به نام خدا

دانشگاه تهران

دانشکدگان فنی

دانشکده مهندسی برق و کامپیوتر

درس بازیابی هوشمند اطلاعات

محمد ناصری

۸۱۰۱۰۰۴۸۶

تمرین دوم

آبان ۱۴۰۰

مقدمه

در تمرین اول با معیارهای ارزیابی و توابع امتیازدهی به اسناد آشنا شدید. دیدید که یک تابع امتیازدهی با توجه به میزان ارتباط یک سند با پرس وجو، امتیازی به سند تخصیص میدهد تا در نهایت اسناد براساس امتیازشان، رتبه بندی و نمایش داده شوند. در این تمرین قصد داریم، روشهای مختلف هموارسازی توابع بازیابی و پارامترهای آنها را مورد مطالعه قرار بدیم و همچین به بسط پرسوجوی کاربر با استفاده از روش Psuedo Relevance Feedback خواهیم پرداخت

سوال اول : بررسی روش‌های هموارسازی

ابزار گالاگو، به صورت پیش فرض بازیابی را به روش Query-Likelihoodانجام میدهد. هدف از این سوال آشنایی با روشهای هموارسازی و مقایسه تاثیر هر یک بر روی کیفیت رتبه بندی میباشد. یکی از مشکلات مطرح در حوزه بازیابی اطلاعات، وجود احتمالهای صفر است که محاسبات را در عمل دچار مشکل میکند. روشهای هموارسازی برای حل این مشکل مطرح شدند تا احتمال رخداد کلمات دیده نشده پرسوجو در اسناد را تخمین بزنند.

روشهایی که قصد داریم در این تمرین مورد بررسی قرار دهیم عبارتند از:

- روش JM با پارامتر λ
- روش Dirichlete Prior با پارامتر π
- روش Additive Smoothing

هدف ما مطالعه رفتار روش های هموارسازی و همچنین مقایسه روش های مختلف است. همانطور که مشخص است، عملکرد یک الگوریتم بازیابی ممکن است به طور قابل توجهی با توجه به مجموعه آزمایشی مورد استفاده متفاوت باشد. به طور کلی داشتن مجموعه های بزرگتر و پرس و جوهای بیشتر مطلوب است. ما از یک مجموعه داده TREC و مجموعه ۵۰ پرس و جو (۵۱ تا ۱۰۰) استفاده میکنیم. همچنین تعداد اسناد بازیابی شده مدنظر ۱۰۰۰ عدد میباشد. در این آزمایش تنها روش tokenization استفاده شده stemming با porter-stemmer میباشد. برای هر روش هموارسازی برای هر مجموعه، ما با طیف گسترده ای از مقادیر پارامتر آزمایش می کنیم.

به منظور مطالعه رفتار یک روش هموارسازی، مجموعه‌ای از مقادیر پارامتر نماینده را انتخاب می‌کنیم و دقت بازیابی اسناد را با توجه به معیارها ارزیابی MAP و P@20 بررسی می‌کنیم.

مرحله اول: LM With Jelineck-Mercer Smoothing

اولین رویکردی که می توانیم انجام دهیم این است که یک مدل مخلوط[1] با هر دو توزیع[2] ایجاد کنیم:

$$P(q|\boldsymbol{\theta}_d) = \prod_i \big((1-\alpha)P(q_i|\boldsymbol{\theta}_d) + \alpha P(q_i|C)\big)$$

[1] Mixture
[2] Distribution

احتمال بدست آمده از سند را با فرکانس کلمه در مجموعه کلی [3]مخلوط می کند.

مقدار α را به ثابت λ تبدیل میکنیم و از تقریب maximum-likelihood استفاده میکنیم.

$$P(q|\boldsymbol{\theta}_d) = \prod_i \left((1-\lambda)\frac{freq(q_i,d)}{|d|} + \lambda\frac{freq(q_i,C)}{|C|} \right)$$

برای رتبه‌بندی از لگاریتم استفاده میکنیم:

$$Score(q,d) = \sum_i \log\left((1-\lambda)\frac{freq(q_i,d)}{|d|} + \lambda\frac{freq(q_i,C)}{|C|} \right)$$

- ارزش بالای λ : جستجو تمایل به بازیابی اسنادی دارد که حاوی تمام کلمات پرس و جو هستند. در واقع، اگر λ به یک نزدیک شود، تمام وزن‌های عبارت به صفر می‌رسد و فرمول امتیازدهی به Coordination-level-matching نزدیک می‌شود، که به سادگی تعداد عبارت‌های منطبق است.
- مقدار کم λ : جداکننده تر[4]، مناسب برای پرس و جوهای طولانی. یک λ کوچک به معنای تأکید بیشتر بر وزن دهی نسبی است.
- انتخاب مقدار مناسب λ در این روش برای کارایی بهتر، از اهمیت بالایی برخوردار است. تفاوت در مقدار λ بهینه نشان می دهد که پرس و جوهای طولانی نیاز به هموارسازی بیشتری دارند و تأکید کمتری بر وزن نسبی عبارت ها می شود.

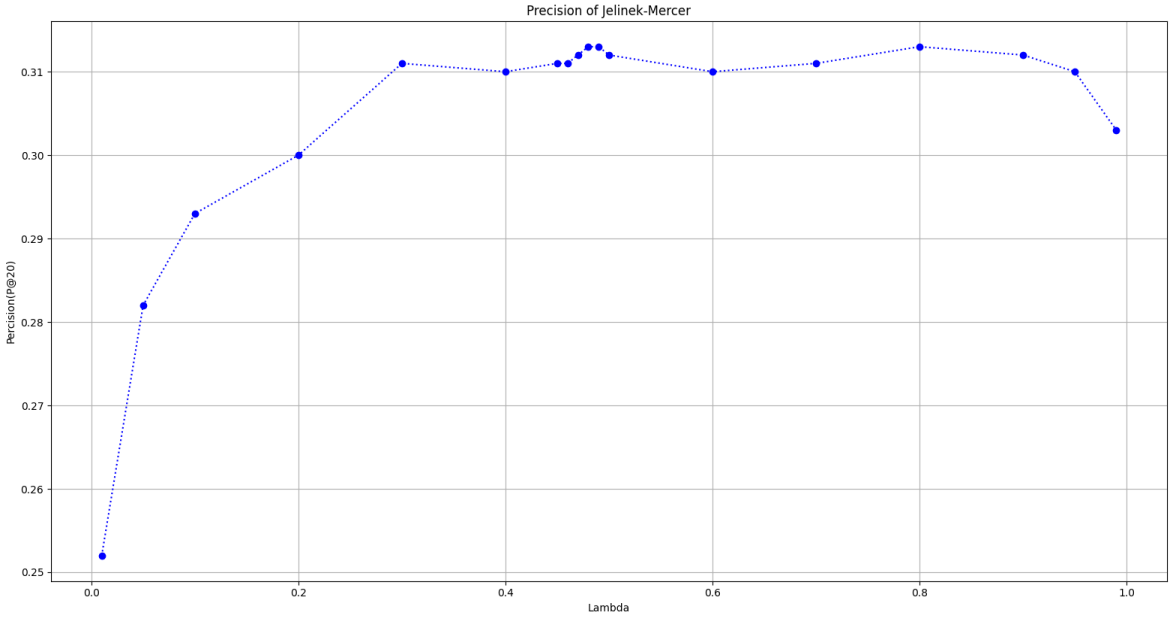
[3] General Collection Frequency
[4] More Disjunctive

