ARYA VARASTE

۱۴۰۰/۰۹/۰۱

بازیابی هوشمند اطلاعات

تمرین اول

a.varaste.n@gmail.com

در ابتدا محیط لازم برای انجام تمرین را به این ترتیب فراهم کردیم:

- دانلود و نصب نسخه ۲۰.۴ از سیستم‌عامل اوبونتو
- نصب جاوا
- نصب Maven
- نصب Galago

پیش‌نیاز: ایجاد شاخص

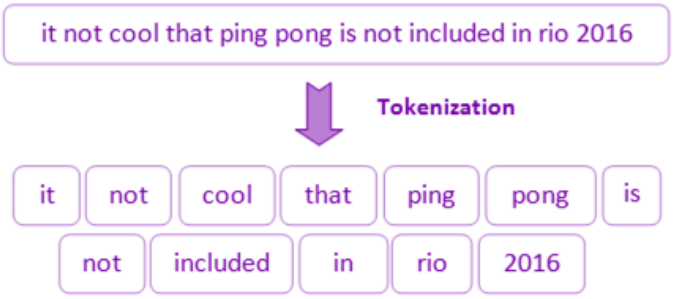
در هر زبانی، کلمات با توجه به نقشی که در جملات ایفا می کنند، به شکل‌های ظاهری متفاوتی خواهند بود. اما تمامی آن‌ها از یک ریشه ساخته می شوند. لذا در بسیاری از روش‌ها، ابتدا می‌بایست ریشه کلمات را پیدا کنیم. یکی از روش‌های متداول برای ریشه‌یابی کلمات، روش ریشه یابی Stemming است. الگوریتم‌های مختلفی جهت 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: تابع بازیابی BM25

الف) در این بخش با روش BM25 بازیابی را ابتدا با مقادیر پیش‌فرض b و k روی مجموعه پرس‌وجو ۱۰۱ الی ۱۵۰ با مقدار ۱۰۰ برای تعداد Requested، انجام داده و مقادیر Recall، MAP، nDCG و P@5 را بدست می‌آوریم. در ادامه مقادیر b و k 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.

در جدول زیر نتایج بدست آمده قابل مشاهده است:

	b	k	۱۰۱-۱۵۰				
			Recall			MAP	nDCG	P@5
BM25 روش ۱-	۰.۷۵	۱.۲	۱۰۹۹	۴۸۰۵	۰.۲۲۹	۰.۱۶۸	۰.۳۳۱	۰.۳۶۸
	۰.۳۰	۱.۲	۱۰۶۱	۴۸۰۵	۰.۲۲۱	۰.۱۵۹	۰.۳۱۸	۰.۳۶۸
	۰.۷۰	۱.۲	۱۱۰۰	۴۸۰۵	۰.۲۲۹	۰.۱۶۵	۰.۳۳۲	۰.۳۷۶
	۰.۵۵	۱.۲	۱۱۲۴	۴۸۰۵	۰.۲۳۴	۰.۱۶۹	۰.۳۳۵	۰.۳۸۰
	۰.۵۵	۱.۲	۱۱۲۴	۴۸۰۵	۰.۲۳۴	۰.۱۶۹	۰.۳۳۵	۰.۳۸۰
	۰.۵۰	۱.۵	۱۱۵۱	۴۸۰۵	۰.۲۴۰	۰.۱۷۴	۰.۳۴۱	۰.۴۱۲
	۰.۵۰	۱.۷	۱۱۵۶	۴۸۰۵	۰.۲۴۱	۰.۱۷۶	۰.۳۴۵	۰.۴۰۸
	۰.۶۰	۱.۷	۱۱۴۵	۴۸۰۵	۰.۲۳۸	۰.۱۷۳	۰.۳۴۴	۰.۳۹۲
	۰.۵۰	۱.۸	۱۱۵۳	۴۸۰۵	۰.۲۴۰	۰.۱۷۷	۰.۳۴۵	۰.۴۰۸
	۰.۵۰	۲.۰	۱۱۵۴	۴۸۰۵	۰.۲۴۰	۰.۱۷۸	۰.۳۴۷	۰.۴۱۶
	۰.۶۰	۲.۵	۱۱۳۵	۴۸۰۵	۰.۲۳۶	۰.۱۷۷	۰.۳۴۶	۰.۴۰۰
	۰.۴۰	۲.۶	۱۱۶۸	۴۸۰۵	۰.۲۴۳	۰.۱۸۲	۰.۳۴۸	۰.۴۲۰
	۰.۷۰	۲.۵	۱۱۰۷	۴۸۰۵	۰.۲۳۰	۰.۱۷۳	۰.۳۴۲	۰.۳۹۲
	۰.۳۰	۱.۹	۱۱۶۹	۴۸۰۵	۰.۲۴۳	۰.۱۷۹	۰.۳۴۴	۰.۳۹۲
	۱.۰۰	۲.۰	۹۴۵	۴۸۰۵	۰.۱۹۷	۰.۱۴۳	۰.۲۲۹	۰.۳۴۰

