به نام خدا

بازیابی پیشرفته‌ی اطلاعات

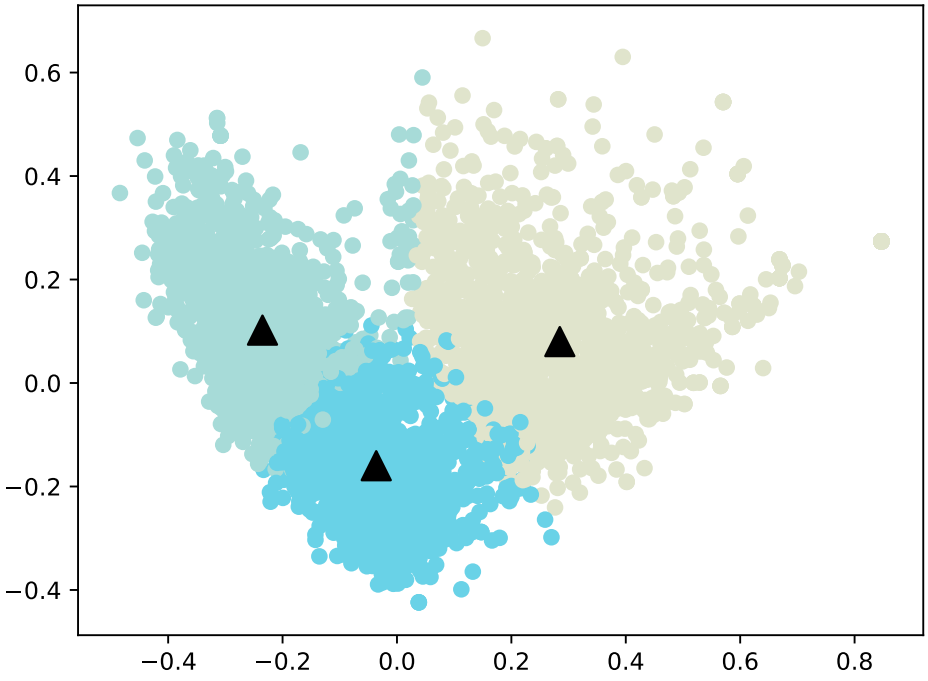
گزارش فاز سوم

امین رخشا ۹۵۱۰۹۳۱۵
مهبد مجید ۹۵۱۰۹۳۷۲

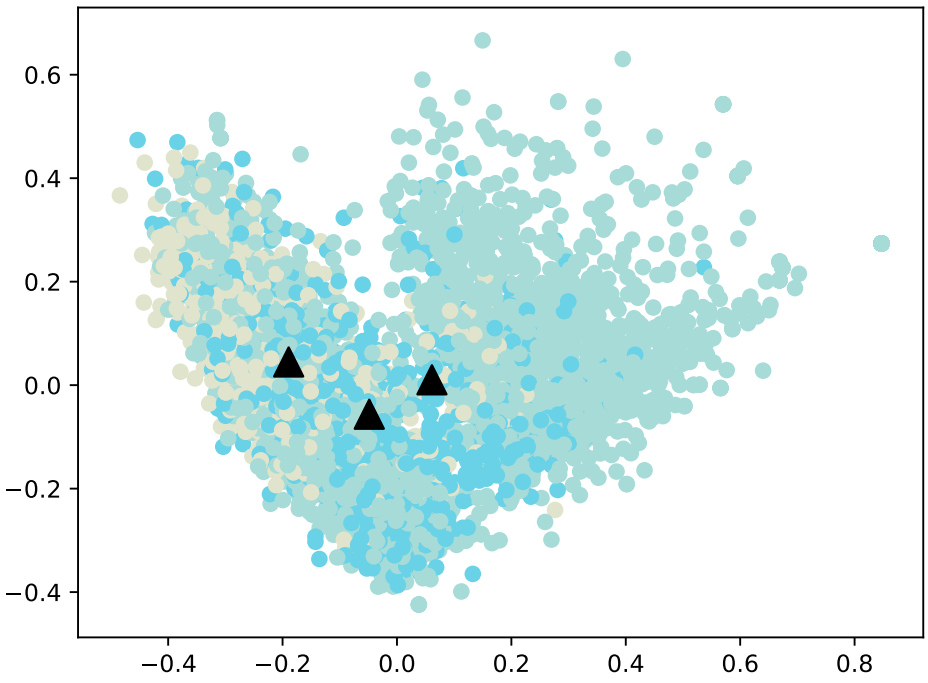
۲۲ دی ۱۳۹۸

۱ بخش اول ـ خوشه‌بندی

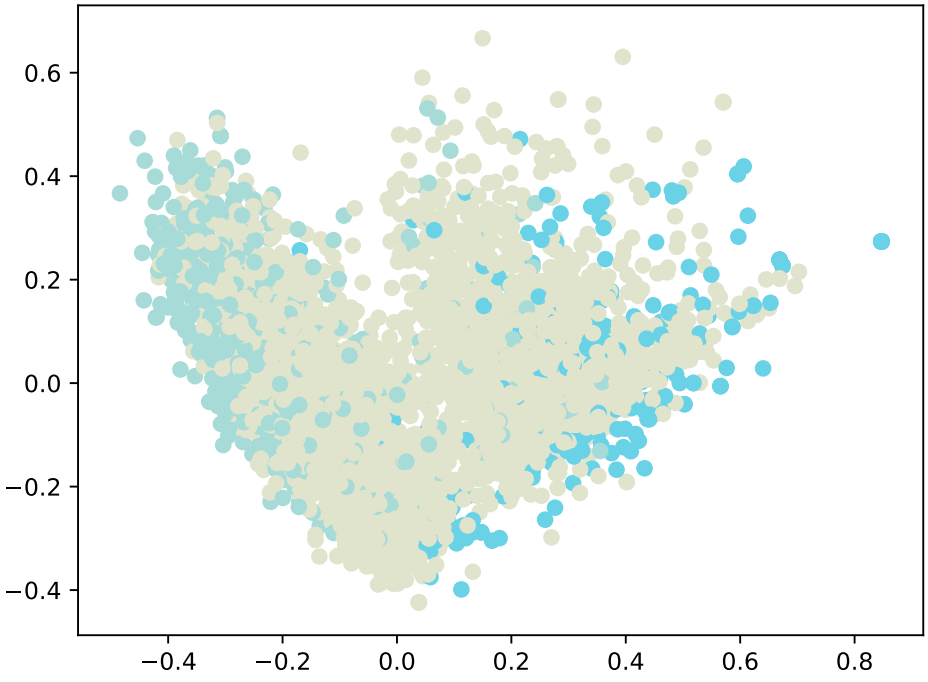
در این بخش ابتدا داده‌ها را به ۲ صورت tf-idf و doc2vec به ماتریس‌های ۱۰۰ × ۵۰۰۰ word × tweet تبدیل می‌کنیم. سپس از ماتریس حاصل را در روش‌های کلاسترینگ گفته‌شده استفاده می‌کنیم. سپس برای تصویرسازی از pca استفاده می‌کنیم که به ما ۲ مولفه‌ای که بیشترین تمایز میان سطرهای ماتریس را می‌دهند می‌دهد. هم‌چنین برای روش hierarchical دندروگرام هم می‌کشیم که نشان می‌دهد چگونه داده‌هایمان در این روش کلاسترشده‌اند. با توجه به شکل‌های مشاهده‌شده در صفحه‌ی دو مولفه‌ی اول pca، به‌نظر می‌آید که الگوریتم KMeans بهتر از باقی الگوریتم‌ها عمل‌کرده است زیرا به‌خوبی در این صفحه رنگ‌های متفاوت از یک‌دیگر جدا شده‌اند. هم‌چنین به‌نظر به طور کلی word2vec بهتر از tf-idf عمل‌کرده‌است. هم‌چنین طبق مشاهدات تعداد ۳ کلاستر به‌نظر مناسب‌است. یکی از خوبی‌های دندروگرام هم این است که نیازی نیست که تعداد کلاسترها را در آن مشخص‌کنیم.



