به نام خدا

دانشگاه تهران

دانشکدگان فنی

دانشکده مهندسی برق و کامپیوتر

درس بازیابی هوشمند اطلاعات

پاسخ بخش تئوری تمرین ۱

نام و نام خانودگی: روژین پناو

شماره دانشجویی: ۲۲۰۷۰۱۰۴۶

مهر ماه ۱۴۰۳

فـهـرست

پـاسخ سـوال اول

پاسخ بخش اول

b. BM25 :

$$\sum_{w\in q\cap d} \underbrace{\ln\left(\frac{N-df(w)+0.5}{df(w)+0.5}\right)}_{\textit{IDF weighting}} \cdot \underbrace{\frac{(k_1+1)\cdot c(w,d)}{k_1\cdot\left((1-b)+b\cdot\frac{|d|}{avgdl}\right)+c(w,d)}}_{\text{TF weighting + Length Normalization}} \cdot \underbrace{\frac{(k_3+1)\cdot c(w,q)}{k_3+c(w,q)}}_{\text{Query Term Weight}}$$

بخش TF weighting + Length Normalization به صورت زیر است:

$$\frac{(k_1+1)\cdot c(w,d)}{k_1\cdot\left((1-b)+b\cdot\frac{|d|}{avgdl}\right)+c(w,d)}$$

که داریم:

$$\text{Length Normalization} = (1-b)+b\cdot\frac{|d|}{avgdl}$$

$$\text{TF weighting} = \frac{(k_1+1)\cdot c(w,d)}{k_1\cdot(\text{Length Normalization})+c(w,d)}$$

این فرمول وزن ترم رو با در نظر گرفتن هم تعداد تکرار واژه و هم طول سند محاسبه می‌کنه. نرمالایزر طول سند به‌صورت جداگانه تعریف شده تا تأثیرش شفافتر باشه، و در مخرج TF قرار می‌گیره تا اثر طول سند رو کنترل کنه.

c. PL2 :

$$\sum_{c(w,q).\ln(1+\epsilon)\in q\cap d} \left[\left(\frac{tfn_w^d}{tfn_w^d+1}\right).log_2(tfn_w^d\cdot\lambda_w^d)+log_2 e\cdot\left(\frac{1}{\lambda_w}-tfn_w^d\right)+0.5.\log_2(2\pi\cdot tfn_w^d)\right]$$

فرمول‌های داخلی:

$$\boldsymbol{tfn}_w^d = c(w,d)\cdot\log_2\left(1+c\cdot\frac{avdl}{|d|}\right)$$

$$\boldsymbol{\lambda_d^w} = \frac{\boldsymbol{N}}{\boldsymbol{c(w,C)}}$$

از خاصیت لگاریتم استفاده می‌کنیم:

$$\log_2(tfn\cdot\lambda) = \log_2(tfn)+\log_2(\lambda)$$

پس:

$$\left(\frac{tfn}{tfn+1}\right)\cdot\log_2(tfn\cdot\lambda) = \frac{tfn}{tfn+1}\cdot[\log_2(tfn)+\log_2(\lambda)]$$

می‌تونیم بنویسیم:

$$\frac{1}{tfn+1}\cdot[tfn\cdot\log_2(tfn)+tfn\cdot\log_2(\lambda)]$$

که همون مؤلفه اول به‌صورت ضربی در کل عبارت هست.

عبارت:

$$\log_2 e\cdot(\frac{1}{\lambda}-tfn)$$

با تقریب استیرلینگ:

$$\frac{1}{\lambda}\approx\lambda+\frac{1}{12\cdot tfn}$$

جایگذاری:

$$\log_2 e\cdot(\lambda+\frac{1}{12\cdot tfn}-tfn)$$

کل عبارت را می‌نویسیم

$$\sum_{w\in q\cap d}\frac{1}{tfn+1}[tfn\cdot\log_2(tfn)+tfn\cdot\log_2(\lambda)]+(\lambda+\frac{1}{12\cdot tfn}-tfn)\cdot\log_2 e+\frac{1}{2}\cdot\log_2(2\pi\cdot tfn)$$

می‌تونیم بنویسیم:

$$\boxed{\sum_{w\in q\cap d}\frac{1}{\boldsymbol{tfn+1}}[\boldsymbol{tfn\cdot\log_2(\frac{tfn}{\lambda})}]+(\boldsymbol{\lambda+\frac{1}{12\cdot tfn}-tfn})\cdot\log_2 e+\frac{1}{2}\cdot\log_2(2\pi\cdot tfn)}$$

چون:

$$tfn\cdot\log_2(tfn)+tfn\cdot\log_2(\lambda)=tfn\cdot[\log_2(tfn)-\log_2(\frac{1}{\lambda})]=tfn\cdot\log_2(\frac{tfn}{\lambda})$$

TF Transformation

برای اینکه بفهمم کدوم بخش از فرمول نقش tf weighting رو داره، بررسی کردم کدوم مؤلفه مستقیماً به تعداد تکرار واژه در سند (tf) وابسته‌ست. چون tfn از $c(w,d)$ مشتق می‌شه و خودش نرمال‌شده‌ی tf هست، بخش زیر:

$$\frac{1}{tfn+1}\cdot tfn\cdot\log_2(\frac{tfn}{\lambda})$$

تنها بخشیه که هم به tf وابسته‌ست، هم به‌صورت ضربی ظاهر شده. این مؤلفه سه ویژگی داره که نشون می‌ده واقعاً tf weighting محسوب می‌شه:

- مستقیماً به تعداد تکرار واژه در سند وابسته‌ست

- نرمال‌سازی طول سند رو در خودش داره (از طریق تعریف tfn)
- با ضریب $\frac{1}{tfn+1}$ جلوی رشد بیش‌ازحد tf رو می‌گیره (اشباع می‌کنه)

در مقابل، دو مؤلفه‌ی دیگه بیشتر نقش اصلاح آماری و نرمال‌سازی دارند و مستقیماً به tf وابسته نیستند.

پس نتیجه می‌گیرم که:

$$\text{TF weighting} = \frac{1}{tfn+1} \cdot tfn \cdot \log_2(\frac{tfn}{\lambda})$$

IDF Weighting:

چون تو مدل‌های رتبه‌بندی مثل PL2 ، IDF معمولاً به این بستگی داره که یه واژه چقدر تو کل مجموعه سندها تکرار شده. یعنی هرچی یه واژه کمتر تو مجموعه ظاهر بشه، باید وزن بیشتری بگیره.

