به نام خدا

دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر

درس بازیابی اطلاعات

موضوع: گزارش پروژه جستجوی محصول (Product Search)

استاد درس: دکتر ممتازی

دانشجو: محمد قربانی ۹۷۱۳۱۰۹۹

نیم‌سال اول ۹۹-۱۳۹۸

فهرست مطالب

۱. مقدمه

در این پروژه می‌خواهیم بازیابی کالای مرتبط انجام دهیم بدین صورت که ما یک کوئری که از تعدادی توکن تشکیل شده‌است را در اختیار داریم و می‌خواهیم کالاهای مرتبط با این کوئری را بدست بیاوریم. این موضوع کاربرد مشهودی در سیستم‌های جستجو و پیشنهادکننده[۱] دارد. یافتن کالای مرتبط روش‌های مختلفی دارد که در این پروژه از روش مبتنی بر گراف استفاده شده‌است. روشی که در این پروژه استفاده می‌شود BiNE نام دارد که برای گراف‌های دو بخشی استفاده می‌شود. در بخش دوم این گزارش ابتدا کمی به تعریف مسئله می‌پردازیم. در بخش سوم به بیان توضیحات در رابطه با چگونگی انجام پروژه پرداخته‌ایم. در بخش چهارم نتایجی که بدست آورده‌ایم را بیان کرده‌ایم. در بخش پنجم ایده‌های که به نظرمان می‌توان برای بهبود این جستجوگر استفاده شود را بیان کرده‌ایم و در بخش آخر منابع استفاده‌شده آورده‌ایم.

۲. تعریف مسئله: جستجوی کالای مرتبط

برای یافتن آیتم مرتبط یکی از راه‌ها استفاده از رویکردهای مبتنی بر بردار و ارزیابی شباهت از طریق معیارهایی همچون Cosine، Jaccard و ... بود. اما چطور می‌توان بردار یک آیتم را بدست آورد؟ در کلاس درس مشاهده شد که دسته‌ای از روش‌ها به نام Content-Based وجود دارد. در این روش باید ویژگی‌های محصول بدست می‌آمد که این طاقت فرسا بود. همچنین این روش تنها برای دامنه‌های تک منظور مناسب بود. در این پروژه ما می‌خواهیم به عنوان یک جایگزین از رویکردهای گرافی برای بدست آوردن بردار برای کلمات و محصولات استفاده کنیم.

۳. مراحل انجام پروژه

۳_۱. بررسی مجموعه داده

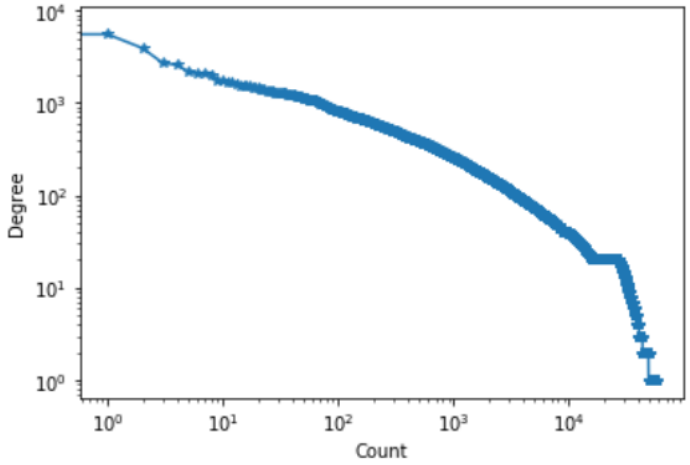
جدول train-queries دارای ۹۲۳۱۲۸ رکورد می‌باشد که از این تعداد فقط ۵۱۸۸۸ رکود دارای مقدار برای ستون searchstring.tokens می‌باشد و ما فقط این رکوردها را در نظر می‌گیریم. از این رکوردها تعداد ۱۳۸۲۶۰ توکن استخراج می‌شود که تعداد ۱۹۰۷۵ توکن متمایز در آن دیده می‌شود. همچنین تعداد ۹۹۸۲۱۴ آیتم وجود دارد که ۳۸۰۸۴ عدد آن متمایز می‌باشد.

