به نام خدا

تمرین چهارم بازیابی اطلاعات

محمد ناصری

۸۱۰۱۰۰۴۸۶

دی ۱۴۰۰

Contents

Word Association – بخش ۱

در این تمرین هدف استخراج روابط syntagmatic و paradigmatic بین کلمات با استفاده از Mutual-Information (MI) است.

در مرحله اول، جهت نرمال‌سازی مجموعه داده[1]، پیش‌پردازش‌های لازم را بر روی داده‌ها انجام میدهیم.

پیش‌پردازش‌های انجام شده در این پروژه عبارت‌اند از:

- حذف Stop-wordها
- عملیات Tokenization توسط nltk-tokenizer
- عملیات Stemming توسط SnowballStemmer
- حذف اعداد با کمک RegEx
- حذف کلمات و رشته‌های با طول کمتر از ۳ با کمک RegEx
- تبدیل همه حروف به حرف کوچک (lowerize)
- حذف ترم‌های با Doc frequency بیشتر از 0.5

در ادامه با کمک کتابخانه sklearn.feature_extraction.text و کلاس CounterVectorizer ماتریس باینری وقوع کلمات در اسناد را بدست می آوریم که سطرها نشان‌دهنده اسناد و ستون‌ها نشان‌دهنده هر کلمه میباشند و هر Occurrence[I, j] نشان‌دهنده وقوع یا عدم وقوع کلمه j در سند i میباشد.

{ نکته‌ای که در پیاده‌سازی این پروژه قابل ذکر است این است که در مرحله‌های متفاوت هر داده محاسبه شده و به دست آمده را در فایل‌هایی ذخیره میکنیم تا در آینده برای استفاده مجدد نیاز به محاسبات نباشد. برای ذخیره‌سازی و بازیابی این object ها از کتابخانه Pickle استفاده میکنیم که ماژول مربوطه در data_manager.py تعریف و پیاده‌سازی شده است. }

با داشتن ماتریس وقوع کلمات میتوانیم با ضرب ماتریس ترانهاده این ماتریس در خود آن، مقادیر وقوع همزمان کلمات را بدست آورد. دلیل این اتفاق هم به وضوح قابل مشاهده است که چون ماتریس وقوع باینری است با ضرب ترانهاده آن در خودش به ازای هر کلمه با کلمه دیگر تعداد (۱) های مشترک مشاهده شده با هم جمع میشوند و

[1] Dataset

ماتریس حاصل یک ماتریس co_occurrence[n, n] میباشد که n تعداد کلمات vocabulary میباشد و هر co_occurrence[i, j] نشان دهنده تعداد وقوع همزمان دو کلمه i, j در اسناد میباشد.

با به دست آوردن این مقادیر و ذخیره‌سازی آنها به مرحله بعدی پردازش می‌وریم.

[مراحل بالا در COUNTER_VECTORIZER انجام شده است]

در ادامه برای محاسبه مقادیر MI بین کلمات داده‌های ذخیره شده را بازیابی کرده و در ابتدا تعداد تکرار هر کلمه در corpus را بدست آورده و ذخیره میکنیم. در ادامه با توجه به مقادیر احتمالی نرمال شده وقوع یک کلمه و وقوع همزمان کلمات مقدار MI را محاسبه میکنیم و در ماتریسی ذخیره میکنیم (باقی احتمالات از روی همین ۳ احتمال قابل محاسبه هستند)

$$I(X_{w1}; X_{w2}) = \sum_{u \in \{0,1\}} \sum_{v \in \{0,1\}} p(X_{w1} = u,\ X_{w2} = v) \log_2 \frac{p(X_{w1} = u,\ X_{w2} = v)}{p(X_{w1} = u) p(X_{w2} = v)}$$

$$p(X_{w1} = 1) = \frac{count(w1) + 0.5}{N + 1}$$

$$p(X_{w2} = 1) = \frac{count(w2) + 0.5}{N + 1}$$

$$p(X_{w1} = 1,\ X_{w2} = 1) = \frac{count(w1,\ w2) + 0.25}{N + 1}$$

$$p(X_{w1} = 1,\ X_{w2} = 0) = p(X_{w1} = 1) - p(X_{w1} = 1,\ X_{w2} = 1)$$

$$p(X_{w1} = 0,\ X_{w2} = 1) = p(X_{w2} = 1) - p(X_{w1} = 1,\ X_{w2} = 1)$$

$$p(X_{w1} = 0,\ X_{w2} = 0) = 1 - p(X_{w1} = 0,\ X_{w2} = 1) - p(X_{w1} = 1,\ X_{w2} = 0) - p(X_{w1} = 1,\ X_{w2} = 1)$$

