-۱

برای پروژه این درس پیاده‌سازی سگمنتر گفتمانی (ابتدا با روش SVM و بعد امتحان سایر روش‌های پیشنهادی) در چارچوب نظریه ساخت معانی بیان (RST) را انتخاب کرده‌ایم. هدف این سگمنتر گفتمانی تقطیع متن به واحدهای گفتمانی اولیه (edu) است. خروجی این سگمنتر می‌تواند در مرحله بعد به تجزیه‌گری داده شود تا تجزیه‌گر روابط بین واحدهای گفتمانی را پیش‌بینی کند؛ به همین علت، طبعا عملکرد سگمنتر تاثیر مستقیم بر عملکرد تجزیه‌گر خواهد داشت. ما ابتدا سگمنتر را با روش SVM (مانند تجزیه‌گر هیلدا[1] که از این روش استفاده کرده است) پیاده‌سازی خواهیم کرد و در صورت نیاز، به روش‌های دیگر متوسل خواهیم شد. به علاوه، مطلوب ما این است که بخشی از پیاده‌سازی تجزیه‌گر را نیز در پروژه این درس به سرانجام برسانیم. هرچند، مطالعات ما درباره روش‌های پیاده‌سازی تجزیه‌گر هنوز در مراحل اولیه است؛ اما از جدیدترین تجزیه‌گرها (که عملکرد بهتری نسبت به روش‌های قاعده‌بنیاد اولیه مانند روش‌های ابتدایی دنیل مارکو داشته‌اند) می‌توانند به تجزیه‌گر DPLP[2] اشاره کرد که تجزیه‌گری با الگوریتم Shift reduce پیاده‌سازی کرده است.

از آن‌جا که کار بر روی سگمنتر در حال حاضر اولویت اول ماست، ابتدا به پیاده‌سازی و بررسی سگمنتر خواهیم پرداخت.

اگر بخواهیم با یک مثال ساده نمونه ورودی و خروجی این سگمنتر را نشان دهیم، می‌توانیم برای مثال فرض کنیم که متنی با محتوای زیر داریم:

«کتابخانه خیلی شلوغ است و من نمی‌توانم کارم را انجام بدهم. اگر بچه‌ها کمی آرامتر صحبت کنند تمرکز کردن آسان‌تر می‌شود.»

آن‌گاه سگمنتر ما باید چهار واحد گفتمانی از این متن استخراج کند و متن را به صورت زیر تقطیع کند:

۱کتابخانه خیلی شلوغ است۱۲و من نمی‌توانم کارم را انجام بدهم.۲۳اگر بچه‌ها کمی آرامتر صحبت کنند۳۴تمرکز کردن آسان‌تر می‌شود.۴

[1] https://github.com/NLPbox/hilda-docker
[2] https://github.com/jiyfeng/DPLP

۲-

الف)

$$1 = \int_{-\infty}^{+\infty} f_X(x)\,\mathrm{d}x$$

$$1 = \int_{-\infty}^{-1} f_X(x)\,\mathrm{d}x + \int_{-1}^{+1} f_X(x)\,\mathrm{d}x + \int_{+1}^{+\infty} f_X(x)\,\mathrm{d}x$$

$$1 = 0 + \int_{-1}^{+1} a(1-x^2)\,\mathrm{d}x + 0$$

$$1 = a\int_{-1}^{+1} (1-x^2)\,\mathrm{d}x$$

$$1 = a(x - \frac{1}{3}x^3\Big|_{-1}^{+1})$$

$$1 = a(\frac{2}{3} - (\frac{-2}{3}))$$

$$1 = \frac{4}{3}a$$

$$a = \frac{3}{4}$$

ب) تابع توزیع تجمعی، برای x <= -1 برابر با صفر و برای x => +1 برابر با یک است و برای بازه بین -۱ تا ۱ برابر با انتگرال زیر است:

$$F_X(x) = \int_{-1}^{+1} \frac{3}{4}(1-x^2)\,\mathrm{d}x = -\frac{x^3}{4} + \frac{3x}{4} + \frac{1}{2}$$