تعداد کل رکوردهای جدول اصلی	**۹۲۳۱۲۸**
تعداد کل رکوردهای مفید	**۵۱۸۸۸**
تعداد کل توکن‌ها	**۱۳۸۲۶۰**

[۱] Recommender Systems

تعداد توکن‌های متمایز	۱۹۰۷۵
تعداد کل آیتم‌ها	۹۹۸۲۱۴
تعداد آیتم‌های متمایز	۳۸۰۸۴
تعداد یال‌ها	۱۰۸۱۲۶۸

همچنین ما با استفاده از کتابخانه network نمودار توزیع درجه مجموعه‌ی داده‌ی مفید را بدست آورده‌ایم که به صورت زیر می‌باشد:



تصویر ۱ نمودار توزیع درجه گراف ساخته شده

۲_۳. انجام پیش‌پردازش

یکی از مشاهدات ما این بود که شناسه‌ی توکن‌ها و آیتم‌ها دارای اشتراک هستند به همین دلیل در ابتدا لازم بود که این اشتراک از بین برود چرا که ممکن است برای ایجاد بردار برای گره‌ها مشکلاتی را ایجاد کند. برای حل این مشکل ایده‌ای که به ذهنمان رسید این بود که به شناسه‌ی توکن‌ها ۱ میلیون و به شناسه‌ی آیتم‌ها ۲ میلیون اضافه کنیم. بدین صورت این مشکل مرتفع شد.

۳_۳. آماده سازی داده برای الگوریتم BiNE

الگوریتم BiNE که در [1] معرفی شده‌است روشی برای ساخت بازنمایی برای گره‌ها برای گراف‌های دو بخشی می‌باشد. ورودی این الگوریتم دو فایل آموزش و تست می‌باشد. هر سطر این فایل نشان دهند یک یال به صورت (مبدا مقصد وزن) می‌باشد.

۳_۳. خواندن بردارهای بازنمایی بدست آمده

الگوریتم BiNE بردار بازنمایی ایجاد شده را بر روی فایل ذخیره می‌کند. ما این بردارها را از روی فایل‌ها خوانده و تبدیل به دیتافریم می‌کنیم که بتوانیم براحتی به آن‌ها دسترسی داشته باشیم.

۴. ارزیابی

۴-۱. بازنمایی با الگوریتم BiNE

منظور از **کمینه وزن یال**: با توجه به این که گرافی که از این مجموعه حاصل می‌شود بسیار بزرگ می‌باشد تصمیم گرفتیم که یال‌هایی را حذف کنیم تا حجم گراف کمتر شود تا بتوانیم در زمان قابل تحمل با استفاده از BiNE به بردارهای مورد نظر خود برسیم. یال‌هایی را برای حذف انتخاب کردیم که تعداد کمی تکرار شده باشند. در واقع منطق ما این است که اگر توکن۱ با آیتم۱ در این داده عظیم فقط یک بار (یا حتی عددهای بالاتر) اتفاق افتاده باشد می‌توان آن را به صورت نویز در نظر گرفت و حذف کرد.

منظور از **تعداد نمونه برداری شده برای ارزیابی**: زمانبرترین قسمت این پروژه بخش ارزیابی می‌باشد. در این بخش باید میلیون‌ها بردار در حالت عادی با یکدیگر مقایسه شوند. ما برای اینکه بتوانیم این قسمت را انجام دهیم تصمیم گرفتیم که ارزیابی را بر روی همه‌ی داده‌ها انجام ندهیم و فقط درصدی از داده را برای این کار در نظر بگیریم. برای اینکه این ارزیابی عادلانه باشد این کار به صورت تصادفی انجام می‌شود.