با جست‌وجو در منابع علمی محدوده مناسب برای پارامتر b، ۰.۳ الی ۱ و محدوده مناسب برای پارامتر k، ۰.۵ الی ۲.۵ به نظر آمد و بر این اساس به تست کردن مقادیر در این نواحی و نیز حاشیه‌ای از بالا و پایین این نواحی پرداختیم و مقایر بهینه برای پرسوجوهای ۱۰۱ الی ۱۵۰ برای پارامترهای b و k به ترتیب ۰.۴ و ۲.۶ دیده شد.

نتایج حاصل از این مقادیر به نسبت نتایج بدست آمده از مقادیر پیش‌فرض (ردیف اول جدول) مقدار بیشتر داشت که نشان‌دهنده این است که بهینه‌سازی این پارامترها در کسب نتایج بهتر موفق بوده است.

ب) در این بخش مقادیری که برای پارامترهای b و k روی پرس‌وجو های ۱۰۱ الی ۱۵۰ بهینه بدست آوردیم را روی پرس‌وجوهای ۵۱ الی ۱۰۰ اعمال کرده و با مقادیر پیش‌فرض مورد مقایسه قرار می‌دهیم.

	b	k	۵۱-۱۰۰				
			Recall		MAP	nDCG	P@5
۱-روش BM25	۰.۷۵	۱.۲	۲۳۳۹	۱۰۶۲۰	۰.۲۲۰	۰.۱۷۰	۰.۳۲۲	۰.۳۸۲
	۰.۴۰	۲.۶	۲۴۱۸	۱۰۶۲۰	۰.۲۲۸	۰.۱۷۹	۰.۳۳۱	۰.۴۱۲

همان‌طور که ملاحظه می‌شود با استفاده از مقادیر بهینه برای پارمترهایی که در قسمت قبل برای پرس‌وجوهای متفاوتی بدست آوردیم، روی این مجموعه از پرس‌وجو هم نتیجه بهتری از مقادیر پیش‌فرض رسیده ایم. این مسئله نشان دهنده اثر مجموعه اسناد روی بهبود مقدارهای پارامترها می‌باشد.

در ادامه از مقادیری که به عنوان مقادیر بهینه برای پارامترها محاسبه کردیم و نیز مقدار پیش‌فرض و دومین بهترین مقدار بهینه‌ای که در بخش قبلی استخراج کرده‌ایم استفاده کرده و روش‌های پیشنهاد شده در تمرین را پس از پیاده‌سازی برای پرس‌وجوهای ۵۱ الی ۱۰۰ مورد بازیابی و ارزیابی قرار می‌دهیم. نتایج حاصل شده در جدول زیر ارائه شده.

	b	k	۵۱-۱۰۰				
			Recall		MAP	nDCG	P@5
۱-روش BM25	۰.۷۵	۱.۲	۲۳۳۹	۱۰۶۲۰	۰.۲۲۰	۰.۱۷۰	۰.۳۲۲	۰.۳۸۲
	۰.۴۰	۲.۶	۲۴۱۸	۱۰۶۲۰	۰.۲۲۸	۰.۱۷۹	۰.۳۳۱	۰.۴۱۲
۲-روش اول	۰.۷۵	۱.۲	۱۷۸	۱۰۶۲۰	۰.۰۱۷	۰.۰۰۸	۰.۰۲۱	۰.۰۲۲
	۰.۴۰	۲.۶						
	۰.۳۰	۱.۹						
۳-روش دوم	۰.۷۵	۱.۲	۲۱۳۰	۱۰۶۲۰	۰.۲۰۱	۰.۱۴۷	۰.۲۸۷	۰.۴۰۲