بنابراین تابع توزیع تجمعی به صورت زیر خواهد بود:

$$F_X(x) = \begin{cases} 0, & x \le -1 \\ -\frac{x^3}{4} + \frac{3}{4}x + \frac{1}{2}, & -1 < x < 1 \\ 1, & 1 \le x \end{cases}$$

ج)

$$E_X(x) = \int_{-\infty}^{+\infty} x f_X(x)\,\mathrm{d}x$$

$$E_X(x) = \int_{-\infty}^{-1} x f_X(x)\,\mathrm{d}x + \int_{-1}^{+1} x f_X(x)\,\mathrm{d}x + \int_{+1}^{+\infty} x f_X(x)\,\mathrm{d}x$$

$$E_X(x) = 0 + \int_{-1}^{+1} x f_X(x)\,\mathrm{d}x + 0$$

$$E_X(x) = \int_{-1}^{+1} \frac{3}{4} x(1-x^2)(x)\,\mathrm{d}x$$

$$E_X(x) = \frac{3}{4}\int_{-1}^{+1} x(1-x^2)\,\mathrm{d}x$$

$$E_X(x) = \frac{3}{4}(\frac{1}{2}x^2 - \frac{1}{4}x^4\Big|_{-1}^{+1})$$

$$E_X(x) = \frac{3}{4}(\frac{1}{4} - \frac{1}{4}) = 0$$

این مقدار به نظر طبیعی و معقول می‌رسد زیرا در این تابع:

$f(x) = f(-x)$

۳-

P(SPAM) = 0.135

P(FREE) = 0.045

P(SPAM|FREE) = 0.035

الف)

P(FREE|SPAM) = ?

P(FREE|SPAM) = (P(SPAM|FREE) * P(FREE)) / P(SPAM)

P(FREE|SPAM) = (0.035 * 0.045) / 0.135 ≈ 0.012

ب)

$P(HAM|FREE) = ?$

$P(HAM|FREE) + P(SPAM|FREE) = 1$

$P(HAM|FREE) = 1 - P(SPAM|FREE) = 0.965$

۴-

$I(x, y) = \log_2 p(x|y)/p(x)$

PMI در واقع به ما نشان می‌دهد آیا دو رویداد x و y بیشتر از زمانی که مستقل در نظرشان بگیریم، با هم رخ می‌دهند یا خیر. PMI، زمانی که برای جفت کلمات محاسبه شود، همان‌طور که مانینگ و شوتز در کتاب بنیان‌های پردازش زبان طبیعی آماری نیز اشاره کرده‌اند، در کشف باهم‌آیی‌ها و ارتباط معنایی دو واژه در متن بسیار پرکاربرد است.در این کاربرد، فرمول فوق به این معنی است که آیا کلمه x و کلمه y بیشتر از زمانی که مستقل در نظرشان بگیریم با هم رخ می‌دهند یا خیر. مانینگ و شوتز در کتاب مشترکشان به تفصیل مثال‌هایی از استخراج باهم‌آیی‌های متن با استفاده از این روش آورده‌اند.یکی از مثال‌های آن‌ها عبارت «آیت‌الله روح‌الله» است. آیا این دو کلمه تشکیل یک باهم‌آیی می‌دهند؟ به جدول زیر نگاه کنیم:

کلمه ۱	کلمه ۲	#کلمه ۱	#کمله ۲	#کلمه ۱کلمه ۲	I(کلمه ۱, کلمه ۲)
آیت‌الله	روح‌الله	۴۲	۲۰	۲۰	۱۸/۳۲

بنابراین می‌توان گفت این دو کلمه باهم‌آیی دارند و به لحاظ معنایی نیز بسیار به هم مرتبط هستند.
همچنین، برای مثال PMI(drink, beer) > PMI(drink, homework)[3] است؛ یعنی ارتباط معنایی بین drink و beer بیشتر از drink و homework است و drink و beer باهم‌آیی بیشتری دارند.
از آن‌جا که PMI همانطور که گفتیم میزان ارتباط معنایی بین کلمات را نیز نشان می‌دهد، از آن در مسائل مختلف تحلیل احساسات، قطبیت و عقیده‌کاوی نیز می‌توان استفاده کرد. برای مثال مقاله افرون

