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: سهیل حاجیان منش

شماره دانشجویی: 810100119

مهر ماه ۱۴۰۳

فهرست

اظهارنامه

تأیید م یکنم که از LLM ها مطابق با دستورالعملهای بارگذاری شده در سامانه Elearn درس به طور مسئولانه استفاده کرده ام. تمام اجزای کار خود را درک میکنم و آماده بحث شفاهی درباره آنها هستم

پاسخ سوال اول

پاسخ بخش اول

فرمول BM25 :

$$TF\ weighting = \frac{(k_1+1)\ .\ c(w,d)\ .\ (k_3+1)\ .\ c(w,d)}{(k1((1-b)+b\ .\ \frac{|d|}{avdl})+c(w,d))(k_3+c(w,q))}$$

$$IDF\ weighting = \ \mathrm{LN}(\frac{N-df(w)+0.5}{df(w)+0.5})$$

$$Length\ Normalization = \frac{1}{k_1\left((1-b)+b\ .\ \frac{|d|}{avdl}\right)}$$

فرمول PL2 :

$$Length\ Normalization => tfn_w^d$$

داخل این فرمول تاثیر منفی (Penalize) افزایش طول سند اتفاق میفتد. در کل فرمول از این تابع استفاده شده تا این تاثیر منفی بالا بودن طول سند یا تاثیر مثبت پایین بودن طول آن کنترل شود.

$$IDF\ Weighting => \lambda_w^d$$

چون λ از فرکانس term در کل collection میاد، عملاً نقش inverse doc frequency رو ایفا می‌کنه. هر چه term کمیاب تر باشد مقدار λ بزرگتر میشود و بلعکس.

TF weighting در کل قسمت های فرمول نقش موثر را ایفا می کند.

پاسخ بخش دوم

همانطور که در قسمت قبل گفته شود پارامتر c موجود در tfn فرمول PL2 بر روی Length Normalization تاثیر میگذارد و کنترل میکند تا چه میزان طول اسناد نرمال سازی شود.

هر چقدر c بزرگتر شود، نرمال سازی بیشتری انجام می شود و تفاوت های بین طول سند با میانگین طول سند با ضرب c شدن بیشتر می شود. در نتیجه اسناد با طول بیشتر از میانگین، شدیدتر penalize می شوند و اسناد با طول کمتر بیشتر بهشون امتیاز داده میشه.

اما هرچقدر c کوچکتر باشد، نرمال سازی کمتری انجام می شود و رفتار به TF خام نزدیک تر می شود و سند های بزرگتر ممکن است امتیاز بیشتری از سند های کوچکتر بگیرند با این اینکه از لحاظ محتوا فرقی ندارند. در واقع خود TF اثر پر رنگ تری اینجا دارد.
هنگامی که اسنادمان تفاوت چندانی در طول با یکدیگر ندارند، میتوان مقدار c را کمتر گذاشت.

پاسخ سوال دوم

پاسخ بخش اول

نیاز به Relevance judgement هایی داریم که مشخص شود که هر سند به پرس و جو مرتبط است یا خیر تا سپس بتوانیم به ازای هر Ai موجود در یک سند ببینیم احتمال اینکه Ai در سند های مرتبط آمده باشد و احتمال اینکه در سند های نامرتبط آمده باشد چقدر است.

اگر relevance judgement موجود نباشد، فرض میکنیم :

۱. **pi یک مقدار ثابت است.**

۲. **qi را تخمین میزنیم با فرض اینکه تمامی اسناد موجود نامرتبط هستند. یعنی** qi برابر است با اینکه هر Ai در تمامی اسناد ظاهر شده است تقسیم بر کل اسناد.

پاسخ بخش دوم

مدل بصورت زیر ساده می شود:

$$\sum_{i=1,di=qi=1}^{k} \log\frac{pi(1-qi)}{qi(1-pi)} \approx \sum_{i=1,di=qi=1}^{k} \log\frac{1-qi}{qi} = \sum_{i=1,di=qi=1}^{k} \log\frac{1-\frac{ni}{N}}{\frac{ni}{N}} = \sum_{i=1,di=qi=1}^{k} \log\frac{N-ni}{ni}$$

$$\approx \sum_{i=1,di=qi=1}^{k} \log\frac{N-ni+0.5}{ni+0.5}$$

پاسخ بخش سوم

هرچه ni بزرگتر شود صورت لگاریتم کوچکتر و مخرج آن بزرگتر می شود که در نتیجه احتمال مرتبط بودن و وزن کلمه کاهش می یابد. این رفتار مشابه محاسبه IDF در الگوریتم BM25 است که کلمات هر چه در داکیومنت های بیشتری آمده باشند، مقدار IDF آنها کمتر می شود و وزن کمتری دارند.