تو فرمول من، یه پارامتر داریم به اسم λکه تعریفش اینه:

$$\lambda = \frac{N}{c(w,C)}$$

که توش N تعداد کل سندهاست و $c(w,C)$ تعداد سندهایی که واژه w توشون ظاهر شده. این دقیقاً همون چیزیه که تو IDF استفاده می‌شه.

وقتی نگاه می‌کنم به این بخش از فرمول:

$$tfn \cdot \log_2(\frac{tfn}{\lambda})$$

یه $\log_2(\lambda)$ داریم که داره وزن واژه رو نسبت به کل مجموعه تنظیم می‌کنه. پس می‌تونم بگم:

$$\text{IDF weighting} = -\log_2(\lambda) = \log_2(\frac{N}{c(w,C)})$$

یعنی هرچی یه واژه کمتر تو مجموعه ظاهر شده باشه، این مقدار بزرگتر می‌شه و وزن بیشتری می‌گیره. این همون رفتاریه که از IDF انتظار داریم.

Length Normalization:

چون همون‌طور که تو فرمول اومده:

$$tfn = c(w,d) \cdot \log_2(1 + c \cdot \frac{avdl}{|d|})$$

اینجا مشخصه که طول سند $|d|$مستقیماً توی فرمول اومده. هرچی سند بلندتر باشه، این ضریب کوچیکتر می‌شه و باعث می‌شه وزن واژه کمتر بشه. این یعنی سندهای خیلی بلند فقط به خاطر طول زیادشون امتیاز بالا نمی‌گیرن.

پس می‌تونم بگم:

$$\text{Length Normalization} = \log_2(1 + c \cdot \frac{avdl}{|d|})$$

که این بخش توی تعریف tfn اومده و باعث می‌شه tf به‌درستی با طول سند تنظیم بشه.

پاسخ بخش دوم

این پارامتر میزان اثرگذاری نسبت طول میانگین سند به طول واقعی سند (یعنی $\frac{avdl}{|d|}$) مشخص می‌کند. به‌عبارتی، تعیین می‌کند که اختلاف طول یک سند نسبت به سایر اسناد تا چه حد باید در امتیازدهی نهایی آن سند نقش داشته باشد.

وقتی مقدار c زیاد باشد، مدل به‌طور ضمنی اسناد کوتاه را ترجیح می‌دهد، چون فرض می‌کند تکرار یک واژه در متن کوتاه‌تر نشانه‌ی ارتباط معنایی قوی‌تری است .در مقابل، اگر مقدار c کم باشد، مدل تفاوت‌های طولی بین اسناد را نادیده می‌گیرد و تمرکز اصلی‌اش بر تعداد دفعات ظاهر شدن واژه در سند خواهد بود.

پاسخ سوال دوم

پاسخ بخش اول

برای اینکه بتونیم احتمال‌های p_iو q_iرو در مدل RSJ محاسبه کنیم، باید بدونیم کدوم اسناد مرتبط هستن و کدوم نه. یعنی نیاز داریم به مجموعه‌ای از **RELEVANCE JUDGMENTS**که مشخص کنه هر سند نسبت به پرسش (QUERY)مرتبط هست یا نه.

وقتی این اطلاعات در دسترس باشه، می‌تونیم برای هر واژه A_i بررسی کنیم که:

- چند تا **سند مرتبط** شامل این واژه هست ← برای محاسبه‌ی p_i
- چند تا **سند نامرتبط** شامل این واژه هست ← برای محاسبه‌ی q_i

فرمول‌های تخمینی به شکل زیر هستن:

$$\hat{p}_i = \frac{\#(\text{RELEVANT DOCUMENTS CONTAINING } A_i) + 0.5}{\#(\text{TOTAL RELEVANT DOCUMENTS}) + 1}$$

$$\hat{q}_i = \frac{\#(\text{NON-RELEVANT DOCUMENTS CONTAINING } A_i) + 0.5}{\#(\text{TOTAL NON-RELEVANT DOCUMENTS}) + 1}$$

این فرمول‌ها از تکنیک **SMOOTHING**استفاده می‌کنن تا احتمال صفر یا یک مطلق نداشته باشیم.

اگر RELEVANCE JUDGMENTS نداشته باشیم:

در این حالت، چون نمی‌دونیم کدوم سند مرتبطه، مجبوریم فرض کنیم:

- مقدار p_i برای همه‌ی واژه‌ها یه عدد ثابته (مثلاً ۰.۵)
- مقدار q_i رو از روی فراوانی واژه‌ها در کل مجموعه اسناد تخمین می‌زنیم، با فرض اینکه همه‌ی اسناد نامرتبط هستن:

$$\hat{q}_i = \frac{\#(\text{DOCUMENTS CONTAINING } A_i) + 0.5}{\#(\text{TOTAL DOCUMENTS}) + 1}$$

پاسخ بخش دوم

با توجه به اینکه در این حالت مجموعه‌ای از *relevance judgments*در اختیار نداریم، نمی‌توانیم مقادیر p_i و q_i را مستقیماً از اسناد مرتبط و غیرمرتبط استخراج کنیم. بنابراین، مدل RSJ به صورت تقریبی بازنویسی می‌شود تا بتوانیم وزن واژه‌ها را فقط بر اساس فراوانی آن‌ها در کل مجموعه اسناد محاسبه کنیم.

فرمول تقریبی به شکل زیر است:

$$\text{Rank} \approx \sum_{i=1}^{k} \log\left(\frac{N - n_i + 0.5}{n_i + 0.5}\right)$$

در این فرمول:

- N :تعداد کل اسناد موجود در پایگاه داده است
- n_i :تعداد اسنادی است که واژه‌ی A_i در آن‌ها ظاهر شده
- k :تعداد واژه‌های مشترک بین پرسش (Query) و سند D

در مجموع، وقتی اطلاعات مرتبط بودن اسناد در دسترس نباشد، مدل RSJ به یک مدل آماری ساده تبدیل می‌شود که فقط بر اساس توزیع واژه‌ها در کل مجموعه اسناد عمل می‌کند.

پاسخ بخش سوم

اگر تعداد اسنادی که واژه A_i در آن‌ها ظاهر می‌شود (یعنی n_i) افزایش یابد، به این معناست که آن واژه در مجموعه‌ی اسناد رایج‌تر شده است. در چنین حالتی، مقدار وزن آن واژه در مدل RSJ کاهش پیدا می‌کند، چون حضورش اطلاعات خاصی درباره‌ی مرتبط بودن سند ارائه نمی‌دهد.