همچنین این نکته را نیز می‌توان گفت که به جز جدول اول، در سایر جداول تعداد رکوردها برای ارزیابی برای روش دوم نصف تعداد رکورد ارزیابی در روش اول می‌باشد. دلیل این کار بسیار زمانگیر بودن روش دوم می‌باشد.

ما برای اینکه روش گفته شده رو با دقت بهتری ارزیابی کنیم پارامترها را به صورت متفاوت در نظر گرفتیم. در واقع ما ارزیابی خود را بر روی مقادیر مختلف برای **کمینه وزن یال** و **تعداد بعد** انجام داده‌ایم.

ارزیابی روش اول: در این روش یک بردار میانگین برای کوئری بدست می‌آید و سپس مشابهت تمامی آیتم‌ها با آن بدست می‌آید و در آخر لیست آیتم‌ها براساس امتیاز مرتب می‌شود.

ارزیابی روش دوم: در این روش امتیاز هر آیتم می‌شود میانگین امتیاز آن آیتم با هر توکن کوئری. این ارزیابی بسیار زمانگیر می‌باشد.

کمینه وزن یال	**۳**
تعداد بعد در نظر گرفته شده برای بردارها	**۱۲۸**

روش اول		روش دوم	
درصد نمونه برداری شده برای ارزیابی	۰/۰۰۱ (۵۲ رکورد)	درصد نمونه برداری شده برای ارزیابی	۰/۰۰۱ (۵۲ رکورد)
زمان اجرا	۴۰۰ ثانیه	زمان اجرا	۲۲۸۶ ثانیه
داده آموزش		داده آموزش	
MAP	۰/۰۱۲۳۹۶۵۷۱۱۳۳۵۹۵۳۵۴	MAP	۰/۰۳۳۲۱۳۳۵۱۴۲۸۶۸۴۶۶
MRR	۰/۰۸۳۲۲۹۳۵۲۳۴۶۹۹۹۴۱	MRR	۰/۱۴۹۲۸۳۰۶۳۴۲۷۸۰۰۲۶
NDCG	۰/۰۳۶۷۴۴۴۴۹۵۱۸۳۷۴۱۹	NDCG	۰/۰۶۶۲۰۳۹۵۴۹۰۱۹۷۹۸۵
داده تست		داده تست	
MAP	۰/۰۲۱۱۱۶۵۰۱۲۷۰۰۱۰۰۴	MAP	۰/۰۱۷۵۷۷۰۰۹۸۷۰۹۳۴۰۲
MRR	۰/۱۸۴۰۲۷۷۷۷۷۷۷۷۷۸	MRR	۰/۱۰۲۴۱۵۹۶۶۳۸۶۵۵۴۶۲
NDCG	۰/۰۶۰۲۸۴۹۴۵۰۸۳۷۱۸۸۴۴	NDCG	۰/۰۵۳۹۷۲۳۱۱۹۶۸۶۱۶۸۹

کمینه وزن یال		۵	
تعداد بعد در نظر گرفته شده برای بردارها		۱۲۸	
زمان اجرای الگوریتم BiNE		۴۶۱ ثانیه	
روش اول		روش دوم	
درصد نمونه برداری شده برای ارزیابی	۰/۰۰۱ (۵۲ رکورد)	درصد نمونه برداری شده برای ارزیابی	۰/۰۰۰۵ (۲۶ رکورد)
زمان اجرا	۱۲۵ ثانیه	زمان اجرا	۱۰۳۱ ثانیه
داده آموزش		داده آموزش	
MAP	۰/۰۱۹۵۷۶۷۵۰۸۱۵۷۸۵۹	MAP	۰/۰۲۲۵۵۸۴۷۶۰۴۱۹۳۱۹۲۴
MRR	۰/۱۲۰۶۵۵۲۷۰۶۵۵۲۷۰۶۷	MRR	۰/۱۴۲۵۶۵۳۵۹۴۷۷۱۲۴۱۸
NDCG	۰/۰۴۸۰۱۵۴۱۰۴۴۵۶۰۴۷۲	NDCG	۰/۰۵۹۱۷۲۲۳۳۸۳۹۷۸۷۷۶۵
داده تست		داده تست	
MAP	۰/۰۱۲۴۶۷۹۰۳۸۲۸۱۹۷۹۴۷	MAP	۰/۰۰۵۲۹۷۶۱۹۰۴۷۶۱۹۰۴۷۵
MRR	۰/۱۰۲۶۷۸۵۷۱۴۲۸۵۷۱۴۲	MRR	۰/۰۳۶۶۶۶۶۶۶۶۶۶۶۶۷