پاسخ سوال سوم

پاسخ بخش اول

کافیست b = 0 باشد. چون طبق فرمول در مخرج داریم :

$$k_1\left((1-b)+b\,.\,\frac{|d|}{avdl}\right)$$

که اگر b = 0 باشد این مقدار همیشه برای k1 می شود.

پاسخ بخش دوم

خیر مانند بخش اول همواره k1 جواب عبارت نوشته شده می شود و فرمول به صورت زیر ساده می گردد :

$$\mathrm{LN}\left(\frac{N-df(w)+0.5}{df(w)+0.5}\right).\frac{(k_1+1)\,.\,c(w,d)}{k_1+c(w,d)}.\frac{(k_3+1)\,.\,c(w,q)}{k_3+c(w,q)}$$

پاسخ بخش سوم

چون **b** در مخرج است و محور **x** نمودار است پس ضریب آن اگر مثبت باشد، شیب نمودار منفی و ضریب آن اگر منفی باشد، شیب نمودار مثبت می شود. اگر **avdl** از **|d|** بزرگتر باشد، ضریب **b** منفی و شیب خط مثبت و اگر **avdl** از **| d|** کوچکتر باشد، ضریب **b** مثبت و شیب خط منفی میشود. در صورت برابر بودن نیز شیب خط صفر است.

پاسخ بخش چهارم

با افزایش **k۱** از ۰.۱ تا ۳ تاثیر تغییرات مقدار**b** بر امتیاز نهایی شدیدتر می شود و در نقطه **k۱=۳** بیشترین تاثیر تغییر **b** را داریم. دلیل آن هم این است که **k۱** در ضریب پشت متغیر**b** ضرب می شود و تاثیر مستقیم بر شیب دارد. هر چه **k۱** بیشتر باشد، ضریب پشت **b** بزرگتر می شود.

پاسخ سوال چهارم

پاسخ بخش اول

بله به وضوح سند اول تعریف مختصری از ماشین لرنینگ ارائه می دهد و سند دوم یکی از زیر مجموعه های ماشین لرنینگ را توضیح می دهد. پس سند اول با پرس و جو مرتبط تر است و به درستی تشخیص داده شده است.

متوسط طول اسناد ۲۳ و طول سند اول ۲۶ و طول سند دوم ۲۰ است.

با افزایش مقدار پارامتر **b** میتوان ترتیب رتبه بندی اسناد را تغییر داد. چون پارامتر **b** میزان جریمه کردن امتیاز اسناد بزرگ و افزایش امتیاز اسناد کوچکتر از میانگین را تنظیم می کند و هرچقدر بیشتر باشد، اسناد بزرگ تر از میانگین بیشتر جریمه می شوند. پس با رسیدن مقدار **b** از عدد ۰.۹ به بعد، سند دوم (که کوچکتر از میانگین است) در جایگاه ۱ قرار می گیرد و سند اول (که بزرگتر از میانگین است) در جایگاه دوم قرار می گیرد. در این حالت امتیاز سند دوم ۱.۴۳۷۳ و سند اول ۱.۴۳۶۷ می باشد.

پاسخ بخش دوم

بله کماکان رتبه بندی درست است و سند اول ارتباط معنایی بیشتری با پرس و جو دارد و به درستی در جایگاه اول قرار گرفته است.

این می توان با افزایش پارامتر **k۱** ترتیب را بهم زد. پارامتر**k۱** کنترل می کند تا چه اندازه تکرار کلمه در سند تاثیر گذار باشد. چون در سند دوم هم کلمه **machine** و هم کلمه **learning** بیشتر از سند اول تکرار شده و تنها کلمه**algorithm** در سند اول بیشتر از سند دوم تکرار شده است، پس می توان با افزایش **k۱** از مقدار ۲.۱ به بعد از آن سند دوم را در جایگاه اول قرار داد.

در این حالت (k=۲.۱) امتیاز سند دوم ۱.۹۶۹۱ و امتیاز سند اول ۱.۹۶۵۷ می باشد.

چون هر دو سند طول برابر دارند پارامتر b در اینجا بی ثمر است.