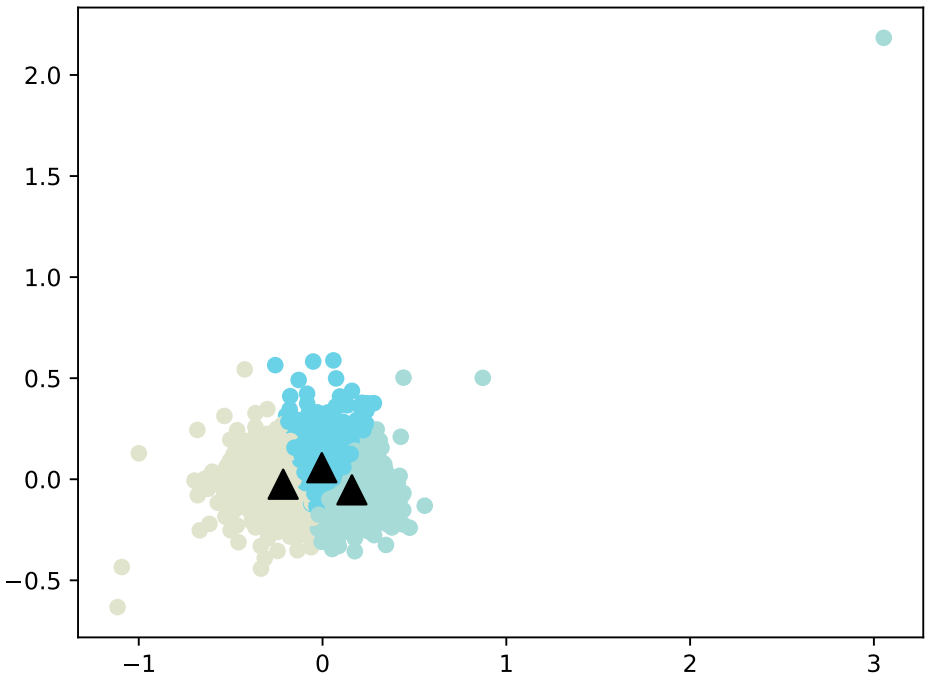
شکل ۱: tf-idf:KMeans:First 2 components PCA:3



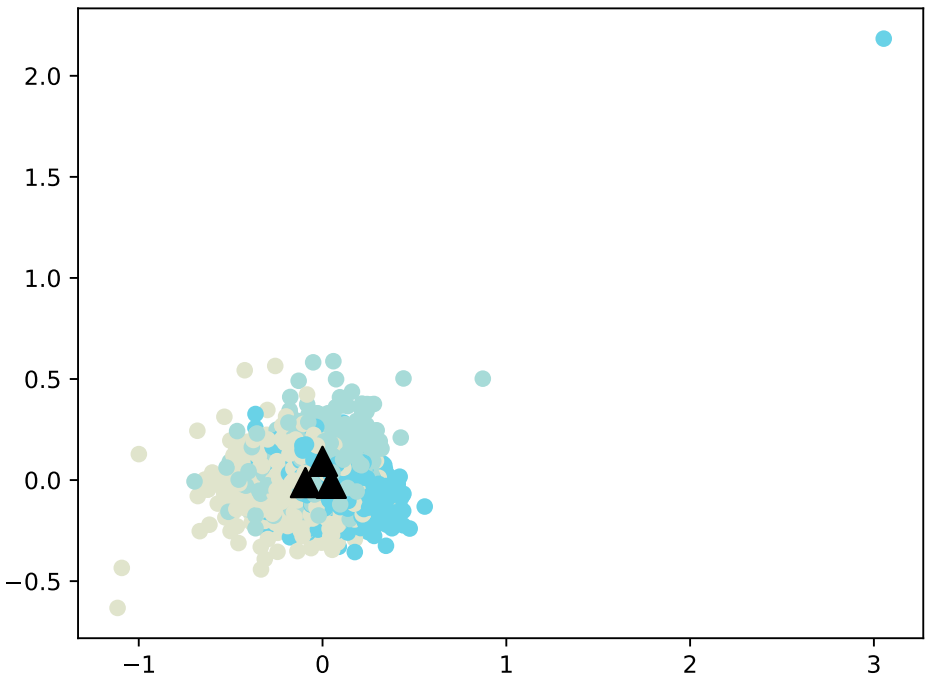
شکل ۲: tf-idf:GMM:First 2 components PCA:3



