۸۴
	۴۰۰۰		۰.۳۸۱
	۱۰۰۰	۰.۱	۰.۳۷۱
	۱۵۰۰		۰.۳۷۵
	۱۷۰۰		۰.۳۷۷
	۲۰۰۰		۰.۳۷۸

۳۰۰۰		۰.۳۸۱
۴۰۰۰		۰.۳۸۱
۱۰۰۰	۰.۲	۰.۳۷۱
۱۵۰۰		۰.۳۷۷
۱۷۰۰		۰.۳۷۷
۲۰۰۰		۰.۳۸
۳۰۰۰		۰.۳۸۱
۴۰۰۰		۰.۳۷۹
۱۰۰۰	۰.۳	۰.۳۷۴
۱۵۰۰		۰.۳۷۵
۱۷۰۰		۰.۳۷۶
۲۰۰۰		۰.۳۷۹
۳۰۰۰		۰.۳۷۹
۴۰۰۰		۰.۳۸
۱۰۰۰	۰.۴	۰.۳۷۱
۱۵۰۰		۰.۳۷۳
۱۷۰۰		۰.۳۷۷
۲۰۰۰		۰.۳۸
۳۰۰۰		۰.۳۸
۴۰۰۰		۰.۳۸
۱۰۰۰	۰.۵	۰.۳۶۹
۱۵۰۰		۰.۳۷۵
۱۷۰۰		۰.۳۷۷
۲۰۰۰		۰.۳۷۹
۳۰۰۰		۰.۳۸۱
۴۰۰۰		۰.۳۷۴

بر اساس نتایج ارائه شده در جدول‌های بالا، بهترین مقدار برای شاخص MAP در $\mu = 1700$ و $\lambda = 0$ و $\mu = 1700$ و $\lambda = 0.1$ و بهترین مقدار برای شاخص P@20 در $\mu = 3000$ و $\lambda = 0$ حاصل می‌شود. مقداری که برای هر دو معیار حالت مناسبی داشته باشد، $\mu = 3000$ و $\lambda = 0$ است.

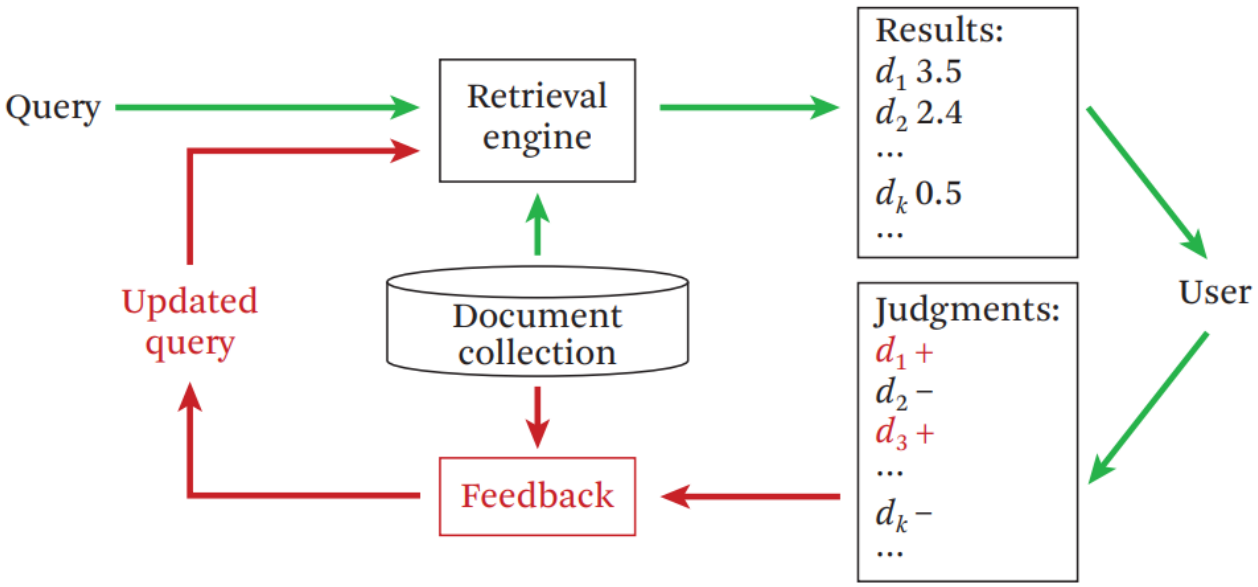
سوال ۳: پیاده‌سازی تابع وزن‌دهی با استفاده از Pseudo Relevance Feedback