پاسخ بخش سوم

Sample 3 : سند دوم طول بیشتر از میانگین و سند اول طول کمتر از میانگین دارد در باقی ویژگی های هر کلمه هر دو سند رفتار یکسانی از خود نشان می دهند. پس هر چقدر هم که b کوچکتر باشد، بازهم امتیاز سند کوچکتر بهتر می شود یا در بهترین حالت (b=0) امتیاز یکسانی دارند. این موضوع به این دلیل است که BM25 به جایگاه و ترتیب کلمات در کنار هم کاری ندارد.

Sample 4 : bm25 هیچ کاری به معنی سند ندارد و تنها به میزان تکرار کلمات در سند توجه می کند به همین دلیل می توان با تکرار بی دلیل یک کلمه به میزان قابل توجه امتیاز بالایی در جست و جو کسب کرد. سند اول امتیازش به مراتب از سند دوم پایینتر است چون کلمه آیفون دو بار در آن تکرار شده و کلمه جدید هم در آن تکرار نشده است.

Sample 5 : باز هم تا حدودی مشابه قبل است، نمیتوان تشخیص داد که منظور از آیفون جدید، موبایل آیفون جدید است پس صرفا به کمک تعداد تکرار کلمات در اسناد رتبه بندی را مشخص می کند و چون b=0 است پس سند دوم رتبه بهتری دارد چون تعداد آیفون بیشتر در آن تکرار شده است.

Sample 6 : این الگوریتم تکرار های مترادف های یک کلمه (ماشین = خودرو، برقی = الکتریکی) را در نظر نمی گیرد به همین خاطر سند اول که کلمات مشابه بیشتری دارد رتبه بهتری گرفته است.(مشکل vocabulary mismatch)

پاسخ بخش چهارم

معمولا Title شامل اطلاعات مهم تری است و باید وزن دهی بیشتری نسبت به بدنه سند داشته باشد. به عنوان مثال TF=1 در title به طور معمول مهم تر از TF=1 در بدنه سند. بنابراین نمیتوان امتیاز Title و بدنه سند را با یک فرمول حساب کرد. پس ایده تجمیع امتیاز ها بصورت خام کار نمی کند همانطور که در مثال هم مشخص است(سند 1 مرتبط تر تشخیص داده شده است با وجود اینکه سند 2 در مورد آیفون 15 صحبت می کند.)

یک ایده می تواند این باشد که دوفرمول جداگانه برای محاسبه امتیاز title و body داشت و در نهایت امتیاز سند برابر است با تجمیع این دو امتیاز. در فرمول title ها length normalization خیلی کمتری نسبت به فرمول سابق داشته باشیم چون انتظار می رود title ها معمولا در یک محدوده طولی باشند. از طرفی مقدار k1 را بزرگتر بگیریم تا تکرار هر کلمه تاثیر بیشتری روی امتیاز title بگذارد.

مشکلات BM25 در اسناد چندبخشی

BM25 پایه‌ای فرض می‌کند سند یک متن بدون ساختار است.

ولی در وب / مقالات علمی / ... اسناد چند بخش دارند.(... title, body, abstract, anchors)

وقتی بخواهیم BM25 را روی این اسناد اجرا کنیم چند مشکل بزرگ ایجاد می‌شود:

1. اگر فیلدها را جدا score بدهیم و جمع کنیم → غیر خطی بودن TF از بین می‌رود → دوباره term independence برمی‌گردد (که غلط است).
2. IDF برای هر فیلد ثابت نیست، چون title کوتاه است و body بلند. در نتیجه آمار جهانی term در هر field متفاوت و unstableمی‌شود.
3. اگر وزن همه‌ی field ها = 1 بگیریم، باز همان حالت unstructured را برنمی‌گرداند چون رفتار TF غیرخطی است.
4. مشخص نیست length normalization را باید روی تک تک فیلدها بدهیم یا مجموع سند؟
5. پارامتر tuning هم مشکل می‌شود (برای هر field باید k1, b جدا optimize شود)

BM25F

هر field یک weight دارد مثلاً title=6 , body=2

TFهای field ها با وزن ضرب می‌شوند و یک pseudo-frequency ساخته می‌شود

سپس BM25 معمولی فقط روی همین pseudo-frequency اعمال می‌شود

اگر همه‌ی weight ها = 1 باشد، رفتار دقیقاً مثل یک سند unstructured واحد می‌شود.

توضیحات خلاصه شده BM25F از مقاله A Tutorial on the BM25F Model گرفته شده است.