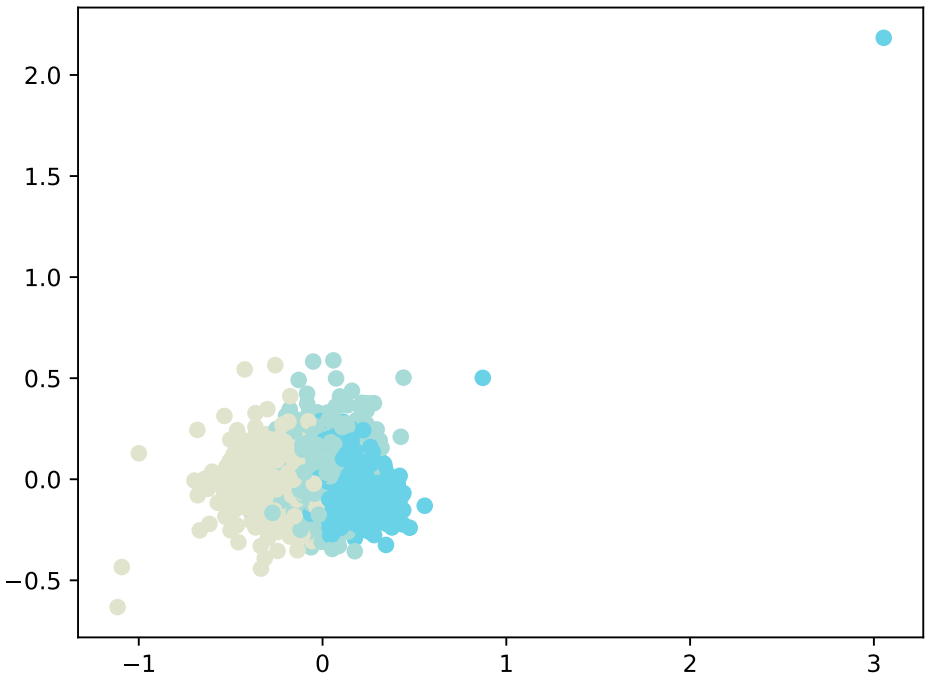
شکل ۳: tf-idf:Hierarchical:First 2 components PCA:3



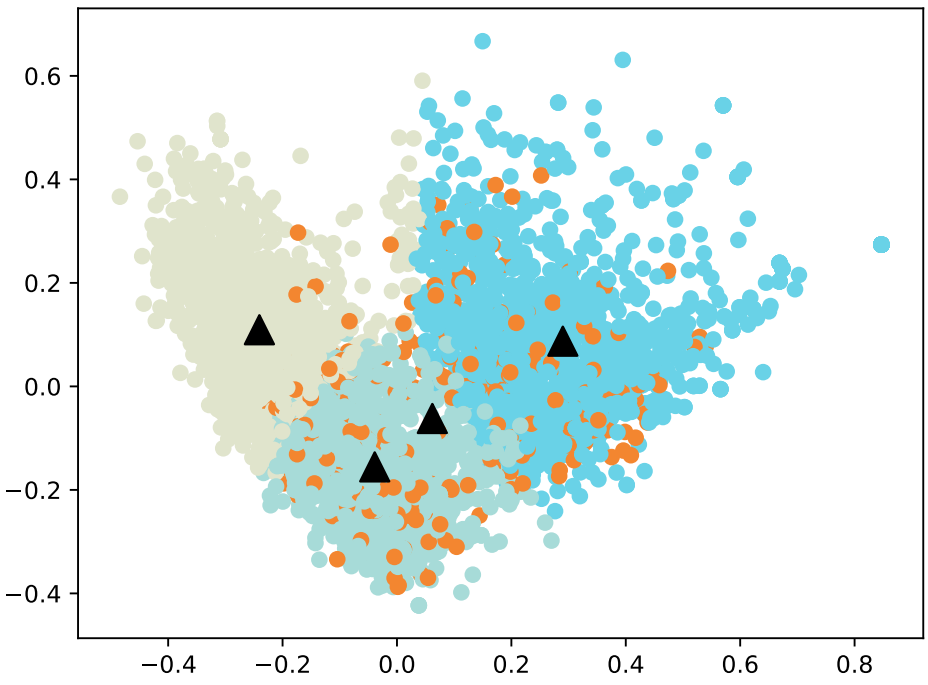
شکل ۴: word2vec:KMeans:First 2 components PCA:3