برای مقادیر تست از مجموعه [0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.99] استفاده میکنیم و پس از پیدا کردن بازه مناسب‌تر از گام‌های کوتاه‌تر در این بازه استفاده میکنیم.

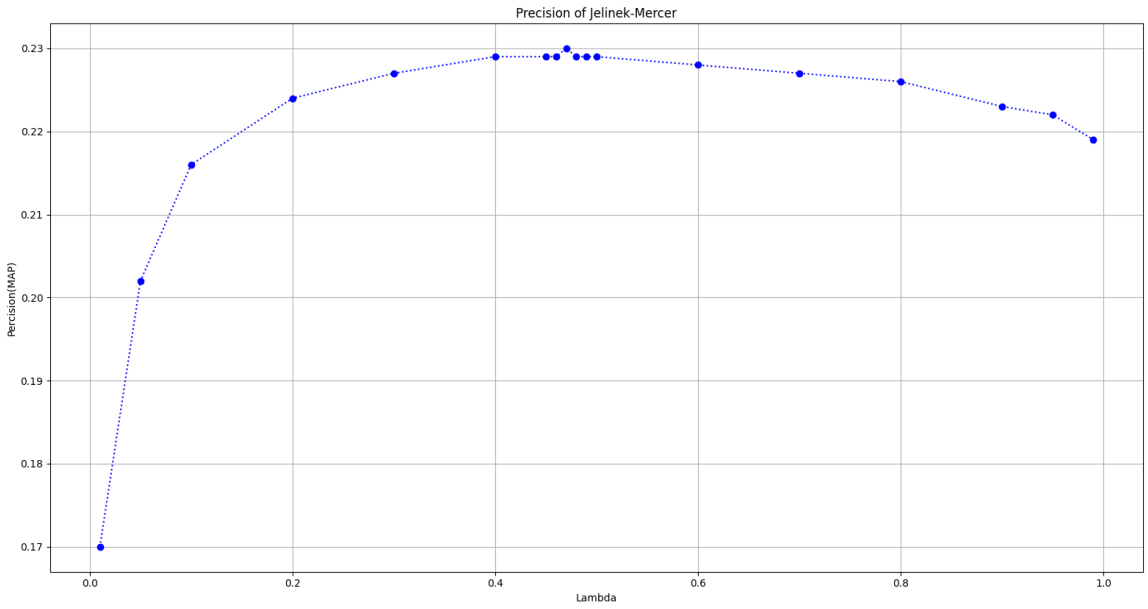
	run-id	lambda	map		P20	
2	/home/mamathew/Desktop/JMTest01.txt	0.01	0.170		0.252	
3	/home/mamathew/Desktop/JMTest02.txt	0.05	0.202	*	0.282	*
4	/home/mamathew/Desktop/JMTest03.txt	0.1	0.216	*	0.293	*
5	/home/mamathew/Desktop/JMTest04.txt	0.2	0.224	*	0.300	*
6	/home/mamathew/Desktop/JMTest05.txt	0.3	0.227	*	0.311	*
7	/home/mamathew/Desktop/JMTest06.txt	0.4	0.229	*	0.310	*
8	/home/mamathew/Desktop/JMTest07.txt	0.45	0.229	*	0.311	*
9	/home/mamathew/Desktop/JMTest08.txt	0.46	0.229	*	0.311	*
10	/home/mamathew/Desktop/JMTest09.txt	0.47	0.230	*	0.312	*
11	/home/mamathew/Desktop/JMTest10.txt	0.48	0.229	*	0.313	*
12	/home/mamathew/Desktop/JMTest11.txt	0.49	0.229	*	0.313	*
13	/home/mamathew/Desktop/JMTest12.txt	0.5	0.229	*	0.312	*
14	/home/mamathew/Desktop/JMTest13.txt	0.6	0.228	*	0.310	*
15	/home/mamathew/Desktop/JMTest14.txt	0.7	0.227	*	0.311	*
16	/home/mamathew/Desktop/JMTest15.txt	0.8	0.226	*	0.313	*
17	/home/mamathew/Desktop/JMTest16.txt	0.9	0.223	*	0.312	*
18	/home/mamathew/Desktop/JMTest17.txt	0.95	0.222	*	0.310	*
19	/home/mamathew/Desktop/JMTest18.txt	0.99	0.219	*	0.303	*
20	Sig-Test: randomized, threshold set to 0.050000					

جدول 1 جدول نتایج آزمایش مقادیر لامبدا

همانطور که در مقادیر جدول بالا مشاهده میشود، بازه مناسب و بهینه برای λ بازه بین 0.4 و 0.5 میباشد که با گام 0.01 به آزمایش میپردازیم تا نتیجه بهینه یعنی مقدار 0.47 با بالاترین نتایج از لحاظ معیارهای ارزیابی، حاصل شود. از نتایج بدست آمده حاصل میشود که تعادل میان وزن‌دهی نسبی و وزن‌دهی بر اساس تعداد تطابق در این پرس و جوها و اسناد بصورت تقریبا یکنواخت لازم به استفاده هستند تا نتیجه مطلوب حاصل شود. گرچه از دید کاربر نهایی (معیار ارزیابی P@20) با اعتبار دادن بیشتر به تعداد تطابق بین سند و پرس و جو نیز میتوان به تعداد اسناد بازیابی شده بیشتر در ابتدای بازیابی رسید (با تنظیم مقدار 0.8 برای λ در معیار P@20 مقدار بهینه حاصل میشود). ولی درحالت کلی برای یافتن تعداد بیشتر اسناد مرتبط در پایان بازیابی نیاز است تا مقدار بهینه λ بر اساس معیار MAP بررسی و قرار داده شود.