این رفتار مشابه بخش **IDF** در فرمول **BM25**است. در آن مدل، واژه‌هایی که در اسناد زیادی ظاهر می‌شوند، وزن کمتری دریافت می‌کنند تا تأثیرشان در رتبه‌بندی محدود شود. به‌طور خلاصه، هرچه واژه‌ای عمومی‌تر باشد، نقش آن در تشخیص ارتباط سند با پرس‌وجو کمتر خواهد بود.

فرمول تقریبی IDF در BM25 به شکل زیر است:

$$\mathrm{IDF}(w) = \log\left(\frac{M+1}{k}\right)$$

که در آن:

- M :تعداد کل اسناد موجود در پایگاه داده
- k :تعداد اسنادی که واژه‌ی مورد نظر در آن‌ها ظاهر شده

بنابراین، با افزایش k یا همان n_i، مقدار IDF کاهش می‌یابد و در نتیجه وزن واژه در امتیاز نهایی کمتر می‌شود.

پاسخ سوال سوم

پاسخ بخش اول

$$\text{BM25}(q,d) = \sum_{i=1}^{n} \text{IDF}(q_i) \times \frac{f(q_i,d) \times (k_1+1)}{f(q_i,d) + k_1 \times \left(1 - b + b \times \frac{|d|}{\text{avgdl}}\right)}$$

در فرمول BM25، یکی از عوامل مؤثر بر امتیاز نهایی، طول سند ($|d|$) نسبت به میانگین طول اسناد (avgdl) است. این تأثیر از طریق پارامتر b کنترل می‌شود. اگر مقدار b را صفر در نظر بگیریم، بخش مربوط به طول سند در مخرج فرمول حذف می‌شود و اثر تفاوت طول اسناد از بین می‌رود.

در این حالت، امتیاز BM25 فقط به تعداد دفعات ظاهر شدن واژه در سند بستگی دارد و طول سند هیچ نقشی در محاسبه نمره نهایی نخواهد داشت. به عبارتی، مدل فقط بر اساس فراوانی واژه‌ها تصمیم‌گیری می‌کند و همه‌ی اسناد صرفنظر از طولشان به یک شکل ارزیابی می‌شوند.

این تنظیم برای مواقعی مناسب است که نمی‌خواهیم طول سند باعث افزایش یا کاهش امتیاز شود و تمرکز فقط روی محتوای واژه‌هاست.

پاسخ بخش دوم

وقتی طول سند $|d|$ برابر با میانگین طول اسناد avgdl باشد، بخش نرمال‌سازی طول در فرمول BM25 عملاً بی‌اثر می‌شود. در این حالت، پارامتر b که وظیفه تنظیم تأثیر طول سند را دارد، نقشی در تغییر امتیاز نهایی ایفا نمی‌کند.

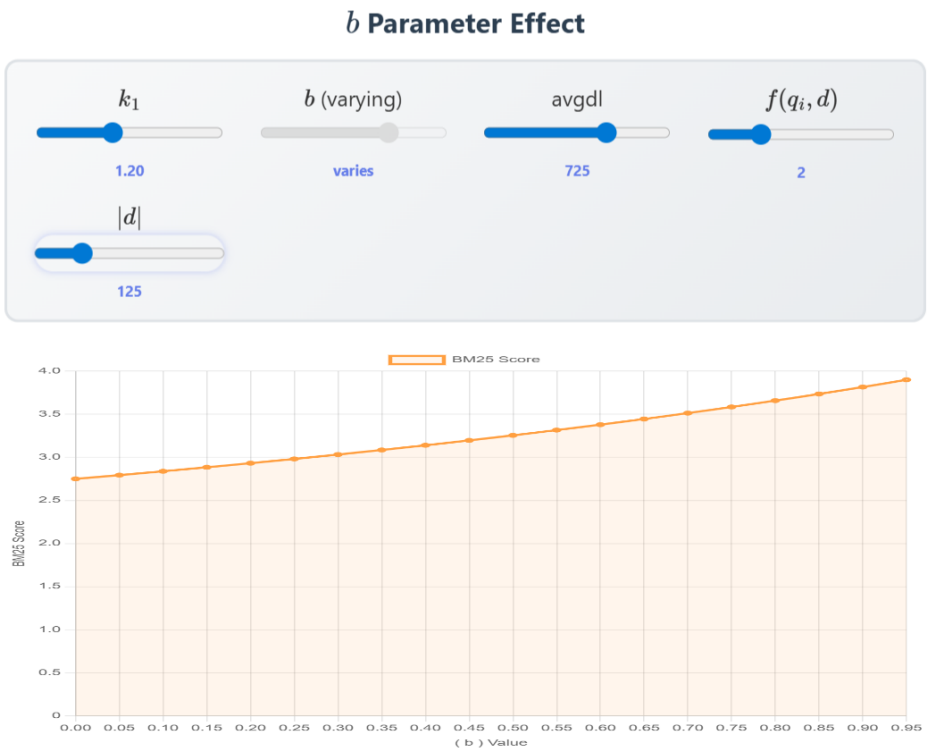
در نتیجه، فرمول ساده‌تر می‌شود و فقط وابسته به تعداد دفعات تکرار واژه در سند و پارامتر kخواهد بود. یعنی وزن هر واژه صرفاً بر اساس فراوانی آن در سند محاسبه می‌شود، بدون در نظر گرفتن طول سند.

فرمول ساده‌شده به شکل زیر است:

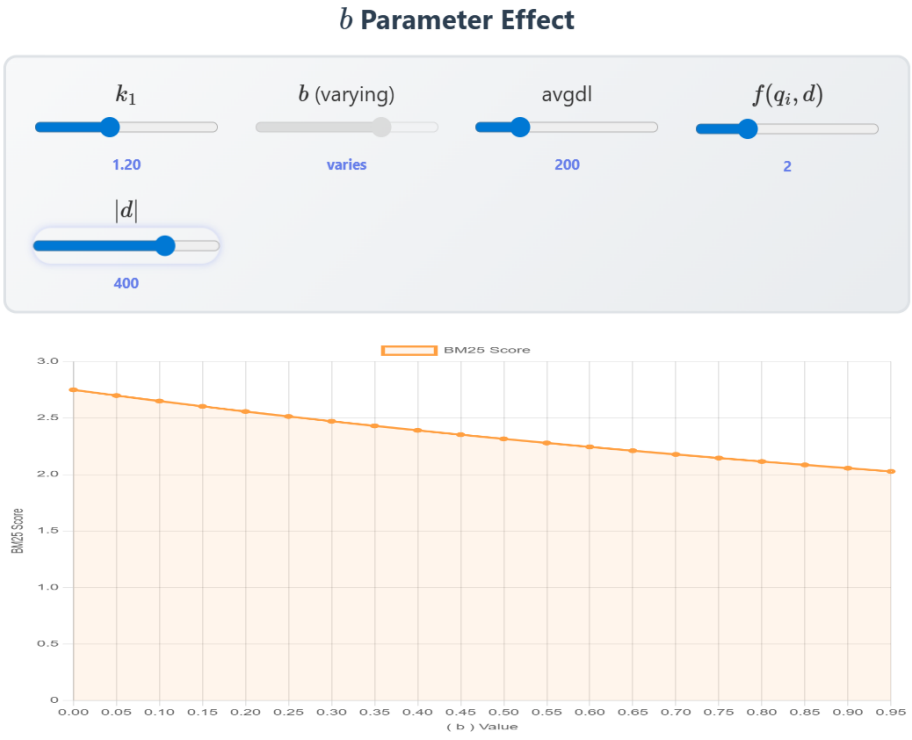
$$f(q,d) = \sum_{w \in q \cap d} \text{IDF}(w) \cdot \frac{c(w,d)(k+1)}{c(w,d)+k}$$