[3] https://www3.cs.stonybrook.edu/~ychoi/cse628/lecture/03-ngram.pdf

(۲۰۰۴)[4] و مقاله رید(۲۰۰۴)[5] به ترتیب از آن در علوم سیاسی محاسباتی و تحلیل حسی متن استفاده می‌کنند. افرون با استفاده از PMI میزان ارتباط یک سند با جامعه دست‌راستی/چپی را بسنجد. و رید نیز در کار خود از PMI برای جهت‌یابی معنایی واژه‌ها و عبارات استفاده می‌کند. او گروه‌های صفتی/قیدی را از متن استخراج می‌کند و برای هر عبارت، میزان نزدیکی آن عبارت را با دو دسته از کلمات (در واقع دو کلاس. دو قطب مخالف. همان‌طور که افرون دو حزب مخالف را دو کلاس در نظر گرفته بود.) محاسبه می‌کند و از PMI میانگین برای تخمین جهت معنایی عبارت استفاده می‌کند.

۵-

$$p(x) = e^{-\lambda}\frac{\lambda^x}{x!}$$

$$l(\phi) = \sum_{x=1}^{n} log(p(x_k|\phi))$$

$$= \sum_{x=1}^{n} log(e^{-\lambda}\frac{\lambda^x}{x!})$$

$$= \sum_{x=1}^{n} log(e^{-\lambda}) + log(\lambda^x) - log(x!)$$

$$= \sum_{x=1}^{n} log(e^{-\lambda}) + \sum_{x=1}^{n} log(\lambda^x) - \sum_{x=1}^{n} log(x!)$$

$$= -n\lambda + log(\lambda)\sum_{x=1}^{n} x - \sum_{x=1}^{n} log(x!)$$

Now we take partial derivative with respect to lambda and equate it to zero:

$$l'(\phi) = -n + \frac{1}{\lambda}\sum_{x=1}^{n} x - 0 = 0$$

$$n = \frac{1}{\lambda}\sum_{x=1}^{n} x$$

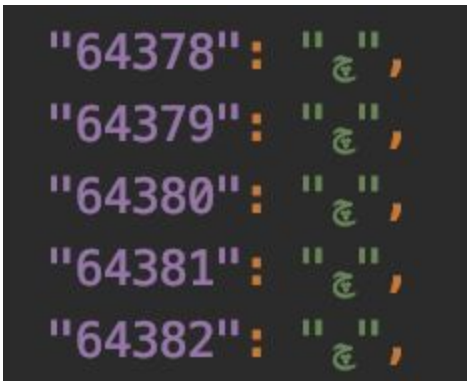
$$\lambda = \frac{\sum_{x=1}^{n} x}{n}$$

[4] Efron, M. (2004, November). The liberal media and right-wing conspiracies: using cocitation information to estimate political orientation in web documents. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management* (pp. 390-398). ACM.

[5] Read, J. (2004). Recognising affect in text using pointwise-mutual information. *Unpublished M. Sc. Dissertation, University of Sussex, UK*.

۶- گزارش پیاده‌سازی:

پیش از محاسبه احتمال‌ها از روی پیکره، سعی کردیم متن را تا حد قابل‌قبولی نرمال کنیم.این کار را متد normalizeCorpus انجام می‌دهد. برای این کار فایلی به اسم CharacterMapping ساختیم که در آن مشخص کرده‌ایم هر کاراکتر باید به چه کاراکتری نگاشت شود؛ برای مثال انواع «ی»، «گ»، «چ» و غیره. در این فایل حدود ۳۸۰ نگاشت وجود دارد. در زیر نگاشت انواع حرف «چ» را می‌بینید:



الف) متد calculateFZero برای محاسبه آنتروپی با در نظر گرفتن احتمال یکسان برای همه حروف نوشته شده است. این متد پارامتری به نام numberOfLetters دارد که در صورت لحاظ نکردن فاصله آرگومان ۳۲ (ما همزه را نشمارده‌ایم) و در صورت لحاظ کردن فاصله عدد ۳۳ را به آن می‌دهیم.