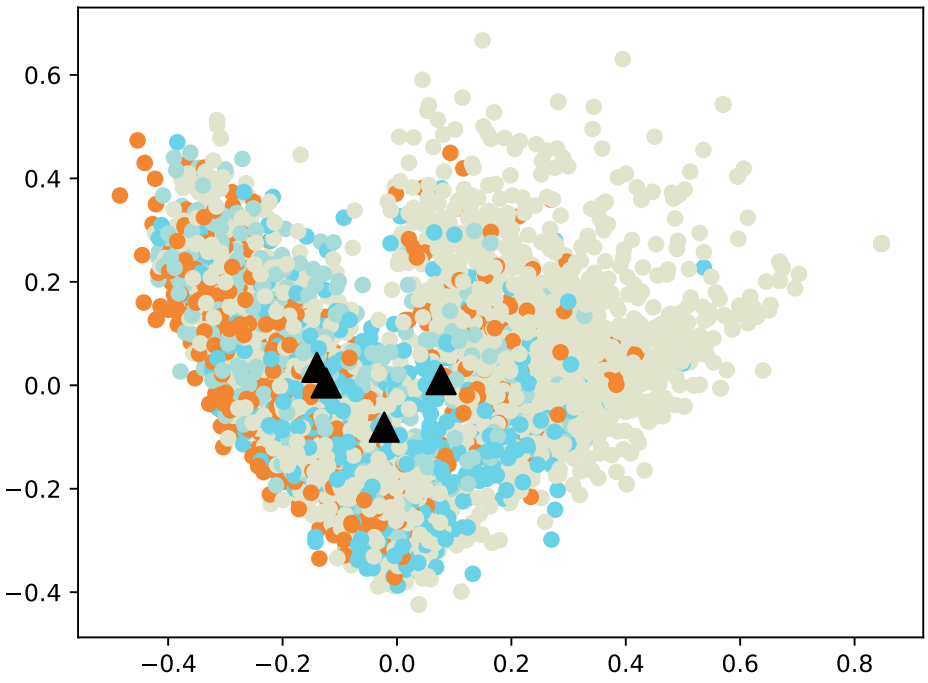
شکل ۵: word2vec:GMM:First 2 components PCA:3



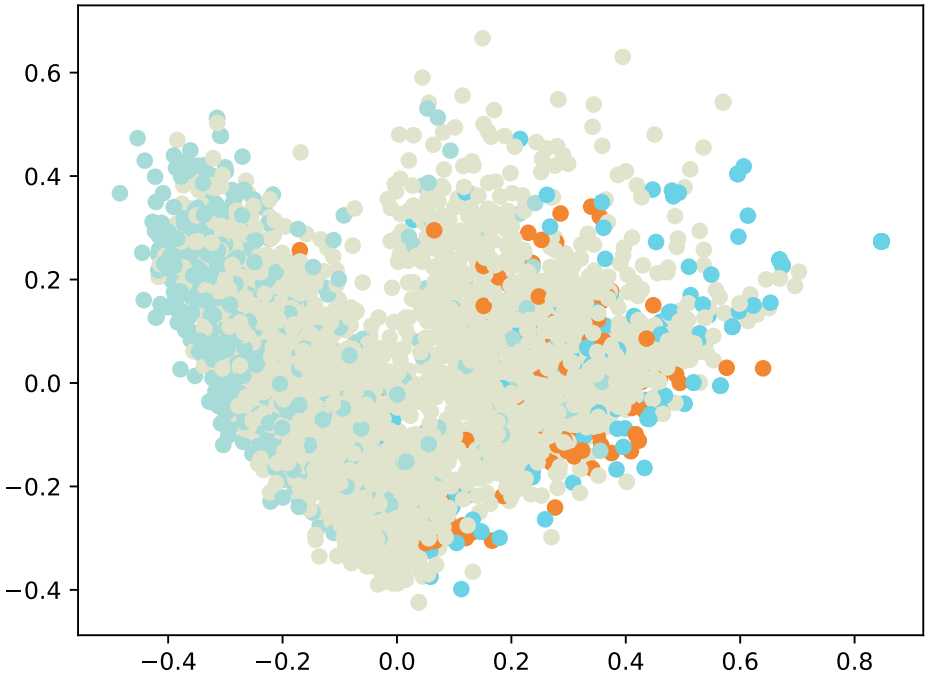
شکل ۶: word2vec:Hierarchical:First 2 components PCA:3



