به نام خدا

دانشگاه صنعتی امیرکبیر

(پلی‌تکنیک تهران)

درس پردازش زبان طبیعی

استاد ممتازی

تمرین اول

علیرضا مازوچی

۴۰۰۱۳۱۰۷۵

بخش دوم: مدل‌های زبانی آماری

پیش از هر چیز لازم است توضیحاتی در مورد نرمال‌سازی استفاده شده بدهم. فرمول نرمال‌ســـازی Absolute Discounting در حالت عادی و برای Bigram عبارت است از:

$$P(W_i|\ W_{i-1}) = \frac{\max(\#(W_{i-1}, W_i) - \delta, 0)}{\#(W_{i-1})} + \alpha . P_{BG}$$

$$\alpha = \frac{\delta}{\#(W_{i-1})} . B$$

چنانچه برای یک جفت کلمه W_{i-1} در زمان آموزش وجود داشــته باشــد، روابط فوق قابل استفاده است. در پیاده‌سازی من برای سرعت بیشتر مقادیر B به ازای کلمات موجود در لغت‌نامه یک بار محاسبه و نگهداری می‌شود. همچنین اگر یک کلمه یک بار و در انتهای یک مصـرع آمده باشـد دارای مقدار B برابر با صـفر خواهد بود. در نتیجه مقدار α هم برابر با صفر می‌شود. بدین ترتیب رابطه فوق امکان تولید احتمال صفر را خواهد داشت که این امر باعث مشکلاتی در قسمت‌های بعدی می‌شود. برای حل این مشکل من تمامی B هایی که برابر با صفر هستند را یک واحد افزایش دادم.

مسـئله دیگری که باید به آن پرداخت اسـتفاده از رابطه نرمال‌سـازی مذکور برای جفت‌هایی اســت که W_{i-1} دیده نشــده اســت. برای این حالت مقدار B، $\#(W_{i-1})$ و $\#(W_{i-1}, W_i)$ همگی برابر با صفر می‌شود. طبیعتا اولین جمله در رابطه نرمال‌سازی و α دیگر قابل تعریف نیســتند. برای این حالت بنا به ســادگی نرمال‌ســازی را به شــکل زیر تعریف کرده‌ام:

$$P(W_i|W_{i-1}) = P_{BG}$$

احتمال P_{BG} هم مطابق معمول برابر با Unigram کلمه W_i تعریف می‌شود.

$$P_{BG} = P(W_i)$$

حال نوبت به تعریف Unigram مطابق با نرمال‌سازی Absolute Discounting می‌رسد! برای Unigram احتمال P_{BG} برابر با Zerogram می‌شود. یعنی:

$$P_{BG} = P(\ ) = \frac{1}{|V|}$$

در این رابطه V برابر با لغت‌نامه داده‌های آموزشی است. از آنجایی که در Unigram کلمه W_{i-1} تعریف نشده است، می‌توان فرض کرد که اندازه کل کلمات پیکره یعنی N را برای $\#(W_{i-1})$ درنظر گرفت و $\#(W_{i-1}, W_i)$ را معادل $\#(W_i)$ دانست. با منطقی مشابه می‌توان اندازه لغت‌نامه را برای مقدار B تعیین کرد. پس روابطی که برای Bigram داشتیم به این شکل تغییر می‌کند:

$$\alpha = \frac{\delta}{\#(W_{i-1})}.B = \frac{\delta}{N}.|V|$$

$$\frac{\max(\#(W_{i-1}, W_i) - \delta, 0)}{\#(W_{i-1})} + \alpha.P_{BG} = \frac{\max(\#(W_i) - \delta, 0)}{N} + \frac{\delta}{N}.|V|.\frac{1}{|V|}$$
$$= \frac{\max(\#(W_i) - \delta, 0) + \delta}{N}$$

نهایتا برای یک احتمال یک کلمه خواهیم داشت:

$$P(W_i) = \begin{cases} \frac{\#(W_i)}{N} & W_i \in Vocabulary \\ \frac{\delta}{N} & W_i \notin Vocabulary \end{cases}$$

در روابط نرمال‌سازی باید ابرپارامتر δ وجود دارد که باید آن را تنظیم دقیق کرد. برای تنظیم این پارامتر مطابق درخواست سوال از مقدار Perplexity روی مجموعه اعتبارسنجی کمک گرفته‌ام. مقدار δ بهینه برای Unigram برابر با ۰/۹۶ و این مقدار برای Bigram برابر با ۰/۸۸ بدست آمد.

