تمرین دوم پردازش زبان های طبیعی

سوال اول

استفاده از N-gram ها در پردازش زبان های طبیعی موثر است اما با چالش هایی مواجه است که در ادامه به آن ها می پردازیم:

1- **پراکندگی داده ها (data sparsity):**
اگر مقدار n بیشتر شود تعداد ترکیب هایی که میتوان ساخت بسیار زیاد می شود و به همین دلیل خیلی از این ترکیب ها در مجموعه داده های آموزشی قرار ندارند و از این رو قابلیت تعمیم در مدل پایین میاید.
2- **مقیاس‌پذیری با مقادیر بزرگتر N:** هر چه مقدار N افزایش یابد، به داده های بزرگتر و منابع محاسباتی بیشتری نیاز دارد و در سیستم های real-time میتواند مشکل باشد.
3- **محدودیت‌های زمینه (Context):** این مدل ها نمی‌توانند وابستگی های طولانی مدت را به خوبی شناسایی کنند یعنی درک زمینه و ارتباطات دورتر در متن برای این مدل‌ها مشکل است.
4- **حساسیت به تغییرات جزئی در داده‌ها:** N-gram ها به تغییرات جزئی در متن و فرمول ها حساس هستند این یعنی اگر از مترادف یک کلمه استفاده کنیم و یا تغییر ترتیب تکلمات ممکن است مشکل ساز شود.

سوال دوم

مبحث perplexity در NLP یک معیار برای ارزیابی مدل است که نشان می دهد که مدل در پیشبینی کلمات متوالی چقدر خوب کار کرده. هر چه perplexity کمتر باشد مدل در پیش بینی کلمات بهتر عمل می کند. به عبارت ساده تر اگر برای پیش بینی یک کلمه در جمله یک گزینه برای انتخاب داشته باشد می گوییم perplexity کم است ولی اگر گزینه های زیادی وجود داشته باشد می گوییم perplexity یا پیچیدگی بالا است.

محدودیت‌های perplexity:

1- **حساسیت به اندازه کلمات و توزیع احتمالات:** مدل هایی که تعداد کلمات بیشتری دارند ممکن است احتمال کمتری به هر کلمه اختصاص دهند و از این رو perplexity بالا می رود چرا که مدل انتخاب های متنوع تری دارد.

2- **عدم ارزیابی معنایی**: perplexity بررسی می کند که احتمال وقوع کلمه بعدی بیشتر باشد و به معنای کلمه یا مناسب بودن آن توجه نمیکند.
3- **وابستگی به dataset و متن trian:** perplexity به شدت به دیتاست وابسته است و اگر روند آموزش روی متن های خبری انجام شده باشد در متن هایی که مشابه هستند perplexity پایینی دارند ولی روی متن های دیگر ممکن است perplexity بالا برود.
4- **عدم توجه به وابستگی های Long-Range:** perplexity فقط به پیش بینی های محلی توجه دارد و به معنای کلی در طول جمله یا پاراگراف اهمیت نمی دهد.

سوال سوم

Perplexity Smoothing برای حل مشکل صفر بودن احتمال در مدل های زبانی استفاده می شود برای مثال در مدل های n-gram هر ترکیبی که وجود نداشته باشد احتمال صفر دارد. تکنیک های smoothing کمک می کند که این نوع احتمالات به مقداری غیر صفر تبدیل شوند تا مدل تعمیم بهتری داشته باشد.

مزایای این امر:

- بهبود تعمیم‌دهی: این کار به مدل کمک می کند تا ترکیب‌های کلمه های دیده نشده را بهتر مدیریت کند و احتمال غیر صفر به آن ها دهد
- پایداری پیچیدگی: با جلوگیری از احتمال صفر باعث می‌شود که مقادیر پیچیدگی قابل اعتمادتر و تفسیر پذیرتر شود
- جلوگیری از Overfitting: این کار تمایل مدل به توجه بیشتر روی n-gram های پرتکرار را کاهش می دهد

مزایا و معایب روش های مختلف Smoothing Perplexity

1- Laplace (Additive) Smoothing: این روش که add-1 Laplace نیز معروف است یک مقدار ثابت که معمولا یک است را به تعداد هر n-gram اضافه میکند که از احتمالات صفر جلوگیری میکند. این روش معمولا در n-gram های بالا به خوبی کار نمیکند و در مدل های پیچیده تر دقت کمتری دارد.
2- Add-k (Additive) Smoothing: این روش مشابه روش بالا است با این تفاوت که به جای عدد 1 یک مقدار کوچکتر مانند 0.1 یا 0.001 را به تعداد اضافه میکند. اگر k درست انتخاب

شود تعادل بهتری در تعمیم دارد و دقت بیشتری نسبت به روش بالا دارد. معایب این روش نیز انتخاب دقیق k است که می تواند زمان بر باشد.

3- Good-Turing Smoothing: این روش احتمالات را براساس تعداد وقوع احتمال های نادر تنظیم می کند و به احتمالات صفر براساس فراوانی n-gram ها احتمال غیر صفر نسبت میدهد. این روش برای مدیریت رخدادهای نادر و داده‌های محدود مناسب است. از معایب این روش میتوان به پیچیدگی محاسباتی بالا و پیاده سازی سخت اشاره کرد.