۰٫۰۳۳۹۷۰۹۳۷۹۶۷۹۳۹۸۲	NDCG	۰٫۰۲۲۷۷۶۷۹۸۲۰۲۳۷۰۲۵۷	NDCG

کمینه وزن یال		**۵**	
تعداد بعد در نظر گرفته شده برای بردارها		**۶۴**	
زمان اجرای الگوریتم BiNE		**۳۳۹ ثانیه**	
روش اول		**روش دوم**	
درصد نمونه برداری شده برای ارزیابی	**۰٫۰۰۱ (۵۲ رکورد)**	**درصد نمونه برداری شده برای ارزیابی**	**۰٫۰۰۰۵ (۲۶ رکورد)**
زمان اجرا	**۶۱ ثانیه**	**زمان اجرا**	**۷۷۲ ثانیه**
داده آموزش		**داده آموزش**	
MAP	۰٫۰۱۹۱۳۴۴۳۰۹۶۰۳۲۶۴۸۶	MAP	۰٫۰۲۵۵۰۷۱۳۴۳۷۲۴۳۶۶۸۶
MRR	۰٫۰۹۳۰۲۹۴۴۸۶۲۱۵۵۳۸۹	MRR	۰٫۰۹۴۴۴۴۴۴۴۴۴۴۴۴۴
NDCG	۰٫۰۴۹۹۶۲۱۷۸۴۸۴۰۳۵۷	NDCG	۰٫۰۶۷۰۳۱۶۱۵۳۳۵۸۵۳۴
داده تست		**داده تست**	
MAP	۰٫۰۱۹۷۸۶۷۰۶۳۴۹۲۰۶۳۴۶	MAP	۰٫۰۰۰۳۷۰۳۷۰۳۷۰۳۷۰۳۷۰۳۵
MRR	۰٫۱۸۶۵۰۷۹۳۶۵۰۷۹۳۶۵	MRR	۰٫۰۰۷۴۰۷۴۰۷۴۰۷۴۰۷۴۰۸
NDCG	۰٫۰۵۱۷۴۶۶۷۶۲۷۳۸۸۸۲۹	NDCG	۰٫۰۰۳۹۴۵۵۵۶۶۵۶۴۹۰۴۸۱