مقدار Perplexity برای دو مدل Unigram و Bigram با مقادیر بهینه و برای سه مجموعه‌داده در جدول زیر آورده شده است:

	Train	Validation	Test
مدل Unigram	۱۷۶۸	۱۸۲۸	۱۸۱۰
مدل Bigram	۳۲۲	۱۳۲۶	۱۳۱۸

بخش سوم: تکمیل جملات ناقص با استفاده از مدل‌های زبانی آماری

در جدول زیر برچسب‌های صحیح به همراه برچسب‌های پیشنهادی دو مدل آورده شده است:

خروجی	متن
خروجی درست	این سخن حقست اگر نزد سخن گستر برند
خروجی Unigram	این سخن حقست اگر نزد سخن گستر و
خروجی Bigram	این سخن حقست اگر نزد سخن گستر و
خروجی درست	آنکه با یوسف صدیق چنین خواهد کرد
خروجی Unigram	آنکه با یوسف صدیق چنین خواهد و
خروجی Bigram	آنکه با یوسف صدیق چنین خواهد کرد
خروجی درست	هیچ دانی چکند صحبت او با دگران
خروجی Unigram	هیچ دانی چکند صحبت او با و
خروجی Bigram	هیچ دانی چکند صحبت او با تو
خروجی درست	سرمه دهی بصر بری سخت خوش است تاجری
خروجی Unigram	سرمه دهی بصر بری سخت خوش است و
خروجی Bigram	سرمه دهی بصر بری سخت خوش است و
خروجی درست	آتش ابراهیم را نبود زیان
خروجی Unigram	آتش ابراهیم را و و
خروجی Bigram	آتش ابراهیم را به دست
خروجی درست	من که اندر سر جنونی داشتم
خروجی Unigram	من که اندر سر و و
خروجی Bigram	من که اندر سر و از
خروجی درست	هر شیر شرزه را که به نیش سنان گزید
خروجی Unigram	هر شیر شرزه را که به نیش و و
خروجی Bigram	هر شیر شرزه را که به نیش و از
خروجی درست	هرکه از حق به سوی او نظریست
خروجی Unigram	هرکه از حق به و و و

خروجی Bigram	هرکه از حق به دست و از
خروجی درست	گفت این از خدای باید خواست
خروجی Unigram	گفت این از و و و
خروجی Bigram	گفت این از آن که در
خروجی درست	کلاه لاله که لعل است اگر تو بشناسی
خروجی Unigram	کلاه لاله که لعل است و و و
خروجی Bigram	کلاه لاله که لعل است و از آن

از آنجایی که معیار این بخش یک معیار کیفی است، من یک معیار کمی هم تعریف کرده‌ام. می‌توان به ازای هر مصرع در مجموعه‌داده یک پنجره لغزان به طول تعداد لغات پیشین و لغت هدف تعریف کنیم و به ازای کلمه‌های پیشین از مدل انتظار داشته باشیم تا کلمه نهایی را پیش‌بینی کند. مثلا برای مصرع «توانا بود هر که دانا بود» و برای مدل Bigram داده‌هایی مانند («توانا» و «بود») و («بود» و «هر») و غیره ساخته می‌شود. در این حالت انتظار داریم مدل با دیدن «توانا» بتواند «بود» را پیشبینی کند. برای مدل Unigram پنجاه هزار داده تصادفی از هر یک از سه مجموعه موجود ساختم و برای مدل Trigram با توجه به محدودیت‌های زمان اجرای مدل، ده هزار داده تصادفی ایجاد کردم. نتایج ارزیابی این قسمت در جدول زیر آورده شده است:

	Train	Validation	Test
مدل Unigram	۴٫۰۳	۳٫۹۹	۳٫۹۷
مدل Bigram	۱۲٫۰۸	۷٫۶۸	۷٫۹۹

حال باید نتایج کمی و کیفی را مقایسه کنیم. طبیعی است که مدل Unigram تنها می‌تواند پرتکرارترین کلمه را در کل مجموعه‌داده آموزشی را به عنوان پیشبینی خود در هر جایگاهی ارائه دهد. برای مجموعه‌داده فعلی کلمه «و» پرتکرارترین است. در نتایج کیفی به وضوح مشخص است که این مدل نتایج خوبی ندارد. با بررسی نتایج کمی هم می‌بینیم که مدل با این پیش‌بینی ثابت به حدود ۴٪ دقت می‌رسد. مدل Bigram اما به یک کلمه قبل‌تر اهمیت می‌دهد برای نتایج کیفی بعضا عملکرد خوبی داشته است. مثلا داده دوم را کاملا درست گفته است و بیش‌بینی‌اش برای سایر داده‌ها بعضا

قابل قبول است هرچند مساوی با پیش‌بینی مدنظر نبوده است. دقت این مدل در معیار کمی و برای مجموعه آزمون برابر ۸٪ است که دو برابر حالت Unigram است. واضح است که این مدل از نظر دقت بسیار کارآمدتر از مدل Unigram است.

در عین حال باید توجه داشت که مدل Bigram از نظر زمان اجرا کند است. چراکه برای پیشبینی باید تمام لغت لغت‌نامه را بررسی کند و ببینید برای کدام کلمه بیشترین احتمال وجود دارد و آن را برگرداند که این کار واقعا زمانگیر است حتی برای مجموعه‌داده کوچک این سوال.

بخش چهارم: ایجاد مدل زبانی با استفاده از شبکه عصبی

مدل این قسمت را مطابق با معماری پیشنهادی ساختیم. بهتر است که چندین توضیح مختصر راجع به پیاده‌سازی مدل داشته باشم:

۱. اندازه لایه تعبیه و اندازه لایه خروجی برابر با یک واحد بیشتر از اندازه لغت‌نامه است. علت این امر در آن است که در زمان آزمون ممکن است یک کلمه برای اولین بار دیده شود و خارج از لغت‌نامه باشد. لذا لازم است لایه تعبیه یک جای خالی برای این لغات داشته باشد. همچنین برای محاسبه Perplexity نیاز است که احتمال خروجی را برای یک کلمه خارج از لغت‌نامه داشته باشیم. برای همین من یک نورون خروجی برای کلمات خارج از لغت‌نامه قرار دادم تا این احتمال را در خود داشته باشد.
۲. لایه‌های شبکه‌ای که من استفاده کردم نسبتا ابعاد بالایی دارد و مدل استعداد شدیدی در بیش‌برازش دارد. برای جلوگیری از این مورد و با کمک یک Callback و مجموعه اعتبارسنجی جلوی بیش‌برازش را گرفته‌ام. زمانی که این Callback تعریف نشده‌بود مدل این امکان را داشت که دقتی بالای ۵۰٪ روی مجموعه‌آموزشی و کمتر از ۴٪ روی مجموعه اعتبارسنجی داشته باشد!
۳. استراتژی آموزش با استفاده از معیار کمی تعریف‌شده انجام‌شده است. یعنی آنکه به مدل کلمات پیشین را دادم و از آن خواستم تا کلمه هدف را پیش‌بین کند.

۴. مدل با دو گام آموزش بهترین نتایج را داشته است.

مقدار Perplexity برای دو مدل Bigram و Trigram و برای سه مجموعه‌داده در جدول زیر آورده شده است:

	Train	Validation	Test
مدل Bigram	۱۹۷۰	۲۱۱۳	۲۱۰۰
مدل Trigram	۲۱۱۳	۲۳۰۹	۲۲۸۹

بخش پنجم: تکمیل جملات ناقص با استفاده از مدل‌های زبانی شبکه عصبی

در جدول زیر برچسب‌های صحیح به همراه برچسب‌های پیشنهادی دو مدل آورده شده است:

خروجی	متن
خروجی درست	این سخن حقست اگر نزد سخن گستر برند
خروجی Bigram	این سخن حقست اگر نزد سخن گستر و
خروجی Trigram	این سخن حقست اگر نزد سخن گستر به
خروجی درست	آنکه با یوسف صدیق چنین خواهد کرد
خروجی Bigram	آنکه با یوسف صدیق چنین خواهد و
خروجی Trigram	آنکه با یوسف صدیق چنین خواهد به
خروجی درست	هیچ دانی چکند صحبت او با دگران
خروجی Bigram	هیچ دانی چکند صحبت او با تو
خروجی Trigram	هیچ دانی چکند صحبت او با را
خروجی درست	سرمه دهی بصر بری سخت خوش است تاجری
خروجی Bigram	سرمه دهی بصر بری سخت خوش است و
خروجی Trigram	سرمه دهی بصر بری سخت خوش است و
خروجی درست	آتش ابراهیم را نبود زیان
خروجی Bigram	آتش ابراهیم را ز من
خروجی Trigram	آتش ابراهیم را و به

خروجی درست	من که اندر سر جنونی داشتم
خروجی Bigram	من که اندر سر و به
خروجی Trigram	من که اندر سر تو و
خروجی درست	هر شیر شرزه را که به نیش سنان گزید
خروجی Bigram	هر شیر شرزه را که به نیش و به
خروجی Trigram	هر شیر شرزه را که به نیش تو و
خروجی درست	هرکه از حق به سوی او نظریست
خروجی Bigram	هرکه از حق به جای آن که
خروجی Trigram	هرکه از حق به و تو دل
خروجی درست	گفت این از خدای باید خواست
خروجی Bigram	گفت این از آن که در
خروجی Trigram	گفت این از دل آن که
خروجی درست	کلاه لاله که لعل است اگر تو بشناسی
خروجی Bigram	کلاه لاله که لعل است و به جای
خروجی Trigram	کلاه لاله که لعل است و و به

مشابه با مدل‌های آماری، دقت‌هایی را مطابق با معیار کمی تعریف‌شده در بخش‌های قبل محاسبه کردیم.

	Train	Validation	Test
مدل Bigram	۸/۳۵	۶/۳۳	۶/۱۹
مدل Trigram	۹/۲۱	۶/۶۵	۶/۵۹

در نتایج کمی به نظر می‌رسد تفاوت خیلی شدید بین دو مدل مانند دو مدل آماری نیست ولی به هر حال مدل Trigram برای هر سه مجموعه‌داده توانسته است دقت بهتری داشته باشد. در نتایج کیفی هیچ مدلی نتوانسته است حتی یک جای خالی را به درستی پیش‌بینی کند.

بخش ششم: تحلیل نتایج

برای انجام مقایسه و تحلیل جامع بهتر است نتایج کمی را در یک جدول تجمیع کنیم:

	Train	Validation	Test
مدل Unigram آماری	۴٫۰۳	۳٫۹۹	۳٫۹۷
مدل Bigram آماری	۱۲٫۰۸	۷٫۶۸	۷٫۹۹
مدل Bigram عصبی	۸٫۳۵	۶٫۳۳	۶٫۱۹
مدل Trigram عصبی	۹٫۲۱	۶٫۶۵	۶٫۵۹

نتایج کمی نشان می‌دهد که مدل Bigram آماری بهترین مدل من بوده است و حتی دقت بهتری از مدل Trigram عصبی داشته است. نتایج کیفی هم نشان می‌دهد که کیفیت خروجی هر دو مدل عصبی از مدل Bigram آماری بدتر است. به عنوان مثال مدل Trigram عصبی دو «و» را پشت سر هم برای داده آخر کیفی آورده است و مدل Bigram آماری حداقل یک مورد توانسته است درست پیش‌بینی کند.

در اینجا باید توجه کنیم که مدل‌های عصبی باتوجه به محدودیت حافظه‌ای که داشتند در روی مجموعه‌داده کوچکتری آموزش پیدا کرده‌اند. این موضوع می‌تواند علت دقت پایین این مدل‌ها در برابر مدل آماری Bigram باشد. شاید اگر مجموعه‌داده‌ها یکسان بود مدل‌های عصبی می‌توانستند بهتر باشند. این موضوع مطرح‌شده در کنار آنکه دفاعی از مدل‌های عصبی من ارائه می‌دهد، ثابت می‌کند که این مدل‌ها با شبکه پیشنهادی و پیاده‌سازی من دچار محدودیت حافظه هستند که نقصی بر مدل‌های عصبی من است.