با داشتن ماتریس امتیازهای MI برای محاسبه ارتباط paradigmatic از کتابخانه sklearn.metrics.pairwise و کلاس cosine_similarity برای محاسبه شباهت کسینوسی بردارهای دو کلمه استفاده میکنیم. لازم به ذکر است بردار هر کلمه را برداری شامل تمامی کلمات vocabulary در نظر

میگیریم که وزن دهی به ترم‌های بردار توسط MI انجام شده. به وضوح این بردار برابر یک سطر یا ستون ماتریس امتیازهای MI بدست آمده می‌باشد.

برای محاسبه ارتباط syntagmatic نیز میتوان از مقایسه مقادیر MI بین کلمات استفاده کرد که این مورد هم برابر درایه‌های ماتریس امتیاز MI می‌باشد.به طور دقیق تر مقادیر یک سطر یا ستون را با یکدیگر مقایسه میکنیم.

[مراحل بالا در MI انجام شده است]

نکته حائز اهمیت در تمامی کدهای این پروژه این است که به صورت مستقل از هم تعریف شده و باید مستقلا و به ترتیب اجرا شوند. برای مثال برای بخش ۱ لازم است ابتدا counter_vectorizer اجرا شده تا اطلاعات لازم پردازش و ذخیره شوند. پس از آن باید mi اجرا شود تا اطلاعات mutual info محاسبه و ذخیره شوند و در نهایت هر یک از syntagmatic-relation یا paradigmatic-relation اجرا شوند تا ارتباطات نمایش داده شوند.

ارتباطات Syntagmatic با teacher, iran

برای کلمه teacher نتایج به صورت زیر است:

1 : unrest --> 0.0022

2 : teacher --> 0.00272

3 : bankwork --> 0.00272

4 : laden --> 0.00245

5 : grind --> 0.0023

6 : paulo --> 0.00245

7 : Fleischer --> 0.00272

8 : immune --> 0.00245

9 : strap --> 0.00272

10 : Brasilia --> 0.00245

برای کلمه iran نتایج به صورت زیر است:

1 : naval --> 0.00662

2 : offici --> 0.00677

3 : chines --> 0.00701

4 : hormuz --> 0.00755

5 : missil --> 0.01219

6 : iran --> 0.03581

7 : tehran --> 0.00755

8 : iraq --> 0.01342

9 : gulf --> 0.01106

10: iranian --> 0.01133

ارتباطات paradigmatic با teacher, iran

برای کلمه teacher نتایج به صورت زیر است:

1 : coalit --> 0.63204

2 : unrest --> 0.68277

3 : grind --> 0.80017

4 : paulo --> 0.8594

5 : brasilia --> 0.85823

6 : fleischer --> 1.0

7 : teacher --> 1.0

8 : strap --> 1.0

9 : bankwork --> 1.0

10: laden --> 0.8791

برای کلمه iran نتایج به صورت زیر است:

1 : kitti --> 0.58113

2 : silkworm --> 0.63646

3 : iranian --> 0.66488

4 : hormuz --> 0.65206

5 : warship --> 0.64884

6 : naval --> 0.60403

7 : iraq --> 0.68498

8 : missil --> 0.72955

9 : iran --> 1.0

10 : tehran --> 0.70276

تحلیل نتایج بخش ۱

همانطور که میدانیم ارتباط paradigmatic به معنی این است که کلمات میتوانند با یکدیگر جایگزین شوند و ارتباط syntagmatic بدین معناست که با دیدن کلمه A احتمال دیدن کلمه B چقدر است.

با توجه به این مطلب میتوانیم ببینیم که اولا ارتباطات paradigmatic از لحاظ امتیازی از امتیاز بالاتری برخوردار هستند و با دقت بیشتری میتوانیم آنها را تعیین کنیم ولی در مورد ارتباطات syntagmatic این روابط امتیازات ضعیف‌تری برخوردار هستند.