در این نسخه، فقط دو عامل اصلی باقی می‌مانند: **فراوانی واژه** و **پارامتر تنظیم‌کننده** k. این حالت برای زمانی مناسب است که بخواهیم همه‌ی اسناد را صرفنظر از طولشان، به شکل یکسان ارزیابی کنیم.

پـاسخ بـخش سـوم



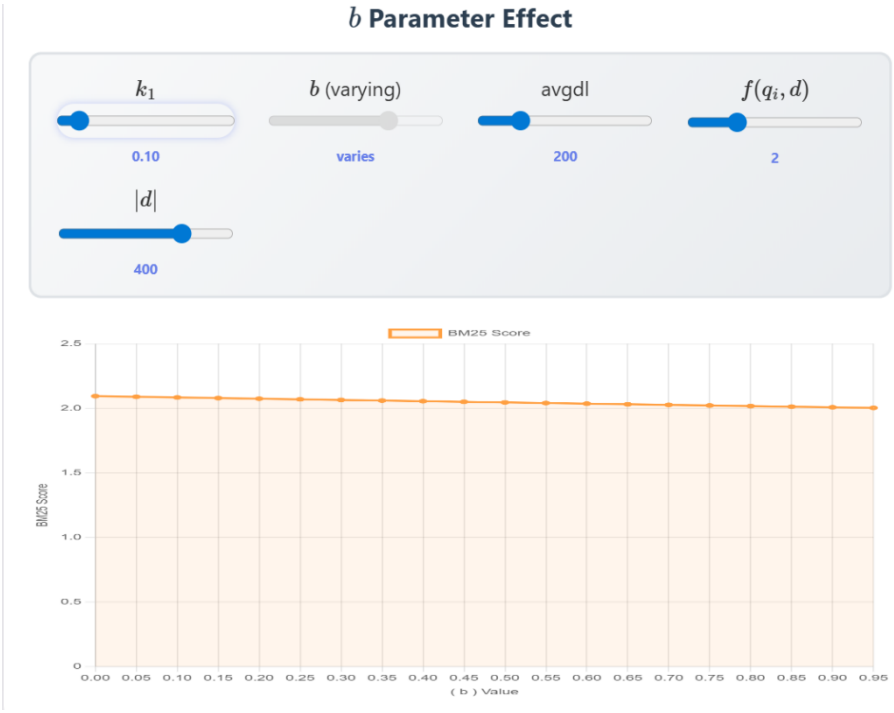
وقـتـی طول سند $|d|$ کـمـتـر از مـیـانـگـیـن طـول اسـنـاد avgdl بـاشـد، پـارامـتـر b بـاعث افـزایـش امـتـیـاز مـی‌شود (شیب مـثـبـت).

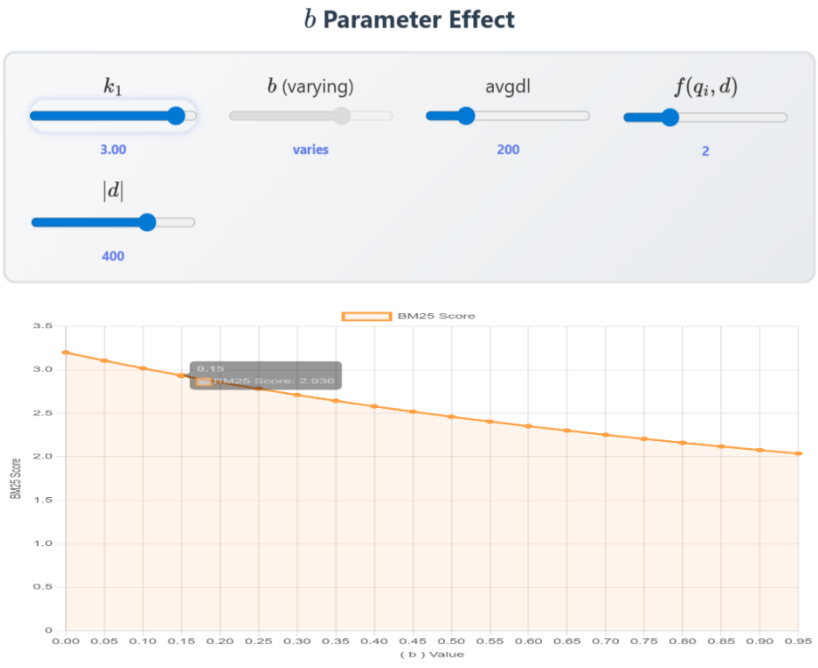


وقـتـی $|d|$ بـیـشـتـر از avgdl بـاشـد، امـتـیـاز کـاهش مـیـابـد (شیب مـنـفـی).