شکل 1 نمودار P@20 برای مقادیر JM



شکل 2 نمودار Map برای مقادیر JM

مرحله دوم: LM with Dirichlete Smoothing

$$P(q|\boldsymbol{\theta}_d) = \prod_i \left((1-\alpha)P(q_i|\boldsymbol{\theta}_d) + \alpha P(q_i|C)\right)$$

ایده اصلی:

ترم هایی که در یک سند طولانی وجود ندارد باید به احتمال هموارسازی کم اختصاص داده شود (هر چه سند طولانی تر باشد، این احتمال کمتر است)

مقدار α را برابر $\frac{\mu}{\mu+|d|}$ قرار میدهیم و از تقریب maximum-likelihood استفاده میکنیم:

$$P(q|\boldsymbol{\theta}_d) = \prod_i \left(\frac{freq(q_i,d) + \mu \cdot freq(q_i,C)/|C|}{\mu + |d|}\right)$$

$$Score(d,q) = \sum_i \log\left(\frac{freq(q_i,d) + \mu \cdot freq(q_i,C)/|C|}{\mu + |d|}\right)$$

در هموارسازی دیریکله: مقدار Dirchlet Prior بر روی پارامترهای مدل فرض می شود و likelihood به صورت چندجمله ای توزیع می شود.

هنگام استفاده از دیریکله برای هموارسازی، می بینیم که α در فرمول بازیابی، وابسته به سند است. برای اسناد طولانی کوچکتر است، بنابراین می تواند به عنوان یک جزء length-normalization تفسیر شود که اسناد طولانی را جریمه می کند. در روش JM وزن ترم‌ها یک length-normalization ضمنی در $P(q_i \mid d)$ داشتند ولی در اینجا وزن ترم‌ها فقط تحت تاثیر تعداد خام ترم‌ها هستند و نه طول سند.

زمانی که از μ کوچکتر استفاده می کنیم بر وزن نسبی ترم‌ها تاکید بیشتری می شود. با بزرگ شدن μ، α به 1 میل می کند و تمام وزن های ترم‌ها به صفر میل می کنند و فرمول امتیازدهی مانند قبل به تعداد انطباق ترم‌های پرس و جو و سند تاکید بیشتری دارد.

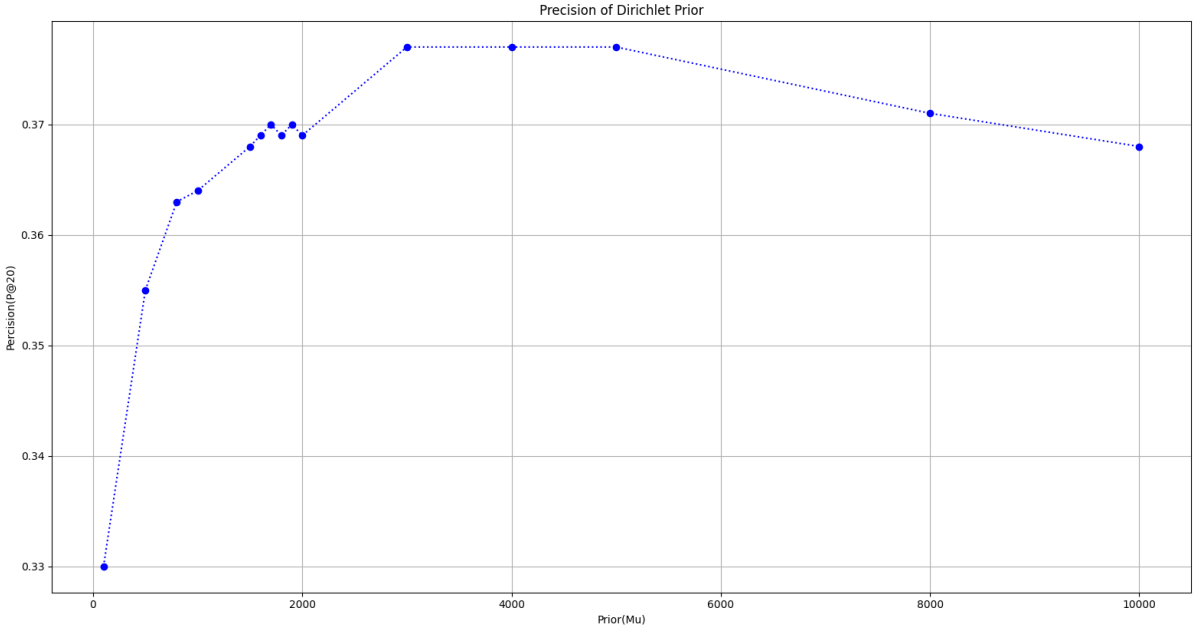
	mu	map		P20		num_ret	num_rel	num_rel_ret	
/Q1_2/DirichletTest01.txt	100	0.231		0.330		44400.000	6100.000	3020.000	
/Q1_2/DirichletTest02.txt	500	0.247	*	0.355	*	44400.000	6100.000	3145.000	*
/Q1_2/DirichletTest03.txt	800	0.251	*	0.363	*	44400.000	6100.000	3179.000	*
/Q1_2/DirichletTest04.txt	1000	0.252	*	0.364	*	44400.000	6100.000	3205.000	*
/Q1_2/DirichletTest05.txt	1500	0.253	*	0.368	*	44400.000	6100.000	3239.000	*
/Q1_2/DirichletTest06.txt	1600	0.254	*	0.369	*	44400.000	6100.000	3242.000	*
/Q1_2/DirichletTest07.txt	1700	0.254	*	0.370	*	44400.000	6100.000	3247.000	*
/Q1_2/DirichletTest08.txt	1800	0.254	*	0.369	*	44400.000	6100.000	3249.000	*
/Q1_2/DirichletTest09.txt	1900	0.253	*	0.370	*	44400.000	6100.000	3250.000	*
/Q1_2/DirichletTest10.txt	2000	0.253	*	0.369	*	44400.000	6100.000	3250.000	*
/Q1_2/DirichletTest11.txt	3000	0.250	*	0.377	*	44400.000	6100.000	3259.000	*
/Q1_2/DirichletTest12.txt	4000	0.250	*	0.377	*	44400.000	6100.000	3259.000	*
/Q1_2/DirichletTest13.txt	5000	0.245	*	0.377	*	44400.000	6100.000	3251.000	*
/Q1_2/DirichletTest14.txt	8000	0.240		0.371	*	44400.000	6100.000	3230.000	*
/Q1_2/DirichletTest15.txt	10000	0.236		0.368	*	44400.000	6100.000	3224.000	*