تفاوت دیگر در ارتباط کلمه با خودش است. در ارتباط paradigmatic هر کلمه بطور قطع با خودش در ارتباط است ولی در ارتباط syntagmatic برای مثال میبینیم کلمه iran با خودش امتیاز 0.03581 را دارد.

همینطور نمیتوان بطور دقیق با دانستن یکی از ارتباطات، دیگری را نتیجه گرفت برای مثال کلمه تهران با ایران امتیاز 0.7 در ارتباط paradigmatic دارد و در رتبه دوم امتیازات قرار میگیرد ولی در ارتباط syntagmatic از امتیاز 0.00755 برخوردار است که از لحاظ رتبه بندی در اواخر لیست ده تایی قرار میگیرد.

در کل نتیجه میگیریم با اینکه بین کلمات مرتبط بدست آمده شباهت وجود دارد ولی از روی یکی نمیتوان دیگری را نتیجه گرفت.

بخش ۲- Clustering

در این بخش هدف آشنایی با انواع روشهای خوشه بندی میباشد.

در مرحله اول مانند بخش قبل کلمات را نرمال‌سازی و پیش پردازش میکنیم و این بار از کتابخانه klearn.feature_extraction.text از کلاس TfidfVectorizer استفاده میکنیم تا ماتریس وزن‌دهی TF-IDF را بدست بیاوریم.

در مرحله بعد بر روی داده بدست آمده هر یک از روش‌های خوشه‌بندی ذکر شده را پیاده سازی و سپس توابع امتیازدهی را روی نتایج این داده و داده Truth بدست آمده از corpus اصلی (با استفاده از tag هر داکیومنت) مقایسه میکنیم. لازم به ذکر است در topicهای ذکر شده در متن تمرین ۲ موضوع هیچ سندی در مجموعه اسناد ندارند پس بطور پیشفرض آنها را حذف میکنیم.(Corn, wheat)

برای خوشه بندی K-mean و AgglomerativeClustering(single, average, complete) از کتابخانه sklearn.cluster و کلاسهای مربوطه استفاده میکنیم.

برای معیارهای ارزیابی از کتابخانه sklearn.metrics و کلاسهای normalized_mutual_info_score و f1_score و rand_score و Confusion_matrix و contingency_matrix استفاده میکنیم.

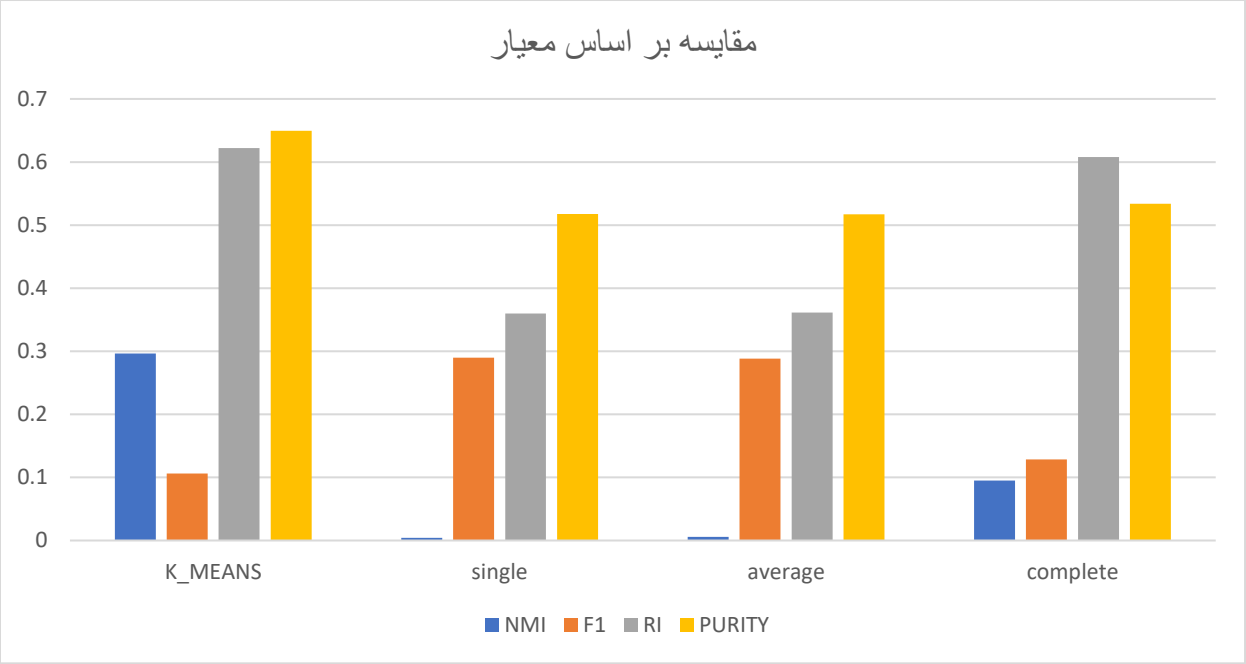
نتایج مربوطه در زیر آورده شده است.