تحلیل دیگری که باید به آن توجه داشت، زمان اجراست؛ مدل‌های عصبی زمان اجرای قابل قبولی دارند درحالی که مدل Bigram آماری کند است. این مسئله باعث می‌شود تا در شرایطی که سرعت برای ما مهم است مدل آماری را کنار بگذاریم. البته باید توجه داشت که در شبکه‌های عصبی موازی‌سازی وجود دارد و این موازی‌سازی باعث کاهش زمان اجرا شده است. طبیعی است که اعمال روش‌های موازی‌سازی روی

مدل‌های آماری هم می‌تواند زمان اجرای آن‌ها را بهبود دهد ولی در پیاده‌سازی فعلی چنین امکانی درنظر گرفته نشده است.

نهایتا خوب است نگاهی هم بر Perplexity مدل‌ها داشته باشیم:

	Train	Validation	Test
مدل Unigram آماری	۱۷۶۸	۱۸۲۸	۱۸۱۰
مدل Bigram آماری	۳۲۲	۱۳۲۶	۱۳۱۸
مدل Bigram عصبی	۱۹۷۰	۲۱۱۳	۲۱۰۰
مدل Trigram عصبی	۲۱۱۳	۲۳۰۹	۲۲۸۹

باتوجه به این مقادیر باید مدل Bigram آماری بهترین دقت را داشته باشد که صحیح است. از طرف دیگر مقدار Perplexity برای داده‌های آموزش همواره کمتر از دو داده دیگر است که شدیدترین حالت آن در Bigram آماری دیده می‌شود؛ چنین چیزی قابل توجیه است چراکه در زمان آموزش لغات دیده‌نشده وجود ندارد و احتمال‌های پایین کمتر است برخلاف زمان تست که تعداد زیادی احتمال پایین Perplexity مدل را خراب می‌کند. نکته عجیب در این نتایج Perplexity بیشتر مدل‌های عصبی از مدل نامناسب Unigram آماری است. به نظر می‌رسد کوچک‌تر بودن مجموعه آموزشی مدل‌های عصبی باعث این امر است. طبیعی است که هرچه مجموعه آموزشی غنی‌تر باشد مدل می‌تواند احتمال‌های بهتری را محاسبه کند. شاید این شبهه به وجود آید که چرا برای مجموعه آموزش هم Perplexity مدل‌های عصبی بیشتر است. در پاسخ باید گفت که این امر به دلیل عدم استفاده از تمام داده‌های آموزشی در زمان آموزش مدل ولی سنجش با کل داده‌های آموزشی در زمان محاسبه Perplexity بوده است.

جدای از مقایسه مدل‌ها با یکدیگر، دقت کلی مدل‌ها واقعا پایین است. یک مدل که همواره یک کلمه را پیشنهاد می‌دهد می‌تواند دقت ۴٪ داشته باشد. پس در این صورت دقت ۸٪ نباید چندان جالب باشد. برای توجیح این امر باید ببینیم که چه داده‌ای را در اختیار مدل قرار داده‌ایم و از آن چه خواسته‌ایم. هر مدل تنها می‌تواند به حداکثر دو کلمه قبل خود دسترسی داشته باشد و از تمام دانشی که در یک مصرع وجود دارد محروم است؛ طبیعی است که در این شرایط کار مدل واقعا سخت است.

مثلا مدل Trigram برای داده سوم کیفی باید بفهمد که بعد «او با» چه کلمه‌ای می‌آید. یا مدل‌های Bigram تنها با استفاده از کلمه «با» باید کلمه بعد را تشخیص بدهند که واقعا لیست کلمات زیادی را می‌توان پیشنهاد داد. قطعا «و» پیشنهاد مناسبی نیست که برخی از مدل‌ها داده‌اند. اما کلمه «تو» مناسب است ولی مورد قبول ما نیست. با این تفاسیر به نظر می‌رسد که چندان جای پیشرفت با این مدل‌های فعلی وجود ندارد. البته که استفاده از n-gramهای طولانی‌تر و استفاده از مجموعه‌داده بیشتر می‌تواند موثر باشد.