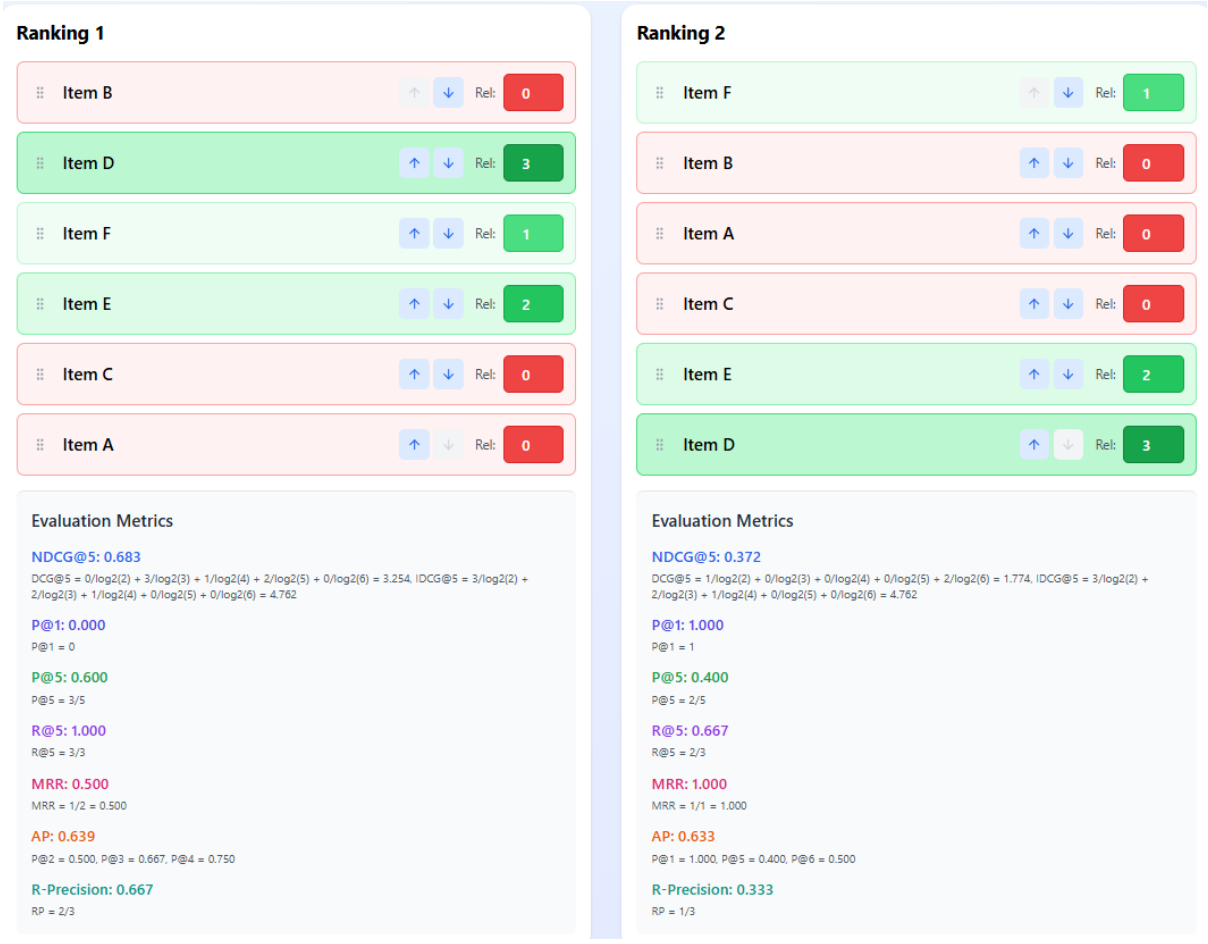
پاسخ بخش پنجم

رتبه بندی درست نیست اما چون کلمه پرواز IDF یکسانی با کلماتی مثل تهران ، مالزی و vip دارد ولی تعداد تکرار آن 2 است پس امتیاز بیشتری گرفته و همان باعث شده که رتبه سند دو بالاتر شود.

اهمیت توکن مالزی بیشتر است چون از لحاظ مفهومی فرد لزوما دنبال پرواز نیست و حتما دنبال پرواز به سمت مالزی است. پس مالزی مهم تر است ولی در معیار BM25 هر دو IDF یکسانی گرفته اند. پس لزوما وجود IDF نمیتواند تعیین کننده اهمیت یک term در جمله باشد و مفهوم کلمات در کنار هم نیز اهمیت دارد.