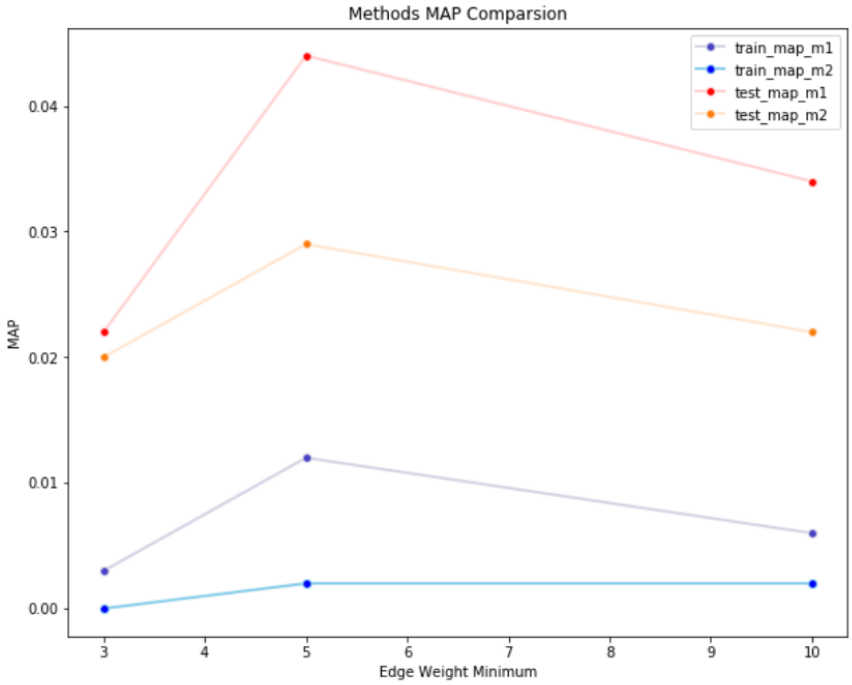
کمینه وزن یال		**۱۰**	
تعداد بعد در نظر گرفته شده برای بردارها		**۱۲۸**	
زمان اجرای الگوریتم BiNE		**۲۲۳ ثانیه**	
روش اول		**روش دوم**	
درصد نمونه برداری شده برای ارزیابی	**۰٫۰۰۱ (۵۲ رکورد)**	**درصد نمونه برداری شده برای ارزیابی**	**۰٫۰۰۰۵ (۲۶ رکورد)**
زمان اجرا	**۱۲۸ ثانیه**	**زمان اجرا**	**۱۲۹۳ ثانیه**
داده آموزش		**داده آموزش**	
MAP	۰٫۰۱۲۶۹۱۴۴۳۶۰۷۸۲۳۹۶	MAP	۰٫۰۱۶۵۲۶۳۱۵۷۸۹۴۷۳۶۸۴

MRR	۰٫۲۰۶۶۶۶۶۶۶۶۶۶۶۶۷	MRR	۰٫۱۲۳۰۱۵۸۷۳۰۱۵۸۷۳۰۱
NDCG	۰٫۰۴۶۱۲۶۲۱۶۷۲۱۵۸۰۱۶	NDCG	۰٫۰۳۶۲۸۴۰۴۸۲۹۴۸۹۳۲۴
داده تست		**داده تست**	
MAP	۰٫۰۳۶۳۳۹۴۸۸۵۲۸۴۸۳۷۴	MAP	۰٫۰۳۱۰۰۸۵۰۳۸۸۰۰۲۰۹۱۴
MRR	۰٫۲۷۷۵۱۱۹۶۱۷۲۲۴۸۸	MRR	۰٫۱۶۷۴۶۸۸۰۵۷۰۴۰۹۹۸۳
NDCG	۰٫۰۸۹۶۸۴۸۷۵۱۳۱۶۷۶۵۷	NDCG	۰٫۰۹۲۸۷۰۴۴۴۴۵۸۵۴۴۴۸