threshold set to 0.050000

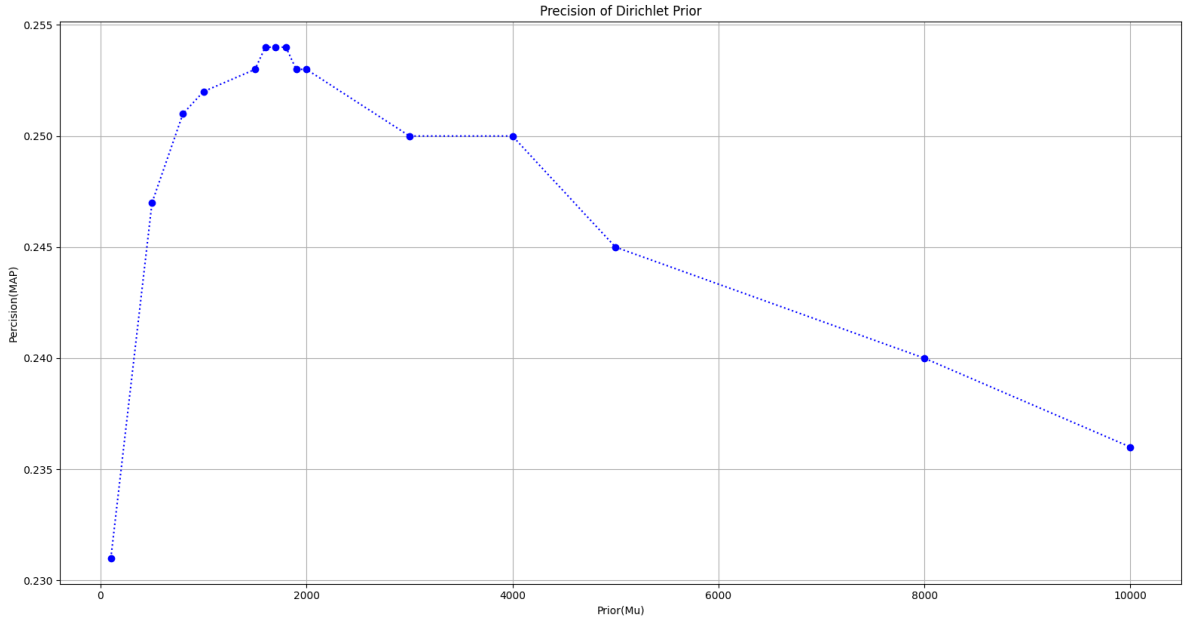
جدول 2 جدول نتایج آزمایش مقادیر μ

همانطور که از جدول بالا و نمودارهای بعدی مشخص است از لحاظ معیار ارزیابی Map مقدار بهینه μ برابر ۱۷۰۰ میباشد این در حالیست که از لحاظ معیار P@20 و تعداد کل اسناد مرتبط بازگردانده شده مقدار بهینه برابر ۳۰۰۰ میباشد. مشخص است که از دید کاربر نهایی مقدار ۳۰۰۰ میتواند مناسب‌تر باشد در حالی که مقدار ۱۷۰۰ در حالت کلی مقدار اندکی دقت بهتری را در جستجو دارد که اختلاف آن با ۳۰۰۰ = μ مقدار اندک و قابل چشم پوشی‌ست و ما مقدار ۳۰۰۰ را به عنوان مقدار بهینه μ در نظر میگیریم.

تفاوت مقدار بهینه بدست آمده با مقدار پیش‌فرض (۱۵۰۰) نشان از نیاز به تاکید بیشتر بر تعداد انطباق ها در پرس و جو برای رسیدن به جواب بهتر دارد.



شکل 3 نمودار P@20 برای مقادیر Dirichlet



شکل 4 نمودار Map برای مقادیر Dirichlet

مرحله سوم: LM with Additive Smoothing

در این بخش به بررسی روش Additive Smoothing (یا Laplace Smoothing) میپردازیم.

Psuedo Count

مقداری است که به تعداد موارد مشاهده‌شده اضافه می‌شود تا احتمال مورد انتظار در مدلی از داده‌ها را تغییر دهد، و از صفر شدن جلوگیری کند.

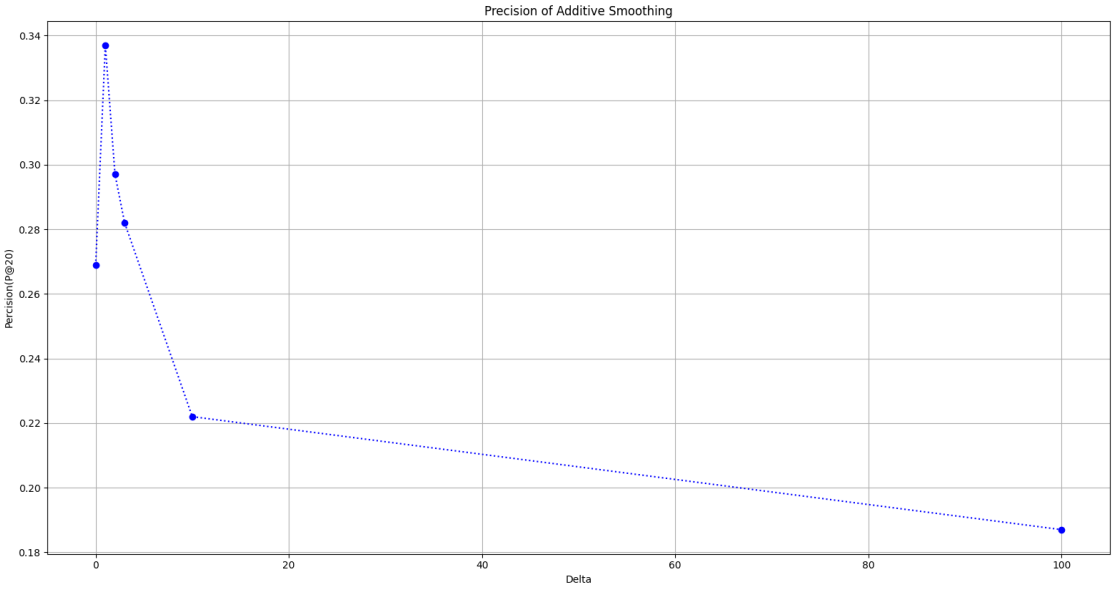
$$p(w|\theta) = \frac{c(w,D) + \delta}{|D| + \delta|V|}$$

اگر $\delta = 1$ آنگاه به آن "+1 Smoothing" گفته میشود.