خروجی این تابع به ترتیب برای ۳۲ و ۳۳ حرف به صورت زیر است:

ب) متد calculateMonogramEntropy (پس از نرمال شدن پیکره توسط متد normalizeCorpus و/یا محاسبه احتمال مونوگرام حروف) آنتروپی را بر اساس همه کاراکترها من‌جمله علائم سجاوندی و کاراکترهای خاص محاسبه می‌کند. این متد یک پارامتر normalized دارد: در صورتی که بخواهید احتمالات مونوگرام را با پیکره نرمال‌شده محاسبه کنید، آرگومان True به آن بدهید. در زیر خروجی این

متد را با نرمال کردن و بدون نرمال کردن می‌بینید. در صورتی که نرمال‌سازی انجام شود، جمعا ۸۰ حرف و در غیر این صورت ۱۲۲ حرف خواهیم داشت.

```
خروجی قسمت ب - آنتروپی تمام حروف بدون نرمال کردن
4.410366589460107
خروجی قسمت ب - آنتروپی تمام حروف با نرمال کردن
4.251874008120504
```

ج) پاسخ این بخش متد calculateStandardEntropy است که یک پارامتر removeSpace دارد. در صورتی که این پارامتر را True کنیم، فواصل در نظر گرفته نمی‌شوند. برای محاسبه آنتروپی فقط برای حروف استاندارد فارسی، تمام حروف غیراستاندارد از پیکره حذف شد. برای این کار، فایل NonAlphaCharacters تهیه شد که هر کاراکتری جز حروف الفبا در آن به رشته خالی نگاشت می‌شود. متد standardizeCorpus با استفاده از این فایل، عملیات استانداردسازی و حذف حروف اضافه را انجام می‌دهد. در زیر آنتروپی محاسبه‌شده توسط این متد را با و بدون در نظر گرفتن فاصله می‌بینید:

```
خروجی قسمت ج - با در نظر گرفتن فاصله
4.085888501899546
خروجی قسمت ج - بدون در نظر گرفتن فاصله
4.208995175434527
```