	۰.۴۰	۲.۶	۲۰۵۹	۱۰۶۲۰	۰.۱۹۴	۰.۱۴۱	۰.۲۷۹	۰.۳۹۲
	۰.۳۰	۱.۹	۲۱۱۴	۱۰۶۲۰	۰.۱۹۹	۰.۱۴۵	۰.۲۸۴	۰.۴۰۴
۴-روش سوم	۰.۷۵	۱.۲	۱۳۳۶	۱۰۶۲۰	۰.۱۲۶	۰.۰۷۳	۰.۱۸۱	۰.۲۲۲
	۰.۴۰	۲.۶						
	۰.۳۰	۱.۹						
۵-روش چهارم	۰.۷۵	۱.۲	۲۳۴۱	۱۰۶۲۰	۰.۲۲۰	۰.۱۷۱	۰.۳۲۴	۰.۳۹۶
	۰.۴۰	۲.۶	۲۴۰۵	۱۰۶۲۰	۰.۲۲۶	۰.۱۷۸	۰.۳۲۹	۰.۴۱۶
	۰.۳۰	۱.۹	۲۴۳۳	۱۰۶۲۰	۰.۲۲۹	۰.۱۷۹	۰.۳۳۰	۰.۴۱۲
۶-روش پنجم با مقدار تتای ۱	۰.۷۵	۱.۲	۲۳۳۹	۱۰۶۲۰	۰.۲۲۰	۰.۱۷۰	۰.۳۲۲	۰.۳۸۲
	۰.۴۰	۲.۶	۲۴۱۹	۱۰۶۲۰	۰.۲۲۸	۰.۱۷۹	۰.۳۳۱	۰.۴۱۲
	۰.۳۰	۱.۹	۲۴۴۳	۱۰۶۲۰	۰.۲۳۰	۰.۱۸۰	۰.۳۳۲	۰.۴۰۶

فایل‌های پیاده‌سازی‌های مربوط به روش‌های پیشنهادی اول تا پنجم در فایل‌های av2 الی av6 در کنار گزارش قرار دارند که به دلیل این که در صورت تمرین اشاره شده که به توضیح آن‌ها نپردازیم از نوشتن در خصوصشان صرف نظر کردیم.

در تفسیر نتایج بدست آمده که در جدول بالا ارائه شده‌اند باید اشاره کرد که روش پیشنهادی اول عملکرد بسیار ضعیفی از خود نشان داده و چون در آن از پارامترهای b و k استفاده نشده است، لذا بهینه‌سازی آن‌ها اثر در بهبود یافتن عملکرد بازیابی ندارد. همچین نتایج بدست آمده از این روش عملکرد بسیار ضعیفی نسبت به سایر روش‌های پیشنهادی و نیز روش اصلی دارد.

روش‌پیشنهادی دوم پس از لحاظ کردن پارامترهای بهینه به مقدار کمتری برای معیار MAP نسبت به استفاده از مقادیر پیش‌فرض دست پیدا کرد اما با این‌حال از عملکرد قابل قبولی از خود نشان داد.

روش‌پیشنهادی سوم علی‌رغم سادگی بسیار عملکردی به مراتب مناسب‌تر از روش اول بر روی داده‌ها و پرس‌وجوهای ارائه شده از خود به نمایش گذاشت هرچند که در کل نتایج جالبی نسبت به روش‌های بعدی نداشت. در این روش هم به دلیل این که در آن از پارامترهای b و k استفاده نشده است، عملیات بهینه‌سازی‌ای که در بخش‌های قبل روی پارامترها داشتیم برای بهبود دادن عملکرد بی‌اثر بود.

در روش‌پیشنهادی چهارم به نتایج بهتری نسبت به روش‌های پیشنهادی قبلی و نیز روش اصلی دست پیدا کردیم. در این روش هم شاهد بهبود قابل ملاحظه معیارهای ارزیابی برای استفاده از پارامترهای بهینه یافته شده در قسمت‌های قبل نسبت به مقادیر پیش‌فرض این پارامترها هستیم.