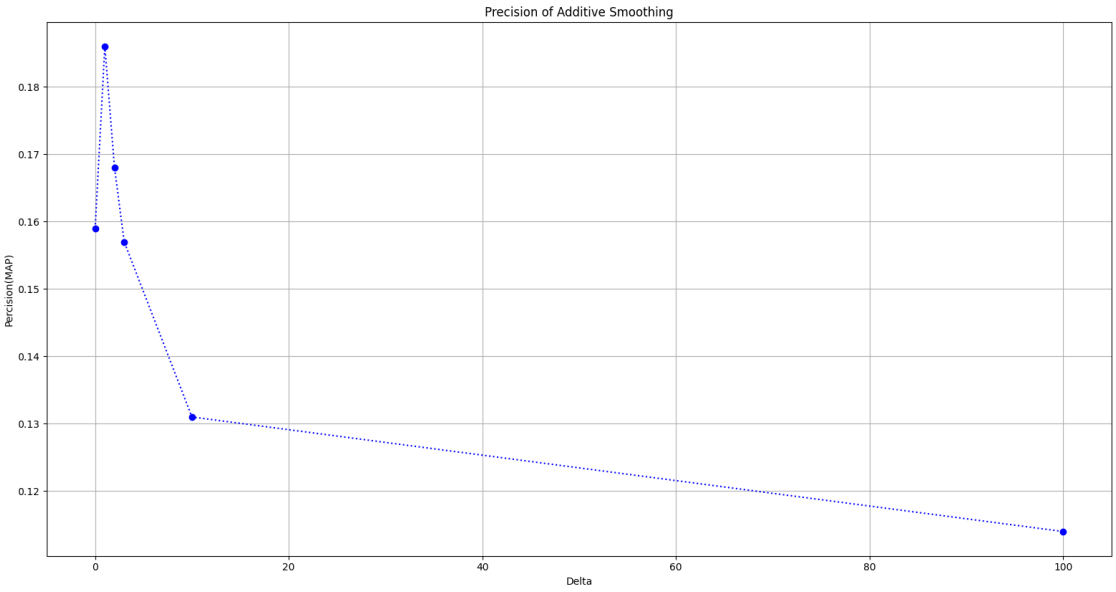
run-id	delta	map	P20	
/home/mamathew/Desktop/AdditiveTest01.txt	0	0.159	0.269	
/home/mamathew/Desktop/AdditiveTest02.txt	1	0.186	0.337	*
/home/mamathew/Desktop/AdditiveTest03.txt	2	0.168	0.297	
/home/mamathew/Desktop/AdditiveTest04.txt	3	0.157	0.282	
/home/mamathew/Desktop/AdditiveTest05.txt	10	0.131	0.222	
/home/mamathew/Desktop/AdditiveTest06.txt	100	0.114	0.187	
Sig-Test: randomized, threshold set to 0.050000				

جدول 3 جدول نتایج آزمایش مقادیر دلتا

همانطور که انتظار میرود نتایج بدست آمده از این روش به طرز قابل توجهی از دقت پایین تری نسبت به روش‌های گذشته برخوردار است. همچنین این در میان این مقادیر، روش {+1 Smoothing} از دقت بهتری برخوردار است.



شکل 5 نمودار P@20 برای مقادیر Additive Smoothing



شکل 6 نمودار Map برای مقادیر Additive Smoothing

سوال دوم:هموارسازی دو مرحله‌ای

مشاهده شد که هریک از روشهای هموارسازی مزایایی داشتند، هدف از این سوال بررسی هموارسازی دو مرحله‌ای میباشد که از ترکیب دو روش هموارسازی Dirichletو JMبدست میآید. در این تمرین قصد داریم به بررسی این روش هموارسازی بپردازیم و نتایج بدست آمده را با سوال قبل مقایسه کرده و تحلیل کنیم

معادله این تابع هموارساز به صورت زیر است:

$$P(w|d) = (1-\lambda)\frac{c(w_و d) + \mu p(W|C)}{|d| + \mu} + \lambda p(W|C)$$

مرحله اول شامل برآورد یک مدل زبان سند مستقل از پرس و جو است، در حالی که مرحله دوم شامل محاسبه احتمال پرس و جو بر اساس یک مدل زبان پرس و جو است،که بر اساس مدل زبان سند برآورد شده است. بنابراین، استراتژی دو مرحله ای به صراحت تأثیرات مختلف مجموعه پرس و جو و اسناد را بر تنظیمات بهینه پارامترهای بازیابی به تصویر می کشد.

در مرحله اول هموارسازی، مدل زبان سند با استفاده از دیریکله و مدل زبان مجموعه به عنوان مدل مرجع هموار می شود. در مرحله دوم، مدل زبان سند هموار شده، با یک مدل زبان پس‌زمینه[5] پرس و جو درون‌یابی[6] می‌شود.