4- Kneser-Ney Smoothing: این روش بر اساس تعداد زمینه های منحصر به فرد که کلمه در آن‌ها دیده شده احتمال را تنظیم می‌کند. به کلمات پرتکرار با زمینه‌های محدود احتمال کمتری می دهد و به کلماتی با زمینه‌های متنوع تر احتمال بیشتری می‌دهد. این روش یکی از بهترین روش ها است و معایب آن پیچیدگی و منابع زیاد برای پردازش است.

سوال چهارم

a- جدول bigram

artificial , intelligence	4
Iran , is	2
in , Iran	2
is , advancing	1
advancing , rapidly	1
rapidly , in	1
in , artificial	1
intelligence , research	1
research , is	1
is , thriving	1
thriving , in	1
intelligence , is	1
is , common	1
common , in	1
is , focused	1
focused , on	1
on , artificial	1

b- محاسبه P(intelligence | artificial) و P(artificial | intelligence) : برای محاسبه این احتمال شرطی باید تعداد bigram هایی که برابر کلمه اول و دوم هستند را تقسیم کنیم به تعداد bigram هایی که اولین کلمه آن برابر کلمه اول ما باشد که به شکل زیر است:

P(intelligence | artificial) = count(artificial , intelligence) / count(artificial) = 4 / 4 = **1**

P(artificial | intelligence) = count(intelligence , artificial) / count(intelligence) = 0 / 2 = **0**

c- برای محاسبه احتمال جمله با استفاده از bigram باید از قانون chain-rule استفاده کنیم که با ضرب احتمال های شرطی هر کلمه به دست میاید.

P(S1) = P(Iran) * P(is | Iran) * P(advancing | is) * P(rapidly | advancing) * P(in | rapidly) * P(artificial | in) * P(intelligence | artificial)

P(Iran) = 1

P(is | Iran) = Count(Iran , is) / Count(Iran)= 2 / 2 = 1

P(advancing | is) = 1 / 4 = 0.25

P(rapidly | advancing) = 1 / 1 = 1

P(in | rapidly) = 1 / 1 = 1

P(artificial | in) = 1 / 3 = 0.25

P(intelligence | artificial) = 4 / 4 = 1

P(S1) = 1 * 1 * 0.25 * 1 * 1 * 0.25 * 1 = **0.0625**

d- برای محاسبه perplexity یک جمله باید اول احتمال آن را حساب کنیم که در تمرین بالا آن را حساب کردیم سپس از فرمول زیر استفاده کنیم. (n برابر تعداد کلمات است)

Perplexity(S1) = $P(S1)^{-\frac{1}{N}}$ = $0.0625^{-\frac{1}{7}}$ = **1.48**

e- برای محاسبه smoothed perplexity باید از فرمول زیر استفاده کنیم.

Perplexity smoothed(S1) = $P\ smoothed(S1)^{-\frac{1}{N}}$

N is the number of words in the sentence = 7

P smoothed(word2 | word1) = $\frac{Count(word1,word2)+1}{\text{Count(word1)} + \text{V}}$

V is the Vocabulary size = 12

حال برای محاسبه P smoothed(S1) باید دوباره با فرمول جدید احتمال جمله رو با قانون chain-rule حساب کنیم.

P(Iran) = 1

P(is | Iran) = $\frac{Count(\text{Iran,is}\ )+1}{\text{Count(Iran)} + 12}$ = 3 / 14 ≈ 0.21

P(advancing | is) = 2 / 16 = 0.125

P(rapidly | advancing) = 2 / 13 ≈ 0.153

P(in | rapidly) = 2 / 13 ≈ 0.153

P(artificial | in) = 2 / 16 ≈ 0.125

P(intelligence | artificial) = 5 / 16 ≈ 0.312

P smoothed(S1) = 1 * 0.21 * 0.125 * 0.153 * 0.153 * 0.125 * 0.312 ≈ 0.00002396

Perplexity smoothed(S1) = $0.00002396^{-\frac{1}{7}}$ = **4.571**

در اینجا مشاهده شد که مقدار perplexity معمولی برابر شد با **1.48** اما مقدار smoothed perplexity شد **4.571**. چرا که این افزایش در smoothed perplexity به این مربوط است که مقدار احتمال را در همه bigram های ممکن از جمله مواردی که در داده train مشاهده نمی شود پخش میکند. این امر باعث عدم قطعیت و در نتیجه perplexity بیشتر می شود.

f- برای محاسبه کلمه بعدی عبارت "iran is" با استفاده از مدل bigram به این صورت عمل کنیم که کلمه ای که بیشترین بار بعد از این عبارت آمده را حساب کنیم که 2 کلمه وجود دارد. 1- advancing و 2- focused که هر دو یک بار آمده است در اینجا می توان به صورت رندوم یکی را انتخاب کرد.

اما برای عبارت "Iran was" در مدل bigram ما کلمه ی was وجود ندارد که بعد Iran آمده باشد. که در این مواقع می گوییم out-of-vocabulary رخ داده و باید از روش های Smoothing و Backoff Models برای حل این مشکل استفاده کرد.

محمد حقیقت – 403722042

برای رفع برخی ایرادات و ابهامات از Chatgpt استفاده شده است.