پاسخ بخش چهارم

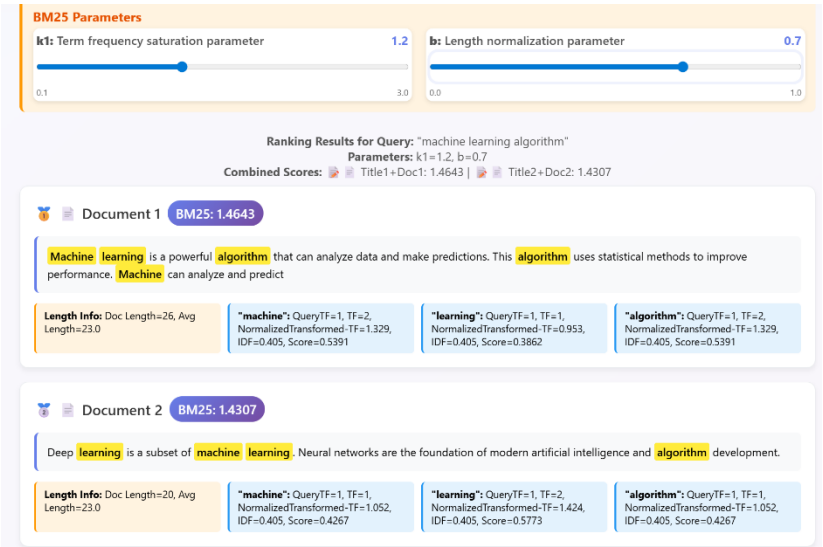
بر اساس نمودار مربوط به تأثیر پارامتر k_1، هرچه مقدار k_1بیشتر شود، حساسیت مدل نسبت به تغییرات پارامتر bنیز افزایش می‌یابد. یعنی شیب نمودار تندتر می‌شود و نوسانات امتیاز نهایی بیشتر تحت تأثیر قرار می‌گیرند.





پاسخ سوال چهارم

پاسخ بخش اول

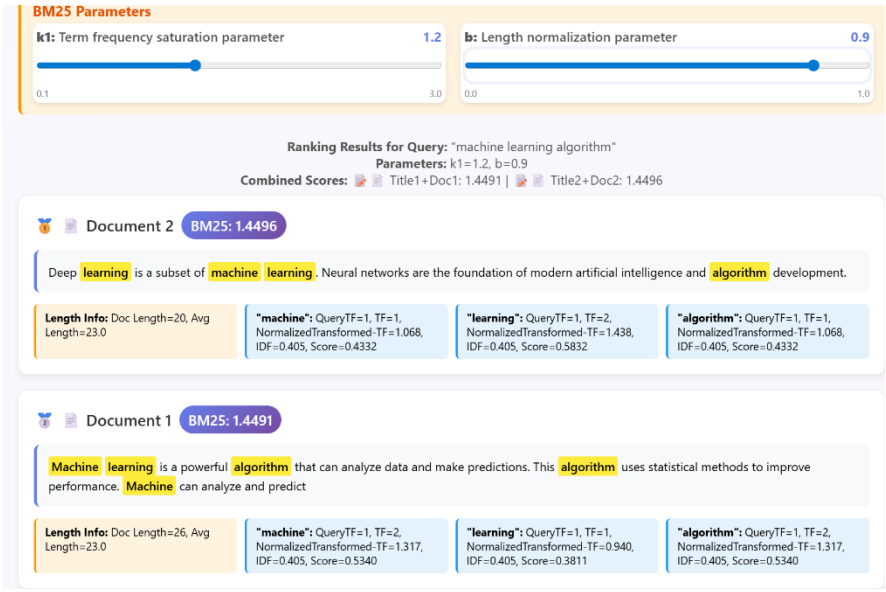


در Sample 1 ، دو سند با پرس‌وجوی "machine learning algorithm" بررسی شده‌اند. در حالت اول با تنظیمات $k_1 = 1.2$ و $b = 0.7$، سند اول امتیاز BM25 برابر با **1.4643** دارد و سند دوم **1.4307**، بنابراین سند اول به‌درستی در رتبه بالاتر قرار گرفته است.

اما وقتی مقدار b به **0.9** افزایش می‌یابد (با ثابت نگه‌داشتن $k_1 = 1.2$)، امتیاز سند دوم به **1.4496** می‌رسد و از سند اول که امتیاز **1.4491** دارد، پیشی می‌گیرد. این تغییر نشان می‌دهد که **افزایش** b باعث می‌شود مدل نسبت به **طول سند** حساس‌تر شود.

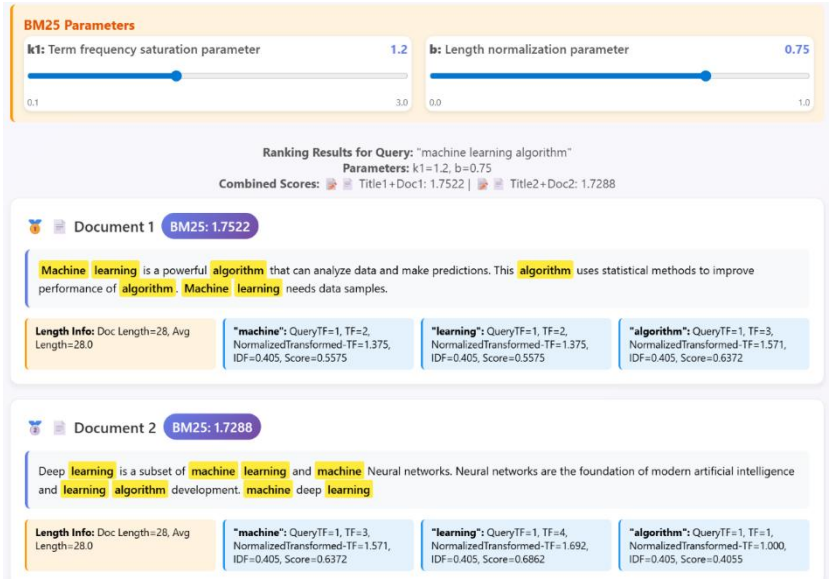
در این حالت، چون سند دوم کوتاه‌تر از میانگین است (۲۰ در برابر ۲۳)، افزایش b باعث تقویت امتیاز آن می‌شود. در مقابل، سند اول که بلندتر از میانگین است (۲۶ واژه)، با افزایش b امتیازش کمی کاهش می‌یابد.

بنابراین، اگر بخواهیم ترتیب رتبه‌بندی اسناد را تغییر دهیم، باید مقدار b را تنظیم کنیم. مقدار بحرانی در این مثال **0.9** است که باعث جابه‌جایی رتبه‌ها می‌شود. این یعنی **پارامتر تأثیرگذار در این نمونه** b است، نه k_1.