روش هموارسازی دو مرحله ای پیشنهادی، گامی به سوی هدف تنظیم پارامترهای بازیابی خاص پایگاه داده و پرس و جو به طور کامل خودکار، بدون نیاز به آزمایش‌های متعدد خسته کننده است. اثربخشی و استحکام این

	mu	lambda	map	P20	num_ret	num_rel	num_rel_re
woStageTest01.txt	1700	0	0.254	0.370	44400.000	6100.000	3247.000
woStageTest02.txt	1700	0.01	0.254	0.370	44400.000	6100.000	3250.000
woStageTest03.txt	1700	0.05	0.253	0.370	44400.000	6100.000	3251.000
woStageTest04.txt	1700	0.1	0.253	0.369	44400.000	6100.000	3248.000
woStageTest05.txt	1700	0.2	0.253	0.369	44400.000	6100.000	3250.000
woStageTest06.txt	1700	0.3	0.252	0.368	44400.000	6100.000	3244.000
woStageTest07.txt	1700	0.4	0.250	0.369	44400.000	6100.000	3238.000
woStageTest08.txt	1700	0.5	0.248	0.366	44400.000	6100.000	3244.000
woStageTest09.txt	1700	0.6	0.246	0.366	44400.000	6100.000	3234.000
woStageTest10.txt	1700	0.7	0.243	0.366	44400.000	6100.000	3233.000
woStageTest11.txt	1700	0.8	0.236	0.357	44400.000	6100.000	3213.000
woStageTest12.txt	1700	0.9	0.223	0.332	44400.000	6100.000	3169.000
woStageTest13.txt	1700	0.95	0.208	0.312	44400.000	6100.000	3133.000
woStageTest14.txt	1700	0.99	0.188	0.274	44400.000	6100.000	3100.000

جدول 4 جدول نتایج آزمایش مقادیر μ برای هموارسازی دومرحله‌ای

[5] Background
[6] Interpolate

رویکرد، همراه با این واقعیت که هیچ تنظیم پارامتری درگیر نیست، آن را به عنوان یک روش پایه محکم برای ارزیابی مدل‌های بازیابی بسیار مفید می‌سازد.

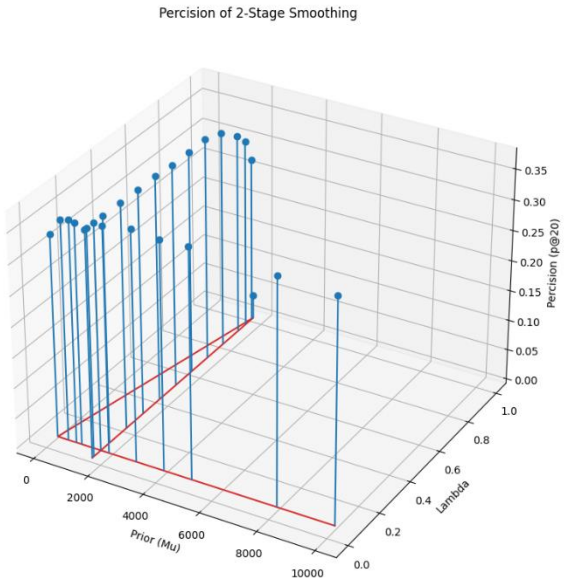
	lambda	mu	map		P20		num_ret	num_rel	num_rel_ret	
/TwoStageTest16.txt	0.05	100	0.233		0.332		44400.000	6100.000	3040.000	
/TwoStageTest17.txt	0.05	500	0.248	*	0.359	*	44400.000	6100.000	3149.000	*
/TwoStageTest18.txt	0.05	800	0.251	*	0.363	*	44400.000	6100.000	3189.000	*
/TwoStageTest19.txt	0.05	1000	0.252	*	0.361	*	44400.000	6100.000	3210.000	*
/TwoStageTest20.txt	0.05	2000	0.253	*	0.369	*	44400.000	6100.000	3250.000	*
/TwoStageTest21.txt	0.05	3000	0.250	*	0.376	*	44400.000	6100.000	3261.000	*
/TwoStageTest22.txt	0.05	4000	0.247	*	0.372	*	44400.000	6100.000	3249.000	*
/TwoStageTest23.txt	0.05	5000	0.245		0.374	*	44400.000	6100.000	3253.000	*
/TwoStageTest24.txt	0.05	8000	0.239		0.369	*	44400.000	6100.000	3228.000	*
/TwoStageTest25.txt	0.05	10000	0.235		0.365	*	44400.000	6100.000	3224.000	*

threshold set to 0.050000

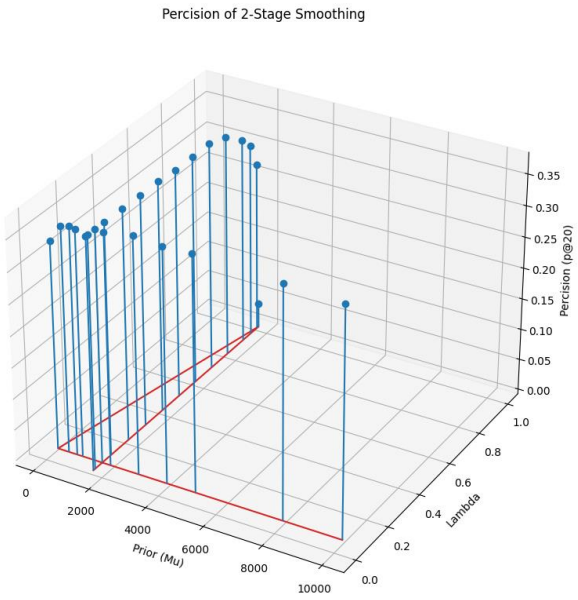
جدول 5 جدول نتایج آزمایش مقادیر λ برای هموارسازی دو مرحله ای

همانطور که در جداول بالا و نمودارهای بعدی مشخص است، ابتدا برای پیدا کردن مقدار بهینه λ از یک مقدار ثابت μ استفاده میکنیم و مقدار بهینه λ = 0.05 نتیجه میشود. در مرحله بعدی با در نظر گرفتن مقدار بهینه بدست آمده از آزمایش اول، مقدار λ را ثابت و مقدار μ را آزمایش میکنیم.

از نتایج آزمایش برمی‌آید که مقدار μ بهینه با مقدار بدست آمده در روش Dirichlet Prior برابری دارد ولی مقدار λ از مقدار پایین‌تری برخوردار است که نشان دهنده این است که برای بدست آمدن پاسخ بهتر نیاز روش دیریکله تاثیر بیشتری نسبت به احتمال مدل Collection باید داشته باشید.



شکل 7 نمودار مقادیر P@20 برای هموارسازی دومرحله‌ای



شکل 8 نمودار مقادیر Map برای هموارسازی دومرحله‌ای

سوال سوم: پیاده سازی تابع وزن دهی با استفاده از Pseudo Relevance Feedback

در این سوال قصد داریم به پیاده سازی تابع وزندهی با استفاده از Pseudo Relevance Feedbackبپردازیم.