نکته حائز اهمیت در تمامی کدهای این پروژه این است که به صورت مستقل از هم تعریف شده و باید مستقلا و به ترتیب اجرا شوند. برای مثال برای بخش ۲ لازم است ابتدا tfidf_vectorizer اجرا شده تا اطلاعات لازم پردازش و ذخیره شوند. پس از آن باید clustering اجرا شود تا مقادیر خوشه‌ها بعلاوه مقادیر معیارها پردازش و ذخیره شوند.

در بخش ۲ خروجی برنامه چندین فایل شامل اطلاعات آماری، جداول confusion ایجاد میشوند.

تحلیل نتایج بخش ۲

Column1	NMI	F1	RI	PURITY
K_MEANS	0.296387343	0.106162516	0.622014047	0.649518269
single	0.004416533	0.289947282	0.359685479	0.51736048
average	0.005859873	0.288311216	0.361211284	0.51699691
complete	0.094852827	0.128522087	0.607729173	0.533721142



جداول Confusiuon-Matrix برای روش‌های مختلف

K-MEAN	acq	crude	earn	grain	interest	money-fx	ship	trade
acq	144	1	1	700	630	0	0	120
crude	176	0	0	2	75	0	0	0
earn	236	249	420	84	1241	241	366	3
grain	30	0	0	0	11	0	0	0
interest	170	1	0	0	20	0	0	0
money-fx	200	0	0	1	21	0	0	0
ship	77	0	0	3	28	0	0	0
trade	234	0	0	0	16	0	0	0

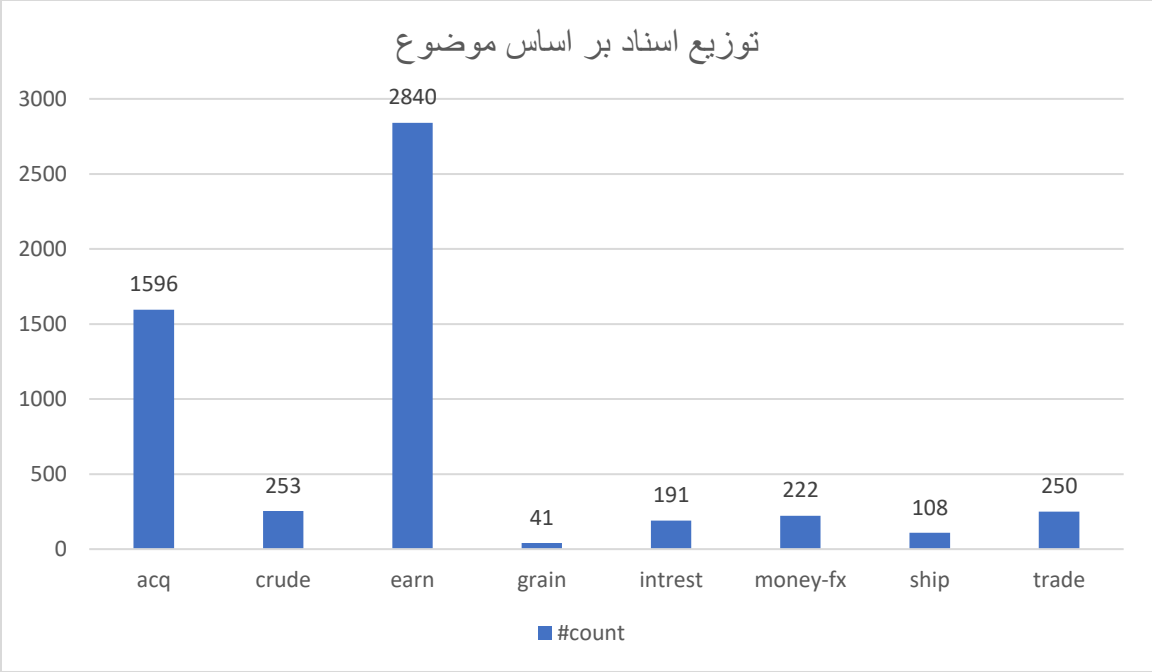
SINGLE	acq	crude	earn	grain	interest	money-fx	ship	trade
acq	1594	0	1	0	1	0	0	0
crude	252	0	0	0	0	1	0	0
earn	2839	0	0	0	0	0	1	0
grain	41	0	0	0	0	0	0	0
interest	191	0	0	0	0	0	0	0
money-fx	221	1	0	0	0	0	0	0
ship	107	0	0	1	0	0	0	0
trade	249	0	0	0	0	0	0	1