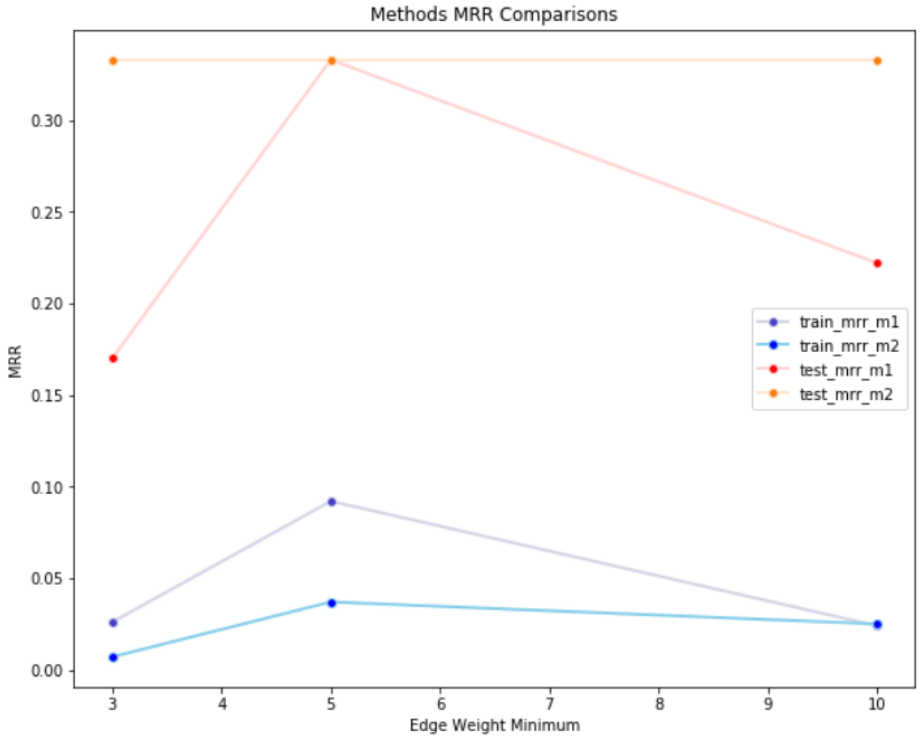
۱۰		**کمینه وزن یال**	
۶۴		**تعداد بعد در نظر گرفته شده برای بردارها**	
۱۶۰ ثانیه		**زمان اجرای الگوریتم BiNE**	
روش دوم		**روش اول**	
درصد نمونه برداری شده برای ارزیابی	**۰٫۰۰۰۵ (۲۶ رکورد)**	**درصد نمونه برداری شده برای ارزیابی**	**۰٫۰۰۱ (۵۲ رکورد)**
زمان اجرا	**۷۷۲ ثانیه**	**زمان اجرا**	**۶۱ ثانیه**
داده آموزش		**داده آموزش**	
MAP	۰٫۰۲۵۵۰۷۱۳۴۳۷۲۴۳۶۶۸۶	MAP	۰٫۰۰۵۱۶۹۲۵۷۲۴۵۳۸۱۸۱۲
MRR	۰٫۰۹۴۴۴۴۴۴۴۴۴۴۴۴۴	MRR	۰٫۰۵۱۵۸۷۳۰۱۵۸۷۳۰۱۵۸۴
NDCG	۰٫۰۶۷۰۳۱۶۱۵۳۳۵۸۵۳۴	NDCG	۰٫۰۱۷۹۳۴۹۱۳۱۴۳۹۸۸۴۶
داده تست		**داده تست**	
MAP	۰٫۰۰۰۳۷۰۳۷۰۳۷۰۳۷۰۳۷۰۳۵	MAP	۰٫۰۰۶۳۹۶۳۰۹۷۴۲۲۲۰۴۷۴۵
MRR	۰٫۰۰۷۴۰۷۴۰۷۴۰۷۴۰۷۴۰۸	MRR	۰٫۰۶۲۸۸۵۸۰۲۴۶۹۱۳۵۸
NDCG	۰٫۰۰۳۹۴۵۵۵۶۶۵۶۴۹۰۴۸۱	NDCG	۰٫۰۲۹۴۹۹۷۲۸۸۹۰۳۷۳۷۷

ارزیابی ما در حالت‌های مختلف نشان می‌دهد که MAP در حدود ۵ درصد و MRR در حدود ۱۰ درصد می‌باشد. همچنین می‌توان گفت که نصف شدن تعداد بعد بردار باعث نصف شدن زمان اجرای ارزیابی‌ها شده است. همچنین می‌توان گفت که زمان اجرای روش دوم در حدود ۱۰ برابر روش اول می‌باشد.