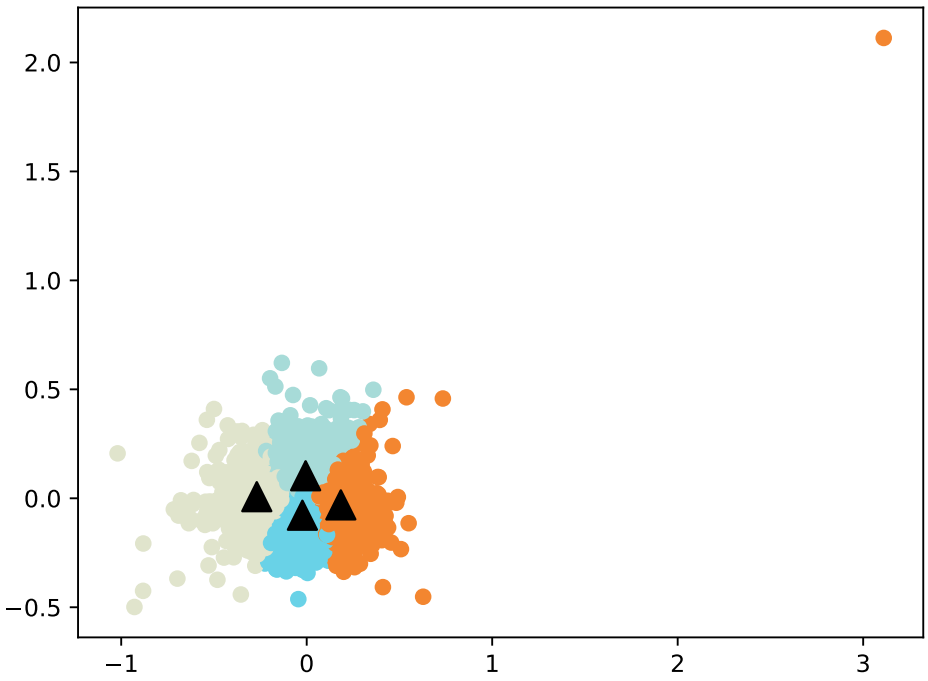
شکل ۷: tf-idf:KMeans:First 2 components PCA:4



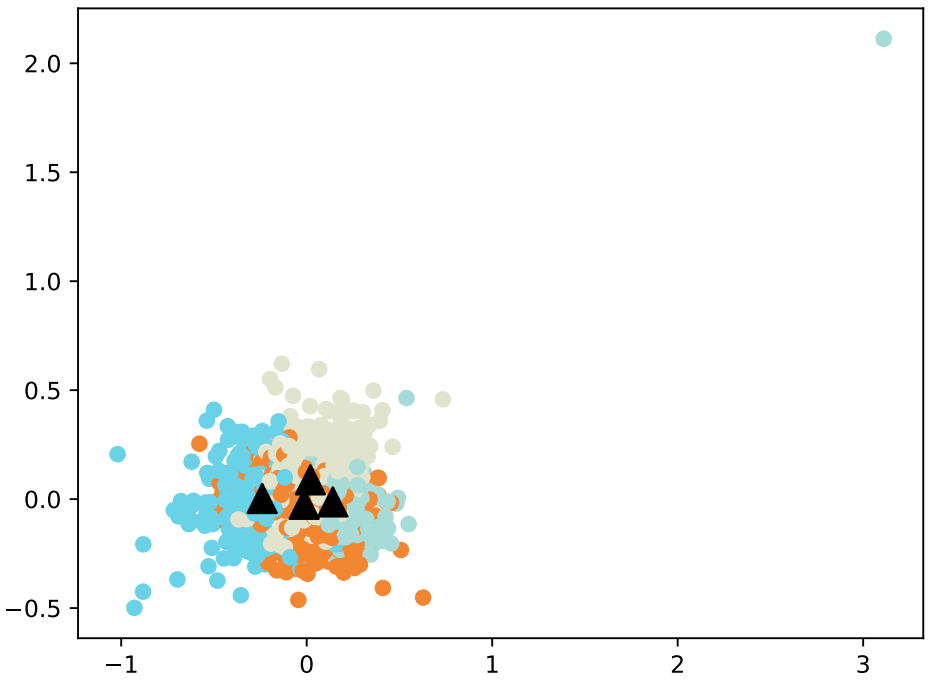
شکل ۸: tf-idf:GMM:First 2 components PCA:4