پاسخ بخش دوم

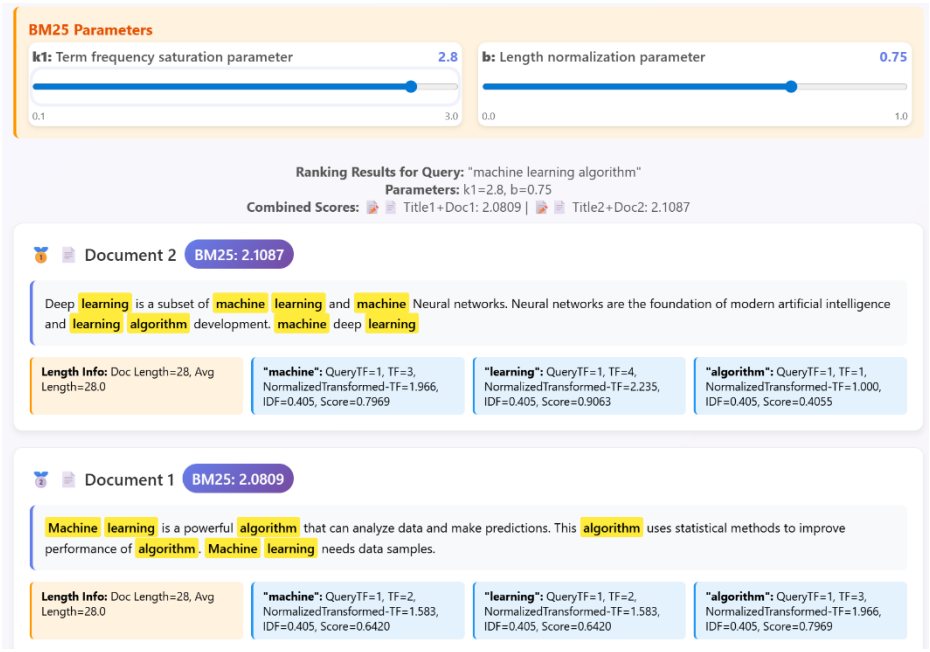
در حالت اول با تنظیمات $k_1 = 1.2$ و $b = 0.75$، سند اول امتیاز BM25 برابر با **1.7522** دارد و سند دوم **1.7288**، بنابراین سند اول به‌درستی در رتبه بالاتر قرار گرفته است.



اما وقتی مقدار k_1 به **3** افزایش می‌یابد (با ثابت نگه‌داشتن $b = 0.75$) امتیاز سند دوم به **2.1087** می‌رسد و از سند اول که امتیاز **2.0809** دارد، پیشی می‌گیرد. این تغییر نشان می‌دهد که **افزایش** k_1 باعث می‌شود مدل نسبت به **تکرار واژه‌ها** حساس‌تر شود.

در این حالت، سند دوم که واژه‌های "machine" و "learning" را بیشتر تکرار کرده، امتیاز بالاتری می‌گیرد حتی اگر واژه "algorithm" را کمتر داشته باشد. چون طول هر دو سند برابر با میانگین است (۲۸ واژه)، **پارامتر** b تأثیر خاصی ندارد و فقط k_1 عامل تعیین‌کننده است.

بنابراین، اگر بخواهیم ترتیب رتبه‌بندی اسناد را تغییر دهیم، باید مقدار k_1 را تنظیم کنیم. مقدار بحرانی در این مثال **2.1** است که باعث جابه‌جایی رتبه‌ها می‌شود



پاسخ بخش سوم

در نمونه‌ی سوم، به نظر می‌رسد که مدل BM25 واژه‌ها را به‌صورت مستقل در نظر می‌گیرد و تفاوتی بین جهت‌گیری عباراتی مثل "تهران به کیش" و "کیش به تهران" قائل نیست. از طرفی، چون سند مرتبط طول بیشتری دارد، امتیاز نهایی‌اش کمتر شده و در رتبه پایین‌تری قرار گرفته.

در نمونه‌ی چهارم، الگوریتم در تشخیص محتوای اسپم دقت کافی ندارد؛ صرفاً به خاطر تکرار زیاد یک واژه، سند نامرتبط امتیاز بالایی گرفته و بالاتر از سند اصلی قرار گرفته است.

در نمونه‌ی پنجم، مدل به معنای واژه‌ها توجهی ندارد. مثلاً واژه "آیفون" ممکن است در زمینه‌های مختلفی استفاده شود، اما چون در سند دوم بیشتر تکرار شده حتی اگر نامرتبط باشدامتیاز بیشتری گرفته و انتخاب شده است.

در نمونه‌ی ششم، سند دوم از نظر مفهومی مرتبطتر است، ولی چون واژه‌های پرس‌وجو با متن سند تطابق ندارند، امتیاز کمتری گرفته. این نشون می‌ده که وقتی پرسش و سند از واژه‌های متفاوت ولی مرتبط استفاده کنن، مدل نمی‌تونه ارتباط معنایی رو درست تشخیص بده.

پاسخ بخش چهارم

در مدلBM25 ، وقتی سند شامل بخش‌های مختلف مثل عنوان و متن باشد، امتیاز هر بخش جداگانه محاسبه می‌شود و برای رسیدن به نمره نهایی، معمولاً این امتیازها به‌صورت ساده با هم جمع می‌شوند. اما این روش ممکنه دقیق نباشه، چون تأثیر طول هر بخش یا تعداد تکرار واژه‌ها به‌درستی در نظر گرفته نمی‌شه و ممکنه باعث خطا در رتبه‌بندی بشه.

برای رفع این مشکل، می‌شه از مدل‌های پیشرفته‌تری مثل BM25F استفاده کرد. درBM25F ، به هر بخش سند وزن جداگانه داده می‌شه؛ مثلاً عنوان سند می‌تونه وزن بیشتری نسبت به متن اصلی داشته باشه. همچنین می‌تونیم برای هر بخش پارامترهای خاص خودش مثل k و b تعریف کنیم تا نرمال‌سازی دقیق‌تری انجام بشه.

در نتیجه، اگر بخوایم امتیازدهی دقیق‌تری برای اسناد چندبخشی داشته باشیم، استفاده از BM25F توصیه می‌شه، چون هم وزن‌دهی به بخش‌ها رو ممکن می‌کنه و هم نمره نهایی رو بر اساس ساختار واقعی سند تنظیم می‌کنه.

پاسخ بخش پنجم

به نظر من رتبه‌بندی انجام‌شده در Sample 8 دقیق نیست. با اینکه سند دوم شامل واژه‌هایی مرتبطتر با موضوع پرس‌وجو هست، سند اول در واقع پاسخ مستقیم به پرسش داده شده. دلیل این خطا اینه که در مدل، همه‌ی واژه‌ها وزن یکسانی گرفتن، در حالی که بعضی ترم‌ها مثل "مالزی" اهمیت بیشتری نسبت به واژه‌هایی مثل "پرواز" دارن.