برای مقایسه‌ی بهتر این دو روش مختلف می‌توان از نمودارهای زیر استفاده کرد:

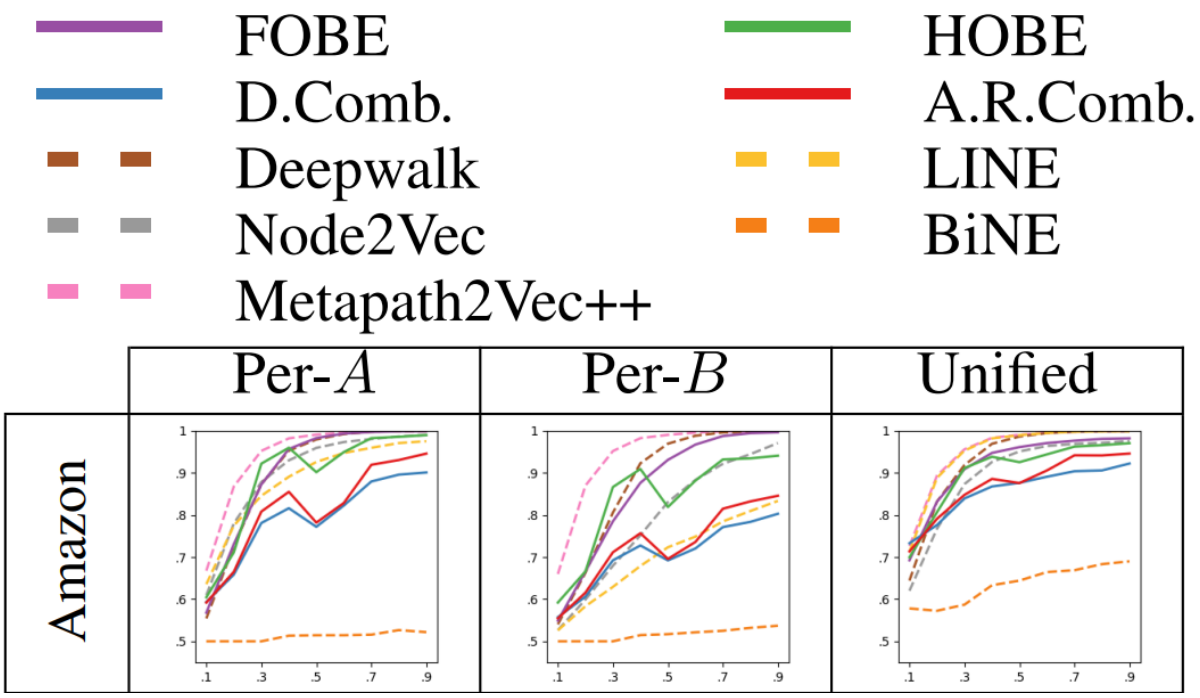


تصویر ۲ مقایسه MAP دو روش



تصویر ۳ مقایسه MRR دو روش

ارزیابی ما این را نشان داد که روش به کاربرده شده برای استخراج آیتم مرتبط بسیار ضعیف عمل می‌کند. البته این نتیجه‌گیری توسط گروه‌های دیگر نیز شده است. در [2] مقایسه‌ای بین الگوریتم‌های بازنمایی صورت گرفته است که نتایج آن نشان داده که الگوریتم BiNE تقریبا بدترین عملکرد را در بین بقیه داشته است. بخشی از نتایج این مقاله را در تصویر می‌توان مشاهده نمود.