پاسخ سوال پنجم

پاسخ بخش اول

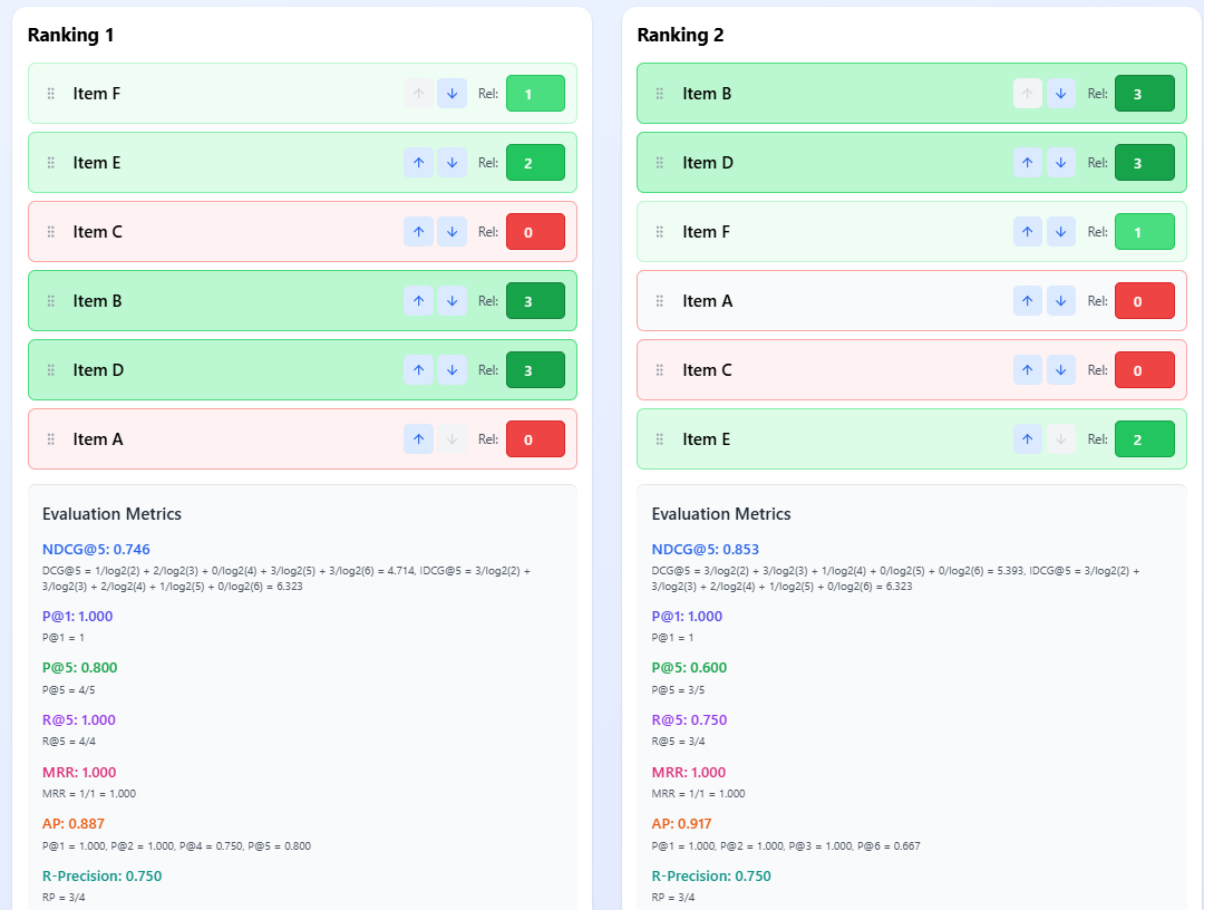


پاسخ بخش دوم

MRR زمانی مفید است که هدف پیدا کردن سریع اولین پاسخ درست باشد، مثل سیستم‌های Question Answering یا چت‌بات‌ها که مهم است جواب صحیح در رتبه‌های اول ظاهر شود.

پاسخ بخش سوم

بله مطالق دو مدل پیشنهاد شده عکس زیر :



پاسخ بخش چهارم

معیار NDCG محاسبه میکند که رتبه بندی انجام شده تا چه اندازه به رتبه ایده آل اسناد برای آن کوئری مشابهت دارد و یک امتیازی برای هر رتبه می دهد.