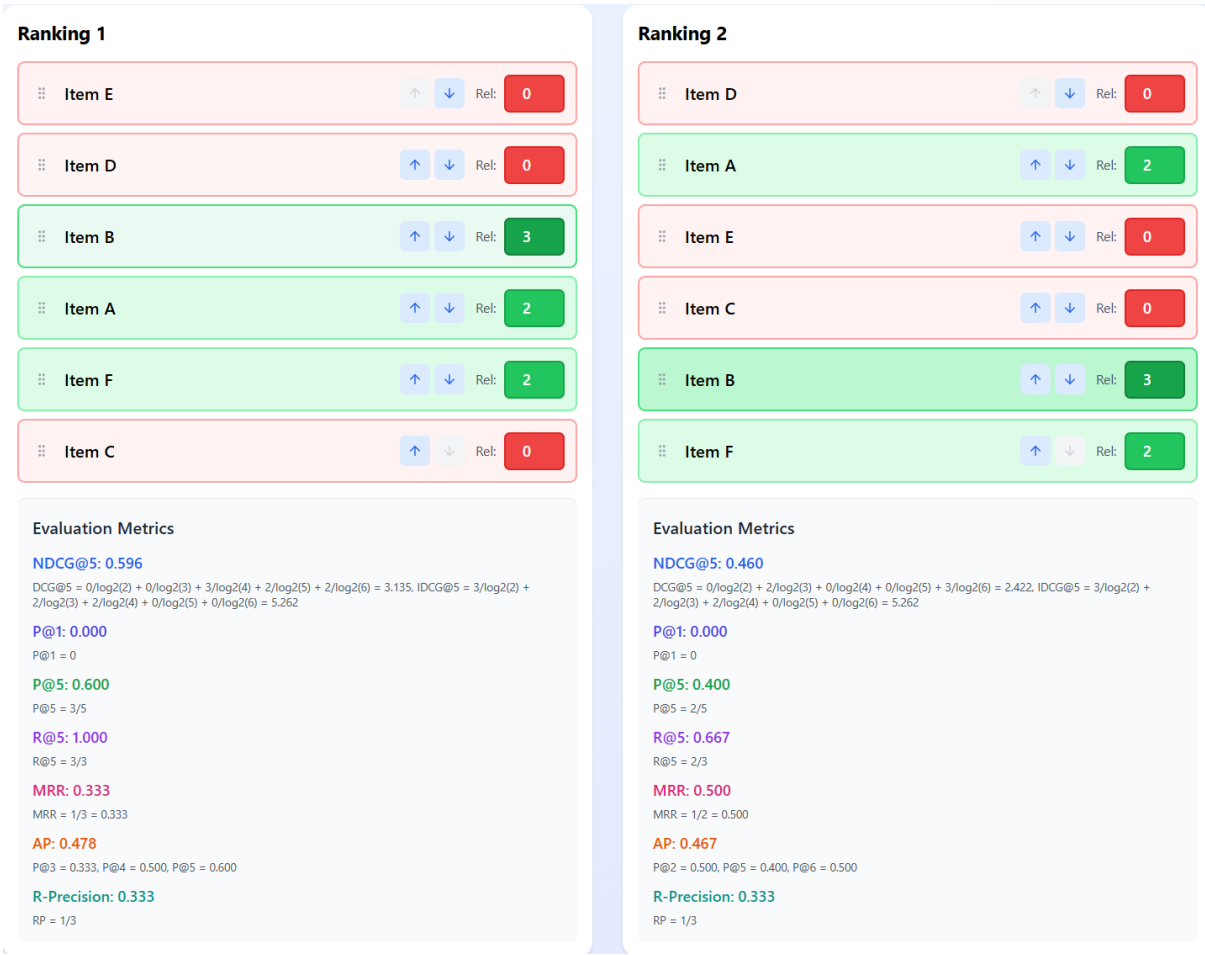
در اینجا مدل فقط بر اساس فراوانی و IDF تصمیم گرفته، ولی باید توجه داشت که IDFفقط میزان کمیاب بودن واژه‌ها رو در نظر می‌گیره، نه لزوماً اهمیت معنایی اون‌ها در پرس‌وجو. مثلاً ممکنه "مالزی" واژه‌ای رایج باشه و IDF پایینی داشته باشه، ولی از نظر معنایی برای پرسش کلیدی باشه و باید وزن بیشتری بگیره.

در نتیجه، برای رتبه‌بندی دقیق‌تر، بهتره علاوه برIDF ، وزن معنایی و نقش ترم‌ها در ساختار پرس‌وجو هم لحاظ بشه تا سند مرتبطتر واقعاً در رتبه بالاتر قرار بگیره.

پاسخ سوال پنجم

پاسخ بخش اول

تعداد آیتم های مرتب در هر دو نمونه ۱ است.



پاسخ بخش دوم

معیار MRR زمانی خیلی مفید و کاربردی میشه که هدف سیستم جستجو پیدا کردن سریع‌ترین پاسخ درست باشه. مثلاً توی سیستم های پرسش‌پاسخ یا وقتی فقط یک جواب مرتبط وجود داره، مهمه که اون جواب توی رتبه های اول باشه. چون MRR فقط به رتبه‌ی اولین پاسخ درست حساسه، می‌تونه نشون بده که مدل چقدر خوب تونسته جواب اصلی رو زود پیدا کنه.

فرمول MRR هم به این صورت تعریف میشه:

$$\text{MRR} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{r_i}$$

که توش:

- $|Q|$ تعداد کل پرس‌وجوهاست
- r_i رتبه‌ی اولین پاسخ درست برای پرس‌وجوی iهست

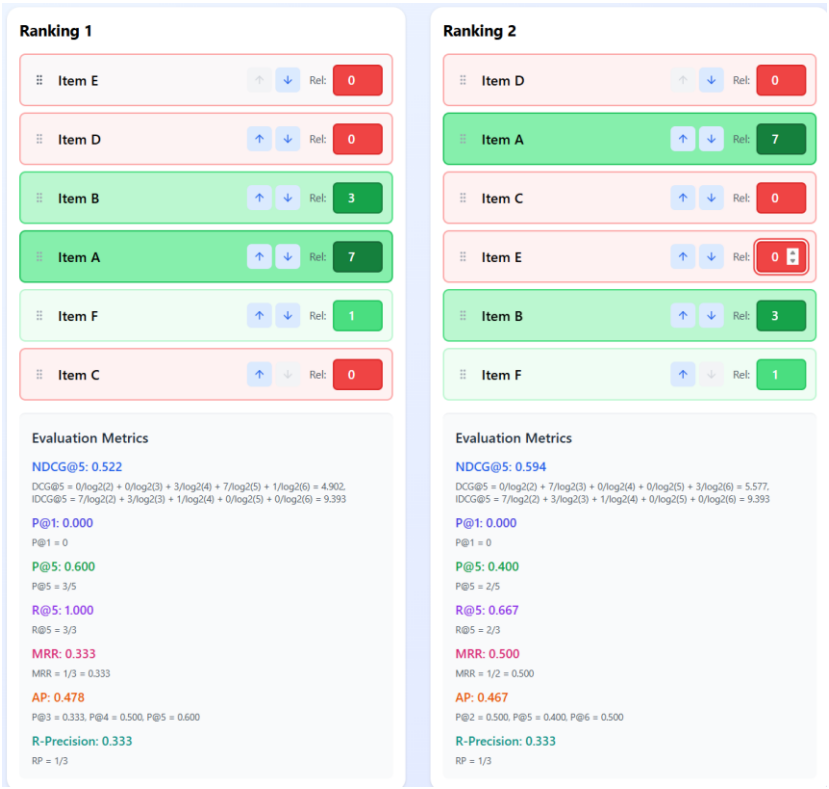
هرچی جواب درست زودتر ظاهر بشه، مقدار MRR بیشتر میشه و نشون میده که مدل عملکرد بهتری داشته. برای ارزیابی دقت در رتبه‌های بالا، این معیار خیلی خوب جواب میده.