تصویر ۴ مقایسه الگوریتم BiNE با سایر الگوریتم‌ها

۲-۴. بازنمایی با الگوریتم FOBE

در مقاله‌ی [2] ضمن مقایسه‌الگوریتم‌های مختلف دو الگوریتم FOBE و HOBE برای بازنمایی گراف دو بخشی پیشنهاد شده است. ما در این پروژه یک ارزیابی غیر جامع با استفاده از این الگوریتم انجام داده‌ایم. ما ۷ هزار سطر اول جدول اصلی را برای این کار در نظر گرفتیم و آموزش و ارزیابی را بر روی این سطرها انجام دادیم. بعد از ایجاد بازنمایی با استفاده از الگوریتم FOBE، ارزیابی را با استفاده از روش اول و آن هم فقط بر روی داده تست انجام دادیم که نتایج آن به صورت زیر بدست آمد.

روش	MAP	MRR	NDCG
روش اول	**۰٫۰۱۴۹۶۹۲۹۸۲۴۵۶۱۴۰۳۲**	**۰٫۰۸۶۶۶۶۶۶۶۶۶۶۶۶۷**	**۰٫۰۵۵۰۱۹۶۵۳۸۰۹۲۹۰۴۱۴**

ما از مقایسه‌ی این دو الگوریتم خودداری می‌کنیم چرا که نیاز به بررسی جامع‌تر می‌باشد ولی ظن ما این است که این الگوریتم نیز دارای بهبود چندانی نسبت به الگوریتم BiNE نمی‌باشد.

۵. پیشنهادات

۵-۱. بهبود دقت

در طی انجام این پروژه به چالش‌هایی برخورد کردیم که برخی از آن‌ها بسیار مهم و ارائه‌ی رویکردهای مناسب لازم و ضروری می‌باشد. یکی از این چالش‌ها **دقت پایین جستوی** ما می‌باشد. برای افزایش دقت پیشنهاد ما این است که تعداد دسته‌های اصلی بدست آید. در بررسی ما بر روی این مجموعه متوجه شدیم که ۱۲۱۶ دسته‌ی محصولات وجود دارد. البته این تعداد دسته‌های اصلی نمی‌باشد. اگر بتوانیم دسته‌های اصلی را بدست بیاوریم می‌توانیم بردار محصولاتی که در یک دسته قرار می‌گیرند را شبیه‌تر بهم کنیم.

در الگوریتم BiNE یک Random Walk وجود دارد. با دانستن این که دو محصول در یک دسته قرار می‌گیرند می‌توانیم احتمال یال بین این دو محصول را افزایش دهیم. بدین صورت هر Random Walkایی که صورت می‌گیرد احتمال اینکه آیتم‌های با دسته‌ی یکسان که پشت سرهم بیایند بیشتر می‌شود. ما فکر می‌کنیم که این کار باعث بهبود دقت می‌شود. البته به دلیل کمبود زمان نتوانستیم این ایده را پیاده سازی کنیم.

۵-۲. بهبود سرعت

یک مشکل غیر قابل تحمل دیگری که وجود داشت زمان‌گیر بودن ارزیابی بود. این موضوع در روش دوم به طور محسوسی به چشم می‌آید. پیشنهاد ما برای حل این مشکل این است که یک کلاسترینگ با تعداد خوشه‌ای برابر با تعداد دسته‌های اصلی داشته باشیم. حال برای بدست آوردن آیتم مرتبط به جای اینکه با تمامی آیتم‌ها مقایسه صورت بگیرد ابتدا با مراکز خوشه‌های صورت بگیرد. سپس مرتبط‌ترین خوشه‌ها بدست آید و حال با آیتم‌های در این خوشه‌ها مقایسه صورت بگیرد. ما فکر می‌کنیم که بکارگیری این روش تاثیر چشم‌گیری بر روی افزایش سرعت جستجو داشته باشد.

۶. منابع

[1] M. Gao, L. Chen, X. He, and A. Zhou, “BiNE,” pp. 715–724, 2018.

[2] J. Sybrandt and I. Safro, “FOBE and HOBE: First- and High-Order Bipartite Embeddings,” pp. 1–10, 2019.