فرمولش به صورت روبرو است :

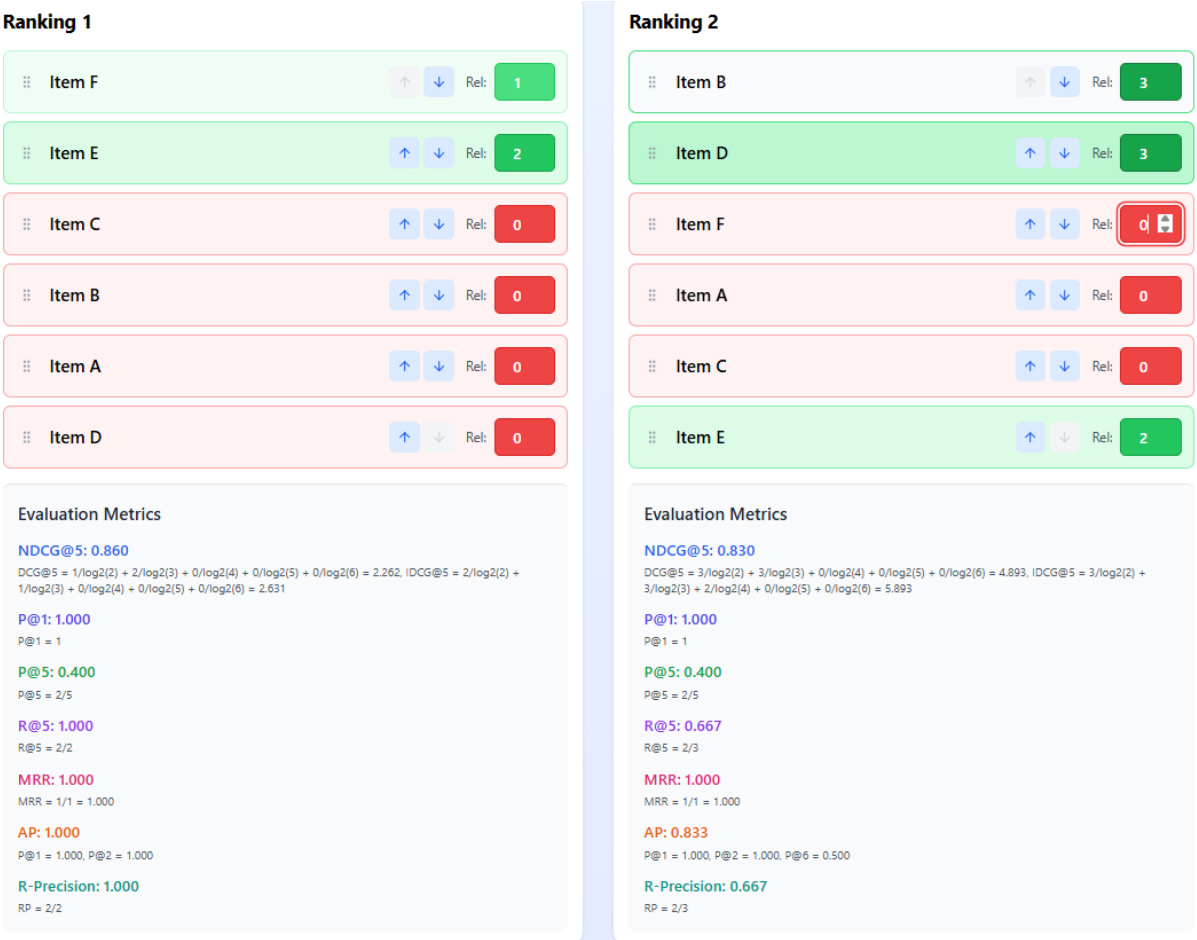
$$DCG = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log i + 1}$$

مقدار IDCG هم با همین فرمول ولی با اسناد بصورت ایده ال مرتب شده حساب می شود.

از تقسیم DCG بر IDCG مقدار NDCG رتبه بند محاسبه می گردد.

پاسخ بخش پنجم

دو مدل پیشنهادی :



فرمول AP :

$$AP = \frac{1}{R}\sum_{k=1}^{n} P\,(k).rel(k)$$

AP به recall حساس است چون تعداد کل اسناد مرتبط در محاسبه میانگین دقت وارد می‌شود و اگر تعداد اسناد مرتبط بیشتری در کل موجود باشد، هر سند مرتبط رتبه پایین‌تر تاثیر کمتری روی ap خواهد داشت.