به طور شهودی، همه روش‌های شبه بازخورد [۷]سعی می‌کنند اطلاعات مفیدی را از اسناد بازخورد بیاموزند. دو کامپوننت Mixture-Model و Background-Model به ما این امکان را می دهد که یک مدل زبان جزء یونیگرام[۸] (یعنی θT) را از اسناد بازخورد با فاکتورگیری کلمات با احتمال زیاد مطابق با مدل پس زمینه [۹]بدست بیاوریم.

$$\log p(D|\theta_T, \alpha_D) = \sum_{w \in V} c(w, D) \log(\alpha_D p(w|\theta_T) + (1-\alpha_D) p(w|\theta_B))$$

[7] Pseudo Feedback
[8] Unigram
[9] Background

یکی از کمبودهای این مدل مخلوط ساده این است که ضریب اختلاط α در تمام اسناد ثابت است، حتی اگر برخی اسناد بازخورد احتمالاً Noise بیشتری نسبت به سایرین دارند. برای مدل‌سازی مقادیر مختلف ارتباط در اسناد مختلف، باید به هر سند اجازه دهیم α_D متفاوتی داشته باشد. به طور طبیعی، ما انتظار داریم که یک سند مربوط

E-step:

$$p(Z_{w,D}=1) = \frac{\alpha_D^{(n)} P^{(n)}(w|\theta_T)}{\alpha_D^{(n)} P^{(n)}(w|\theta_T) + (1-\alpha_D^{(n)}) P(w|\Theta_B)}$$

M-step:

$$\alpha_D^{(n+1)} = \frac{\sum_{w\in V} p(Z_{w,D}=1)c(w,D)}{\sum_{w\in V} c(w,D)}$$

$$P^{(n+1)}(w|\theta_T) = \frac{\mu P(w|Q) + \sum_{D\in F} c(w,D)p(Z_{w,D}=1)}{\mu + \sum_{w'\in V}\sum_{D\in F} c(w',D)p(Z_{w',D}=1)}$$

دارای α_D بزرگتر از یک سند غیر مرتبط باشد. با این توصیف، میتوان از تخمین بیزی [10] برای به حداکثر رساندن احتمال پارامترها، در مقابل به حداکثر رساندن تابع Likelihood پارامترها استفاده کنیم.

یا بطور ساده تر میتوان نوشت:

$$p^{(n)}(z_i=1|w_i) = \frac{\lambda p(w_i|C)}{\lambda p(w_i|C) + (1-\lambda)p^{(n)}(w_i|\theta_F)}$$

Expectation-Step:
Augmenting data by guessing hidden variables

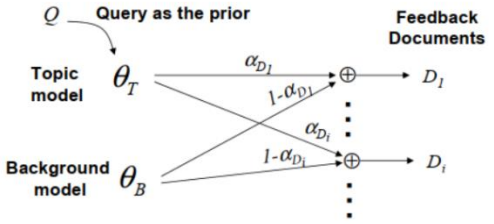
$$p^{(n+1)}(w_i|\theta_F) = \frac{c(w_i,F)(1-p^{(n)}(z_i=1|w_i))}{\sum_{w_j\in vocabulary} c(w_j,F)(1-p^{(n)}(z_j=1|w_j))}$$

Maximization-Step
With the "augmented data", estimate parameters using maximum likelihood

متغیر پنهان Z نشان می دهد که آیا عبارت w در سند D با استفاده از θT در مقابل θB ایجاد شده است یا خیر.

[10] Bayesian

از آنجایی که ما مدل پرس و جو اصلی p(w|Q) را قبلاً گنجانده ایم، مدل مبحث تخمینی θT می تواند مستقیماً به عنوان مدل پرس و جو به روز شده ما، بر اساس اسناد بازخورد گرفته شود.



برای پیاده‌سازی این تابع پس از محاسبات لازم، برای تکرار قدم‌های EM از یک Threshold برای بررسی همسو[11] شدن مقادیر استفاده شده است که مقدار آن برابر 0.00001 میباشد.

مرحله اول: رابطه بین تعداد سند بازخورد و معیارهای ارزیابی

دراین مرحله قصد داریم رابطه بین تعداد سندهای منتخب برای بازخورد به ازای مقادیر بزرگتر از ۱و معیار ارزیابی را نمایش دهید و به تحلیل نتایج بپردازیم، مقدار بهینه را شناسایی و گزارش کنیم.

	#docs	map	P20	num_ret	num_rel	num_rel_ret	
/mixture-model/doc_ranking01.txt	1	0.293	0.410	48000.000	5815.000	3378.000	
/mixture-model/doc_ranking02.txt	10	0.294	0.417	48000.000	5815.000	3448.000	
/mixture-model/doc_ranking03.txt	20	0.294	0.400	48000.000	5815.000	3521.000	*
/mixture-model/doc_ranking04.txt	30	0.292	0.401	48000.000	5815.000	3545.000	*
/mixture-model/doc_ranking05.txt	50	0.287	0.399	48000.000	5815.000	3611.000	*
/mixture-model/doc_ranking06.txt	100	0.280	0.395	48000.000	5815.000	3623.000	*
/mixture-model/doc_ranking11.txt	11	0.295	0.413	48000.000	5815.000	3451.000	
/mixture-model/doc_ranking12.txt	12	0.295	0.411	48000.000	5815.000	3444.000	
/mixture-model/doc_ranking13.txt	13	0.294	0.403	48000.000	5815.000	3444.000	
/mixture-model/doc_ranking14.txt	14	0.294	0.402	48000.000	5815.000	3454.000	
/mixture-model/doc_ranking15.txt	15	0.296	0.408	48000.000	5815.000	3453.000	
/mixture-model/doc_ranking16.txt	16	0.298	0.408	48000.000	5815.000	3496.000	*
/mixture-model/doc_ranking17.txt	17	0.298	0.410	48000.000	5815.000	3514.000	*
/mixture-model/doc_ranking18.txt	18	0.296	0.407	48000.000	5815.000	3508.000	*
/mixture-model/doc_ranking19.txt	19	0.294	0.402	48000.000	5815.000	3511.000	*
threshold set to 0.050000							