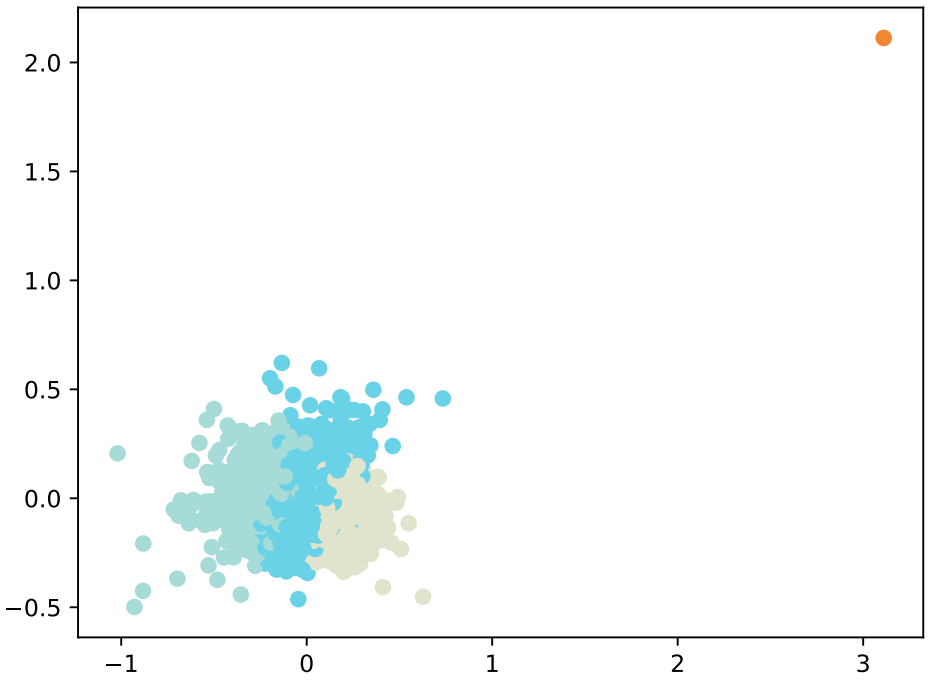
شکل ۹: tf-idf:Hierarchical:First 2 components PCA:4



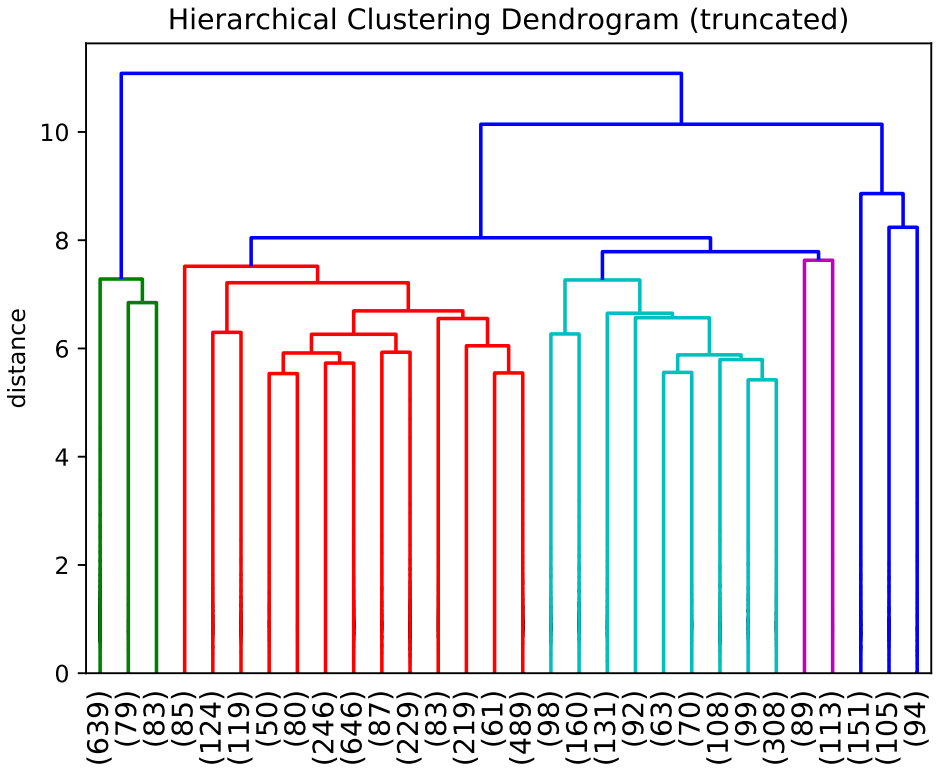
شکل ۱۰: word2vec:KMeans:First 2 components PCA:4