AVERAGE	acq	crude	earn	grain	interest	money-fx	ship	trade
acq	1585	3	5	1	2	0	0	0
crude	252	0	0	1	0	0	0	0
earn	2821	8	1	1	2	2	4	1
grain	39	0	0	0	0	2	0	0
interest	190	0	1	0	0	0	0	0
money-fx	222	0	0	0	0	0	0	0
ship	108	0	0	0	0	0	0	0
trade	249	0	0	1	0	0	0	0

COMPLETE	acq	crude	earn	grain	interest	money-fx	ship	trade
acq	228	86	173	19	51	586	453	0
crude	20	7	100	19	2	44	61	0
earn	418	149	382	56	92	1090	633	20
grain	0	11	7	5	4	11	3	0
interest	21	2	6	10	40	27	24	61
money-fx	92	7	5	33	17	17	35	16
ship	13	39	4	8	4	9	28	3
trade	15	4	7	111	37	50	26	0

توضیح لازم: ماتریس Confusion عبارت است از یک ماتریس C(NxN) بطوریکه هر سطر و ستون آن نشان دهنده خوشه‌ها(دسته بندی ها) و هر درایه $C_{i,j}$ نشان دهنده تعداد کلماتی است که در ground_truth داخل دسته i قرار داشته ولی در مدل بدست آمده در دسته j قرار گرفته اند.

با توجه به مقادیر بدست آمده به تحلیل نتایج میپردازیم. معیار purity در تمامی نمونه خوشه بندی‌ها مقدار تقریبا نزدیک و برابری دارند ولی k-means مقدار اندکی نسبت به باقی برتری دارد. این مقدارهای برابر نشان‌دهنده توزیع تقریبا یکسان کلمات در خوشه‌ها پس از خوشه بندی در روش‌های متفاوت اجرا شده است و در هر خوشه کمی بیش از نیمی از کلمات در یک کلاس قرار دارند.



در مورد معیار RI یا Rand Index مشاهده میشود که مقادیر این معیار برای روش‌های Single-Link و Average-Link از دو روش دیگیر با اختلاف کمتر است دلیل این امر را با بررسی فرمول این معیار بررسی میکنیم:

$$RI = \frac{TP + TN}{TP + FP + TN + FN}$$

یا بطور کلی تر :

RI = (number of agreeing pairs) / (number of pairs)

با دانستن موارد بالا و با بررسی جداول Confusion متوجه علت این تفاوت امتیاز معیار برای روش‌های Single-Link و Average-link میشویم. در این دو روش بر خلاف باقی، اکثریت اسناد در یک دسته بندی قرار گرفته اند و باعث کاهش جفت‌های مثبت شده اند.