در روش‌پیشنهادی پنجم از ما خواسته شده که فورمول ارائه شده را پیاده‌سازی کرده و برای مقادیر مختلف از تتا مورد بررسی قرار دهیم و در نهایت نتایج بدست آمده با تتای برتر را گزارش نمائیم. با بررسی‌های صورت گرفته بهترین مقدار برای تتا در این روش ۱ اندازه‌گیری شده که مقادیر معیارهای ارزیابی مرتبط به آن در جدول قرار دارند. در این روش با بهینه‌سازی مقدار پارامترها موفق شدیم که به نتایج بهتری برای همه معیارهای ارزیابی نسبت به حالت اولیه پارامترها دست پیدا کنیم.

در انتها این بخش جهت سهولت دسترسی بخش‌هایی از قسمت‌های مهم‌تر از کد ضمیمه شده است.

```
  private double score(double count, double length) {
    return idf;
  }
```

```
  private double score(double count, double length) {
    double numerator = count * (k + 1);
    double denominator = count + k;
    return numerator / denominator;
  }
```

```
  private double score(double count, double length) {
    double i = 0;
    if(count != 0){
      i = 1;
    }
    return i;
  }
```

```
  private double score(double count, double length) {
    double numerator = (k + 1) * (count / (1 - b + (b * length /
avgDocLength)) + 0.5);
    double denominator = k + (count / (1 - b + (b * length /
avgDocLength)) + 0.5);
    return idf * numerator / denominator;
  }
```

```
  private double score(double count, double length) {
    double numerator = count * (k + 1);
```

```
    double denominator = count + (k * (1 - b + (b * length /
avgDocLength)));
    return idf * (numerator / denominator + 1);
  }
```

```
    // count -> score iterators
    // Scorers can be named directly as nodes
    {av2.class.getName(), "av2"},
    {av3.class.getName(), "av3"},
    {av4.class.getName(), "av4"},
    {av5.class.getName(), "av5"},
    {av6.class.getName(), "av6"},
```

سوال ۲: تابع بازیابی Pivoted Length Normalization

در جدول زیر مقایسه روش‌های خواسته شده در صورت تمرین برای معیارهای کارایی روی پرس‌جوهای ۵۱-۱۰۰ قابل مشاهده است. با توجه به جدول زیر روش‌های BM25 و BM25+ از نظر معیار کارایی MAP نتایج بهتری برای مقادیر پیش‌فرض پارامترهای b و k دارند.

	۵۱-۱۰۰					
	Recall			MAP	nDCG	P@5
مدل اصلی	۲۱۸۸	۱۰۶۲۰	۰.۲۰۶	۰.۱۲۱	۰.۲۸۵	۰.۲۰۳
BM25	۲۳۳۹	۱۰۶۲۰	۰.۲۲۰	۰.۱۷۰	۰.۳۲۲	۰.۳۸۲
BM25+	۲۳۳۹	۱۰۶۲۰	۰.۲۲۰	۰.۱۷۰	۰.۳۲۲	۰.۳۸۲
مدل بدون موئلفه لگاریتمی تو در تو	۲۰۶۸	۱۰۶۲۰	۰.۱۹۴	۰.۱۱۹	۰.۲۶۱	۰.۲۰۱

در انتها جهت سهولت دسترسی بخش‌هایی از قسمت‌های مهم‌تر از کد ضمیمه شده است.

```
  private double score(double count, double length) {
    double numerator = log(1 + log(1 + count));
    double denominator=1 - b + (b * length / avgDocLength);
    logg = log((documentCount +1 ) / (df));
```

```
    return count * (numerator / denominator) * (logg);
  }
```

```
  private double score(double count, double length) {
    double numerator=1 + log(1 + count);
    double denominator = 1 - b + (b * length / avgDocLength);
    logg = log(( documentCount + 1) / (df));
    return count*(numerator/denominator)*(logg);
  }
```

```
    {avb1.class.getName(), "avb1"},
    {avb2.class.getName(), "avb2"},
```