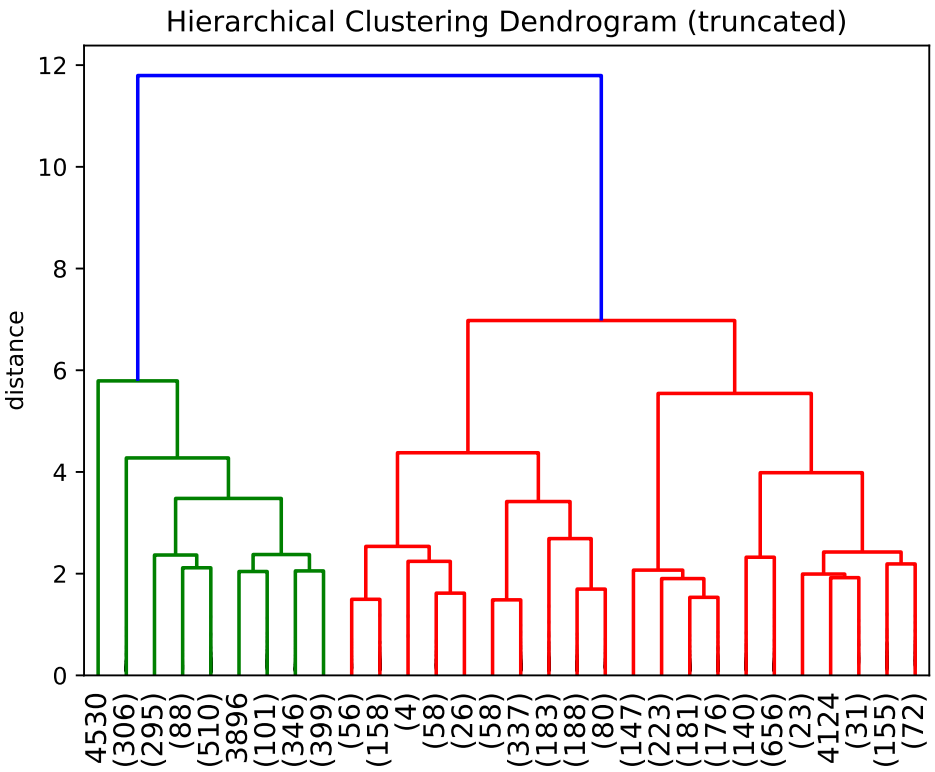
شکل ۱۱: word2vec:GMM:First 2 components PCA:4



شکل ۱۲: word2vec:Hierarchical:First 2 components PCA:4



شکل ۱۳: tf-idf:Hierarchical:Dendrgram



شکل ۱۴: word۲vec:Hierarchical:Dendrgram

۲ بخش دوم ـ خزش مقالات

برای واکشی اطلاعات از کتابخانه‌ی requests پایتون برای دریافت صفحات وب، lxml برای parse کردن و استخراج اطلاعات از آن‌ها، و کتابخانه‌ی بومی پایتون به نام json برای ذخیره و بازخوانی اطلاعات در فایل استفاده کردیم. موارد عنوان، تاریخ انتشار، مقدمه، و نویسندگان در تگ‌های meta که ویژگی name آن‌ها نوعشان و ویژگی content آن‌ها مقدارشان را مشخص می‌کرد موجود بود. برای استخراج ارجاعات از بخش با references id موارد موجود را استخراج کردیم که با یک کلاس مشترک بودند. لینک به مقالات بعدی هم در یک تگ a در داخل آن‌ها بود که چون تنها یک تگ بود کافی بود اولین تگ را را بررسی کنیم.

یک چالش این بود که در بعضی مقالات ارجاعی نبود و باید وجود این بخش را قبل از تلاش برای استخراج بررسی می‌کردیم تا با خطای حین اجرا مواجه نشویم. چالش دیگر حجم زیاد مقالات و دشواری دریافت تمام آن‌ها در یک اجرا بود. برای همین هر ۲۵ مقاله، مقالاتی که تا الان استخراج شده‌اند و صف استخراج را در فایل‌هایی ذخیره می‌کردیم و امکان این که از فایل ادامه دهیم را فراهم کردیم. با این حال استخراج تمام ۵۰۰۰ مقاله بسیار زمان‌بر و دشوار بود. در نهایت هم نتیجه را به صورت یک دیکشنری از id به اطلاعات مقاله به فرمت json ذخیره کردیم.

۳ Page Rank

برای محاسبه‌ی page rank از الگورتم کتاب برای محاسبه‌ی توزیع پایای قدم‌زن تصادفی استفاده کردیم. از کتابخانه‌ی numpy برای محاسبه‌ی گام به گام ماتریس گذر استفاده کردیم و با استفاده از روش توانی با شرط توقف تغییر کم در گام آخر توزیع پایا را محاسبه کردیم.