در رابطه با معیار بعدی داریم:

```
F1 = 2 * (precision * recall) / (precision + recall)
```

از طرفی میدانیم:

$$P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{TP + FN}$$

با این دانش و بررسی جداول و نمودارها متوجه علت بالا بودن این معیار برای دو روش Single-Link و Average-Link میشویم. دلیل این اتفاق این است که با اینکه تقریبا همه اسناد در این روش‌ها به اشتباه به یک خوشه انتساب داده شده‌اند ولی از طرفی مقدار hit و مثبت اسناد همان خوشه عدد بالاییست که موجب افزایش امتیاز این معیار شده است.

در مورد معیار بعدی یعنی NMI میدانیم که:

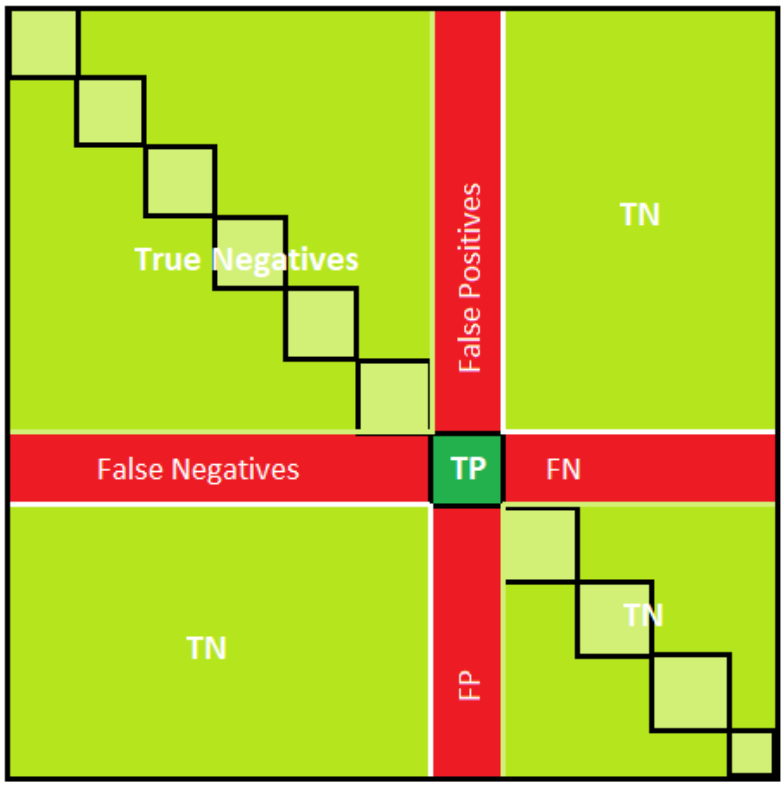
$$NMI(\Omega; C) = \frac{I(\Omega; C)}{[H(\Omega) + H(C)]/2}$$

این معیار بدلیل نرمال بودن از بهترین معیارها برای مقایسه خوشه‌بندی‌هاست. همانطور که از نمودارها و جداول مشخص است امتیاز این معیار برای روش K-means از همه بیشتر و در رتبه دوم روش Complete-link قرار دارد. دو روش دیگر امتیاز بسیار پایین و نزدیک به صفری دارند که دلیل این امر کاملا از روی جداول Confusion قابل بررسی است که کیفیت خوشه بندی در این دو روش بسیار کم است.

به صورت یک نتیجه‌گیری کلی میتوان گفت که در میان معیارهای ارزیابی، عنوان بهترین معیار ارزیابی متعلق به MI بوده و همچنین بدترین معیار ارزیابی معیار F1 میباشد که نتایج کاملا برعکس واقعیت ارائه داده است.

از دو معیار دیگیر نیز اگر بخواهیم برای رتبه‌بندی روش‌های خوشه‌بندی استفاده کنیم جواب درستی میدهند ولی شاید اطلاعات دقیق و درستی برای مقایسه یک روش با روش دیگر در اختیار ما نگذارند.

در ادامه برای به دست آوردن اینکه کدام یک از کلاس‌ها باعث افزایش Fp و Fn میشوند به بررسی ماتریس Confusion میپردازیم.



با توجه به تعاریف به تقسیم بندی بالا برای بدست آوردن مقادیر مورد نظر میرسیم که با توجه به این تصویر برای مثال در روش k-means داریم که کلاس های acq و interest باعث افزایش FP و کلاس earn باعث افزایش FN میشوند.

پایان.