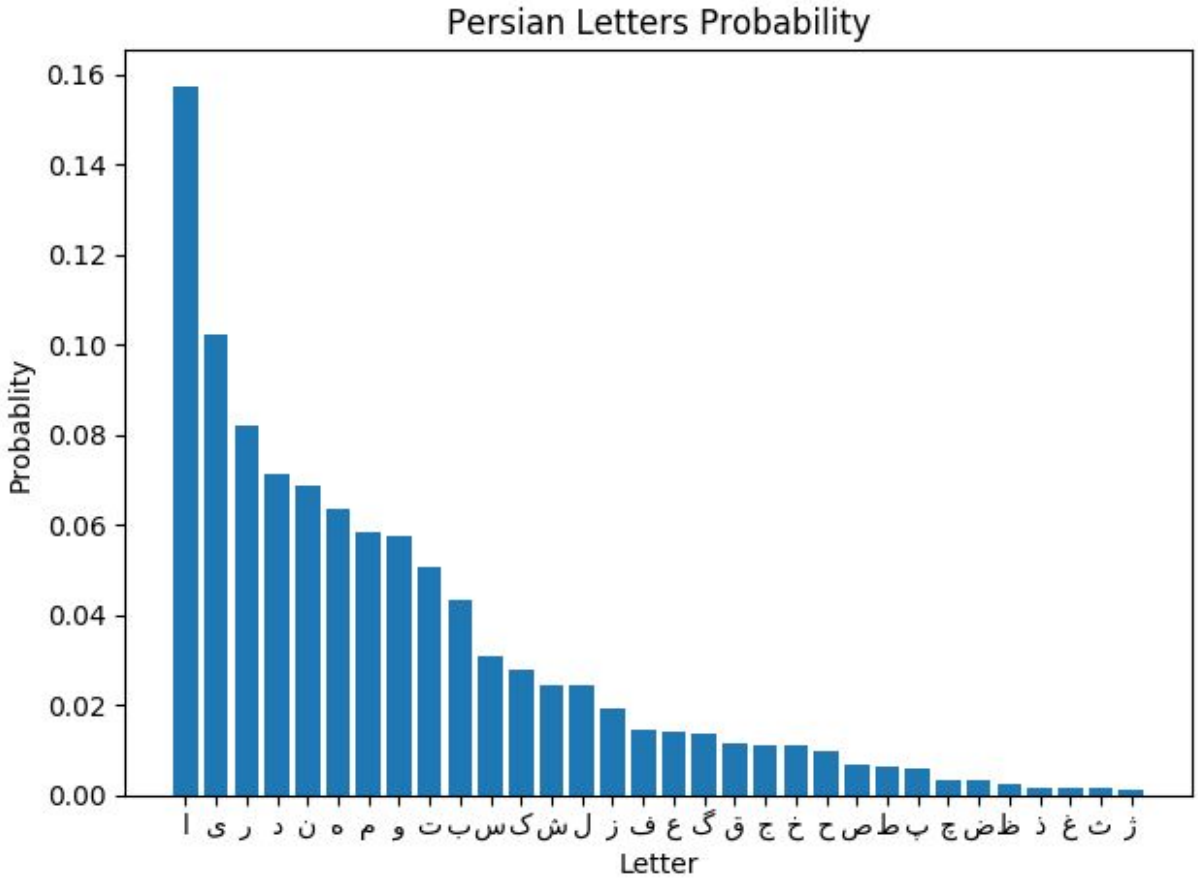
در بخش اول، با یکسان در نظر گرفتن احتمال همه حروف، آنتروپی در ماکسیمم خود قرار دارد و کمینه تعداد بیت‌ها ۵ بیت است. در حالت فعلی، احتمال‌ها از روی پیکره محاسبه شده است. بنابراین آنتروپی و عدم قطعیت کاهش یافته است (به ۴/۲ و ۴/۰۹ رسیده است.). به عبارت دیگر، زبان فارسی در بخش اول غیرقابل‌پیش‌بینی‌تر می‌نماید در حالی که در بخش دوم با در نظر گرفتن احتمالات پیکره قابل‌پیش‌بینی‌تر می‌شود.

د) متد calculateEntropyForWord پاسخ این بخش را محاسبه می‌کند. (به کمک متد calculateAverageWordLength) این متد نیز پارامتر removeSpace دارد؛در صورتی که آرگومان True داده شود، آنتروپی بی احتساب فاصله و در صورتی که False داده شود، آنتروپی با احتساب فاصله خواهیم داشت. خروجی این متد به شکل زیر است.

```
خروجی قسمت د - آنتروپی کلمه با فاصله = آنتروپی حرف با احتساب فاصله * طول کلمه
16.76838471047517
خروجی قسمت د - آنتروپی کلمه بدون فاصله = آنتروپی حرف بدون احتساب فاصله * طول کلمه
17.273611434430496
```

بنابراین، به نظر می‌رسد زبان فارسی نسبت به زبان انگلیسی (که آنتروپی هر کلمه در آن ۱۱/۸ است زبان غیرقابل‌پیش‌بینی‌تری است).