بسیاری از کاربران می گویند یک سند خوب است یا یک سند خیلی مفید نیست. هر تصمیم در مورد یک سند، قضاوت مرتبط نامیده می شود. این فرآیند نوعی بازخورد مرتبط است، زیرا بر اساس قضاوت‌های نتایج جستجو، اطلاعات بازخوردی از کاربر دریافت کرده‌ایم.

همانطور که انتظار می‌رود این روش می‌تواند برای سیستم بازیابی بسیار مفید باشد زیرا ما باید بتوانیم یاد بگیریم که دقیقاً چه چیزی برای یک کاربر یا کاربران خاص جالب است. سپس ماژول بازخورد این قضاوت‌ها را به عنوان ورودی می‌گیرد و همچنین از مجموعه اسناد برای بهبود رتبه بندی‌های آینده استفاده می‌کند.

این نوع قضاوت‌های مرتبط قابل اعتماد هستند، اما کاربران معمولاً نمی‌خواهند تلاش بیشتری انجام دهند، مگر اینکه مجبور باشند. شکل دیگری از بازخورد وجود دارد به نام بازخورد شبه مرتبط[5] یا بازخورد کور. در این مورد، ما مجبور نیستیم کاربران را درگیر کنیم، زیرا به سادگی فرض می‌کنیم که اسناد رتبه‌بندی بالا مرتبط هستند.

فرض کنید $k = 10$ سند بالا مرتبط هستند. سپس از این اسناد برای یادگیری و بهبود پرس و جو استفاده خواهیم کرد. اما اگر اسناد رتبه‌بندی شده تصادفی باشند، چگونه می‌تواند کمک کند؟ در واقع، اسناد برتر در واقع مشابه اسناد مربوطه هستند، حتی اگر مرتبط نباشند. در غیر این صورت، چگونه آنها در لیست رتبه بندی بالا ظاهر می شدند؟ بنابراین، به هر حال می توان برخی از اصطلاحات مرتبط با پرس و جو را از این مجموعه یاد گرفت، صرف نظر از اینکه کاربر بگوید یک سند مرتبط است یا نه.



[5] Pseudo Relevance Feedback

متأسفانه، بازخورد شبه مرتبط کاملاً قابل اعتماد نیست. ما باید خودسرانه یک برش تعیین کنیم و امیدوار باشیم که عملکرد رتبه بندی به اندازه کافی خوب باشد تا حداقل برخی از اسناد مفید را بدست آوریم. همچنین روش بازخورد دیگری به نام بازخورد ضمنی وجود دارد. در این مورد، ما همچنان کاربران را درگیر می‌کنیم، اما لازم نیست صریحاً از آنها بخواهیم قضاوت کنند. در عوض، ما قصد داریم نحوه تعامل کاربران با نتایج جستجو را با مشاهده تعداد کلیک آنها مشاهده کنیم. اگر کاربر روی یک سند کلیک کرد و سند دیگری را رد کرد، این سرنخی درباره مفید بودن یا نبودن یک سند می دهد. حتی می‌توانیم فرض کنیم که در اینجا فقط از قطعه در سندی که در صفحه نتایج موتور جستجو نمایش داده می‌شود (متنی که در واقع توسط کاربر دیده می‌شود) استفاده می‌کنیم. می توانیم فرض کنیم که این متن نمایش داده شده احتمالاً برای کاربر مرتبط یا جالب است زیرا آنها روی آن کلیک کرده اند. این ایده پشت بازخورد ضمنی است و ما می‌توانیم دوباره از این اطلاعات برای به‌روزرسانی پرس و جو استفاده کنیم. این یک تکنیک بسیار مهم است که در موتورهای جستجوی مدرن مورد استفاده قرار می‌گیرد.