پاسخ بخش سوم

بله، کاملاً ممکنه که مقدار **P@10** برای رتبه‌بند اول بیشتر باشه، ولی **NDCG@10** برای رتبه‌بند دوم بهتر باشه. دلیلش اینه که P@10 فقط تعداد آیتم‌های مرتبط در ۱۰ نتیجه اول رو می‌شماره، بدون توجه به اینکه اون آیتم‌ها در چه رتبه‌ای قرار گرفتن. اما NDCG@10 علاوه بر مرتبط بودن، جایگاه دقیق آیتم‌ها رو هم در نظر می‌گیره و به آیتم‌هایی که در رتبه‌های بالاتر باشن، امتیاز بیشتری میده.

مثلاً ممکنه رتبه‌بند اول ۵ آیتم مرتبط در رتبه‌های ۶ تا ۱۰ داشته باشه، ولی رتبه‌بند دوم فقط ۴ آیتم مرتبط داشته باشه که در رتبه‌های ۱ تا ۴ قرار گرفتن. در این حالت، P@10 برای رتبه‌بند اول بیشتره، ولی چون آیتم‌های مرتبط در رتبه‌بند دوم در جایگاه‌های بالاتری هستن، NDCG@10 برای اون بهتر میشه.

در نتیجه، این دو معیار جنبه‌های متفاوتی از کیفیت رتبه‌بندی رو می‌سنجن و ممکنه نتایج متضادی بدن.



پاسخ بخش چهارم

در ارزیابی کیفیت رتبه‌بندی، برخی معیارها به موقعیت دقیق آیتم‌های مرتبط در لیست حساس هستند، در حالی که برخی دیگر صرفاً تعداد آن‌ها را در نظر می‌گیرند.

- NDCG@k به‌طور خاص جایگاه آیتم‌های مرتبط را لحاظ می‌کند. در این معیار، امتیاز هر آیتم با افزایش رتبه کاهش می‌یابد، چون در محاسبه DCG از تابع لگاریتمی رتبه استفاده می‌شود. به همین دلیل، آیتمی که در رتبه‌های بالاتر قرار دارد (مثلاً رتبه ۲)، تأثیر بیشتری نسبت به همان آیتم در رتبه‌های پایین‌تر (مثل رتبه ۵) خواهد داشت.

- MRR تنها به رتبه اولین آیتم مرتبط توجه دارد. مقدار این معیار برابر با معکوس رتبه اولین پاسخ صحیح است، بنابراین هرچه این آیتم زودتر ظاهر شود، امتیاز بالاتری به مدل تعلق می‌گیرد.

- AP (Average Precision) میانگین دقت در نقاطی است که آیتم‌های مرتبط ظاهر شده‌اند. اگر این آیتم‌ها در رتبه‌های ابتدایی لیست باشند، دقت در آن نقاط بیشتر خواهد بود و میانگین نهایی افزایش می‌یابد. در نتیجه، AP نیز به ترتیب قرارگیری آیتم‌های مرتبط حساس است.

در مقابل، معیارهایی مانند P@k و R@k صرفاً تعداد آیتم‌های مرتبط در k رتبه اول را بررسی می‌کنند. این معیارها تفاوتی میان قرارگیری آیتم‌های مرتبط در رتبه‌های ۱ تا ۳ یا ۳ تا ۵ قائل نمی‌شوند؛ در هر دو حالت، مقدار P@5 یکسان خواهد بود.

همچنین، R-Precision که به‌صورت P@R تعریف می‌شود که R برابر تعداد کل آیتم‌های مرتبط است، فقط بررسی می‌کند که چند آیتم مرتبط در R رتبه اول قرار گرفته‌اند و به موقعیت دقیق آن‌ها توجهی نمیکند.

پاسخ بخش پنجم

بله، ممکنه مقدار P@k برای مدل اول بیشتر باشه ولی مقدار AP برای هر دو مدل برابر بشه. چون P@k فقط تعداد آیتم‌های مرتبط در k رتبه اول رو بررسی می‌کنه، ولی AP به دقت در رتبه‌هایی که آیتم‌های مرتبط ظاهر می‌شن حساسه. یعنی ممکنه مدل اول آیتم‌های مرتبط رو در رتبه‌های ۱ و ۵ داشته باشه و مدل دوم در رتبه‌های ۲ و ۵، که باعث می‌شه P@5 متفاوت باشه ولی میانگین دقت‌ها (AP) برابر بشه.

فرمول AP به این صورت تعریف می‌شه:

$$\mathrm{AP} = \frac{1}{R}\sum_{i=1}^{n} P(i)\cdot \mathrm{rel}(i)$$

که توش:

- R تعداد کل آیتم‌های مرتبطه
- $P(i)$ مقدار Precision در رتبه i
- $\mathrm{rel}(i)$ مشخص می‌کنه که آیتم در رتبه i مرتبط هست یا نه

چون در محاسبهAP ، هم دقت و هم تعداد آیتم‌های مرتبط لحاظ می‌شن، این معیار به نوعی به recallهم حساسه. اگه مدل فقط تعداد کمی از آیتم‌های مرتبط رو پیدا کنه، حتی با دقت بالا، مقدار AP پایین می‌مونه. پس برای مدل‌هایی که باید هم دقیق باشن و هم پوشش خوبی داشته باشن، APمعیار مناسبیه.

تایید می‌کنم که ازLLM ها مطابق با دستورالعمل های بارگذاری شده در سامانه Elearnدرس به طور مسئولانه استفاده کرده‌ام. تمام اجزای کار خود را درک می‌کنم و آماده بحث شفاهی درباره آنها هستم.