جدول 6 جدول آزمایش مقادیر تعداد سند بازخورد

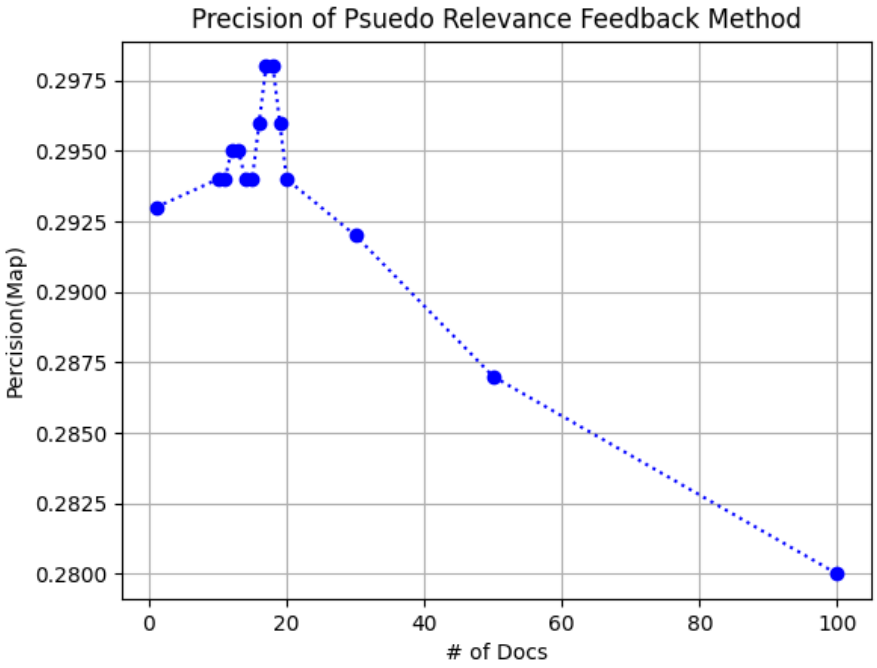
برای این آزمایش تعداد ترم‌های منتخب از اسناد را ثابت فرض کرده و روی تعداد اسناد آزمایش را انجام میدهیم.

همانطور که از جدول بالا و نمودارهای بعد نتیجه میشود، به صورت کلی با افزایش تعداد سندهای منتخب برای بررسی بازخورد، دقت جستجو پایین می‌آید و مقدار بهینه برای تعداد اسناد بازخورد برابر 17 است. این امر میتواند ناشی از نسبت دادن وزن بالا به ترم‌های نامرتبط بیشتر در تعداد اسناد منتخب بیشتر باشد که باعث بازگرداندن

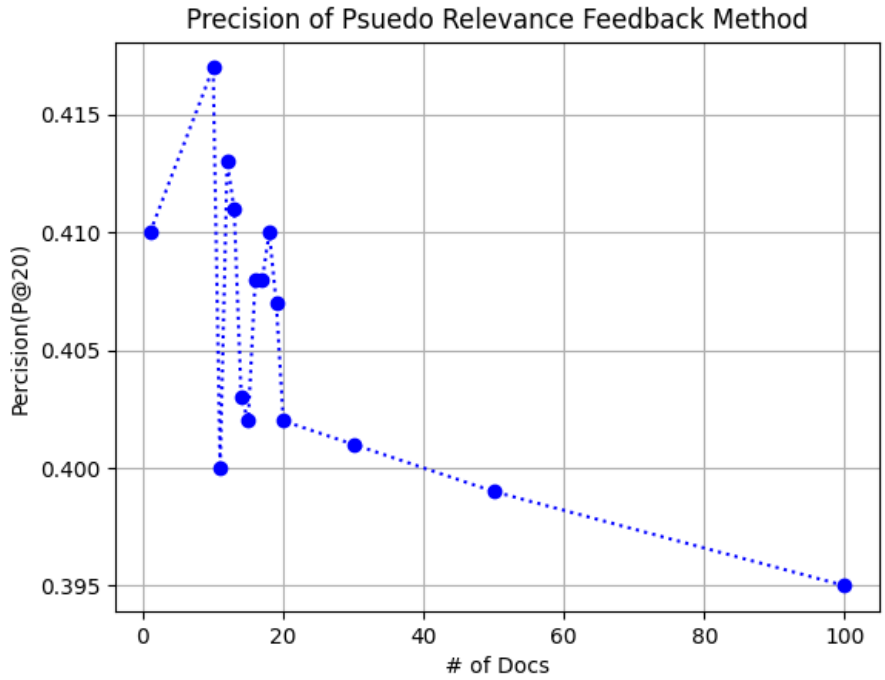
[11] Convergence

سندهای نامرتبط بیشتر نسبت به مرتبط میشود. از این رو در تعداد سندهای منتخب بالاتر، دقت در بازگردانی اسناد کاهش پیدا میکند.

به علاوه میتوان تاثیر به سزای استفاده از بازخورد در نتایج نسبت به روش‌های smoothing را مشاهده کرد که نتایج بسیار بهتر هستند زیرا با توجه به پرس و جو وزن‌دهی و جستجو میشوند.



شکل 10 نمودار Map بر اساس تعداد سند بازخورد



شکل 9 نمودار P@20 بر اساس تعداد سند بازخورد

مرحله دوم: تاثیر تعداد ترم‌های استخراج شده بر اساس تعداد سند بازخورد

در این مرحله تاثیر افزایش تعداد ترم‌های منتخب از اسناد بازخورد را بر روی نتایج بررسی میکنیم.

	#doc	#term	map	P20	num_ret	num_rel	num_rel_ret
/mixture-model/doc_ranking01.txt	19	50	0.298	0.410	48000.000	5815.000	3514.000
/mixture-model/doc_ranking02.txt	19	150	0.298	0.410	48000.000	5815.000	3514.000
/mixture-model/doc_ranking03.txt	19	500	0.298	0.410	48000.000	5815.000	3514.000
/mixture-model/doc_ranking04.txt	5	50	0.296	0.408	48000.000	5815.000	3453.000
/mixture-model/doc_ranking05.txt	5	150	0.296	0.408	48000.000	5815.000	3453.000
/mixture-model/doc_ranking06.txt	5	500	0.296	0.408	48000.000	5815.000	3453.000

threshold set to 0.050000

با توجه به نتایج آزمایشات بالا به وضوح میتوان مشاهده کرد که در آزمایش ما تعداد ترم‌های منتخب تاثیری در نتیجه نهایی نداشته و فقط تعداد اسناد تاثیر گذار هستند. فرض برای این موضوع این است که همسو شدن مقادیر وزن ترم‌ها باعث بی تاثیر شدن تعداد آنها میشود و هربار تنها ترم‌های مرتبط وزن میگیرند و باقی ترم‌های اضافی تاثیری ندارند.