ه)

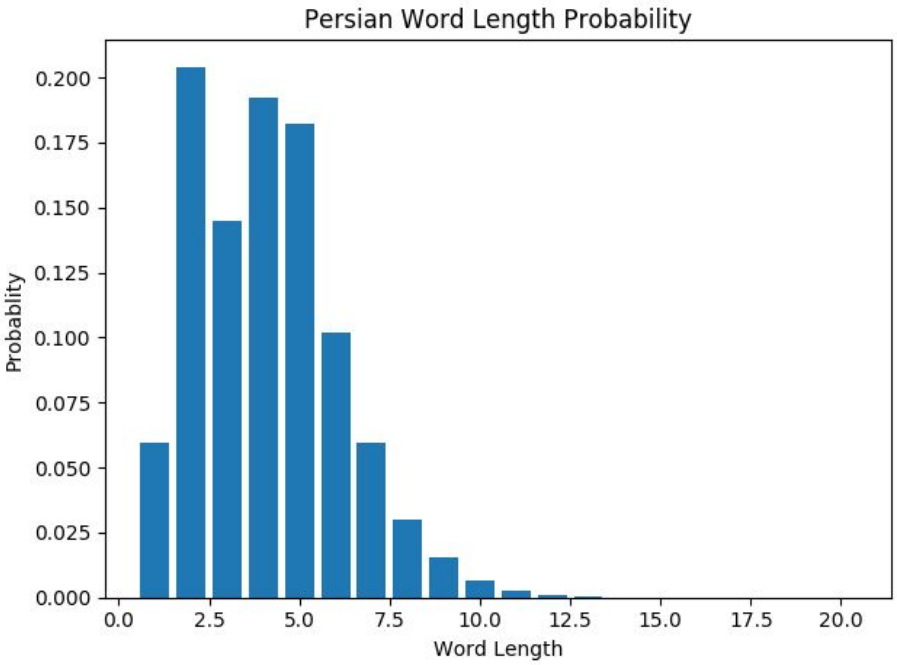


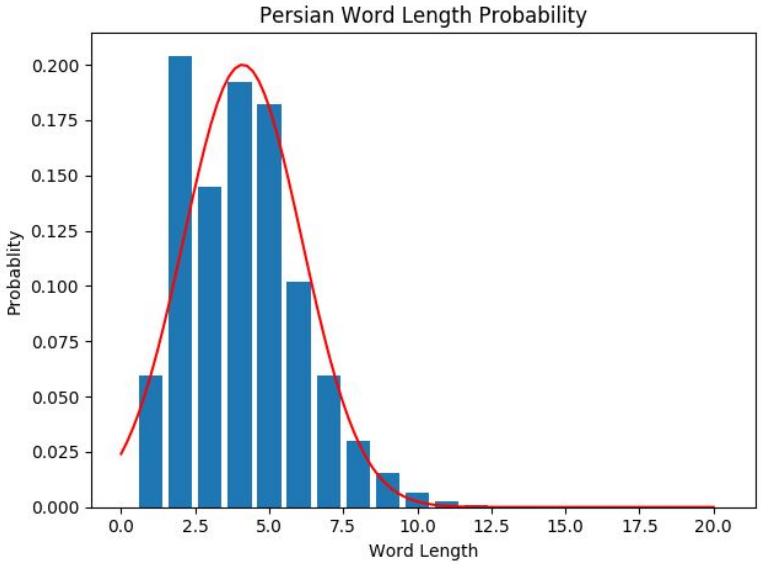
و) متد calculateGaussianParameters برای محاسبه میانگین و واریانس نوشته شده است و متد drawGaussianCurve نیز برای کشیدن منحنی گوسی روی هیستوگرام است.

میانگین = ۴/۱۰۴

واریانس = ۳/۹۷۷

انحراف معیار = 1.9942667243086694





****

ما F2 , F3 (آنتروپی بر حسب بایگرام‌ها و تریگرام‌ها) را نیز با استفاده از توابع calculateF2 و calculateF3 محاسبه کردیم. محاسبه آنتروپی Fn بر اساس فرمول شانون که در زیر آمده بوده است.

$$F_N = -\sum_{i,j} p(B_i, j)\log_2 p(j, B_i) + \sum_i p(B_i)\log_2 p(B_i)$$

در زیر، جدولی متناظر جدول شانون برای زبان انگلیسی، برای فارسی و بر اساس پیکره زبرا می‌بینید:

	F0	F1	F2	F3
۳۲ حرف	5	4.086	3.592	2.778
۳۲ حرف + فاصله	5.044	4.209	3.888	3.205