به نام خدا

دانشگاه تهران

دانشکدگان فنی

دانشکده مهندسی برق و کامپیوتر

درس بازیابی هوشمند اطلاعات

پاسخ بخش تئوری تمرین ۲

نام و نام خانودگی: روژین پناو

شماره دانشجویی: ۲۲۰۷۰۱۰۴۶

مهر ماه ۱۴۰۳

تأیید می‌کنم که از LLM ها مطابق با دستورالعمل‌های بارگذاری شده در سامانه Elearn درس به طور مسئولانه استفاده کرده‌ام، تمام اجزای کار خود را درک می‌کنم و آماده بحث شفاهی درباره آن‌ها هستم.

فهرست

پاسخ سوال اول ۳
پاسخ بخش الف ۳
پاسخ بخش ب ۵
پاسخ بخش ج ۷
پاسخ سوال دوم ۱۱
پاسخ بخش الف ۱۱
پاسخ بخش ب ۱۵
پاسخ سوال سوم ۱۶
پاسخ بخش الف ۱۶
پاسخ بخش ب ۱۷
پاسخ بخش ج ۱۸
پاسخ بخش د ۱۹

پاسخ سوال اول

پاسخ بخش الف

$D_3 =$ "الگوریتم های پیشرفته یادگیری ماشین و کاربردهای آن در سیستم های توصیه‌گر و بازیابی اطلاعات"

۱. **Jelinek-Mercer smoothing:**

$$p(w_i \mid d) = (1-\lambda)\cdot\frac{c(w_i,d)}{\mid d \mid} + \lambda\cdot p(w_i \mid C)$$

$$p(\text{"یادگیری"} \mid D_3) = (1-\lambda)\cdot\frac{c(\text{"یادگیری"},D_3)}{\mid D_3 \mid} + \lambda\cdot p(\text{"یادگیری"} \mid C)$$

$$p(\text{"یادگیری"} \mid D_3) = 0.4\cdot\frac{1}{13} + 0.6\cdot 0.003$$

$$p(\text{یادگیری} \mid D_3) = 0.031 + 0.0018$$

$$p(\text{"یادگیری"} \mid D_3) \approx 0.0328$$

۲. **Dirichlet smoothing:**

$$p(w_i \mid d) = \frac{c(w_i,d) + \mu\cdot p(w_i \mid C)}{\mid d \mid + \mu}$$

$$p(\text{"یادگیری"} \mid D_3) = \frac{c(\text{"یادگیری"},D_3) + \mu\cdot p(\text{"یادگیری"} \mid C)}{\mid D_3 \mid + \mu}$$

$$p(\text{"یادگیری"} \mid D_3) = \frac{1 + 2000\cdot 0.003}{13 + 2000}$$

$$p(\text{"یادگیری"} \mid D_3) = \frac{1+6}{2013}$$

$$p(\text{"یادگیری"} \mid D_3) \approx 0.0035$$

۳. **Absolute Discounting:**

$$p(w_i \mid d) = \frac{\max\left(c(w_i,d) - \delta, 0\right) + \delta\cdot \mid d \mid_u \cdot p(w_i \mid C)}{\mid d \mid}$$

$$p(\text{"یادگیری"} \mid D_3) = \frac{\max\left(c(\text{"یادگیری"},D_3) - \delta, 0\right) + \delta\cdot \mid D_3 \mid_u \cdot p(\text{"یادگیری"} \mid C)}{\mid D_3 \mid}$$

در "الگوریتم‌های پیشرفته یادگیری ماشین و کاربردهای آن در سیستم‌های توصیه‌گر و بازیابی اطلاعات" ← تنها واژه "و" تکرار شده:

$$\text{تعداد واژه‌ها یکتا} = |D_3|_u = 12$$

$$p(\text{"یادگیری"} \mid D_3) = \frac{\max(1-0.5,0) + 0.5 \cdot 12 \cdot 0.003}{13}$$

$$p(\text{"یادگیری"} \mid D_3) = \frac{0.5 + 0.5 \cdot 12 \cdot 0.003}{13}$$

$$p(\text{"یادگیری"} \mid D_3) = \frac{0.5 + 0.018}{13}$$

$$p(\text{"یادگیری"} \mid D_3) \approx 0.45$$

۴. **Additive (Laplace):**

$$p(w_i \mid d) = \frac{c(w_i, d) + \alpha}{|d| + \alpha.|V|}$$

$$p(\text{"یادگیری"} \mid D_3) = \frac{c(\text{"یادگیری"}, D_3) + \alpha}{|D_3| + \alpha.|V|}$$

$$p(\text{"یادگیری"} \mid D_3) = \frac{1+1}{13 + 1.10000}$$

$$p(\text{"یادگیری"} \mid D_3) = \frac{2}{10013}$$

$$p(\text{یادگیری} \mid D_3) \approx 0.0002$$

JM	DIR	AD	Additive	
0.328	0.0035	0.45	0.0002	$p(\text{"یادگیری"} \mid D_3)$

روش Absolute Discounting بیشترین احتمال را داد، چون هم تعداد وقوع کلمه «یادگیری» را نگه می‌دارد $(c(w_i, d) - \delta)$ پس اگر کلمه در سند آمده باشد، سهم مستقیمش از بین نمی‌رود، هم یک مقدار اضافه از مدل مجموعه به آن می‌دهد این به‌خاطر تعداد واژه‌های یکتای سند بزرگتر می‌شود، پس عدد کلی AD بزرگتر می‌شود.

در JM وزن زیادی به داده‌های کلی داده شده ($\lambda = 0.6$) و سهم سند کم می‌شود، برای همین مقدارش کمتر است.

Dirichlet بیشتر به عددهای مدل مجموعه تکیه می‌کند، وقتی μ خیلی بزرگ است، یکبار وقوع کلمه "یادگیری" در سند در برابر μ تقریباً ناچیز می‌شود و نتیجه نزدیک $p(w \mid C)$ می‌ماند؛ بنابراین مقدار نهایی پایین است.

Laplace هم به همهٔ کلمه‌ها یک مقدار ثابت اضافه می کند، اما مشکلش این است که **مخرج خیلی بزرگ** می‌شود $|d|+\alpha|V|$. با واژگان بزرگم (|V|=10000)، حتی اگر شمارش کلمه "یادگیری" یک باشد، نسبت به مخرج عدد خیلی کوچک می‌شود و برای بازیابی اطلاعات مناسب نیست؛ چون واژه‌های دیده‌شده و ندیده را تقریباً یکسان رقیق می‌کند.

پاسخ بخش ب

$$q = \text{"یادگیری بازیابی"}$$

$$D_1 = \text{"بازیابی اطلاعات و یادگیری ماشین در علوم داده"}$$

$$D_3 = \text{"الگوریتم های پیشرفته یادگیری ماشین و کاربردهای آن در سیستم های توصیه‌گر و بازیابی اطلاعات"}$$

$$\log p(q \mid d) = \sum_{i=1}^{m} \log p(w_i \mid d) = \sum_{w \in V, c(w|d)>0} \mathrm{c(w,q)} \log p(\mathrm{w} \mid d)$$

باید به‌جای $p(\mathrm{w} \mid d)$ از نسخهٔ هموارشدهٔ آن استفاده کنیم با روش Dirichlet Prior:

$$p(\mathrm{w} \mid d) = \frac{c(\mathrm{w}, d) + \mu \cdot p(\mathrm{w} \mid C)}{|d| + \mu}$$

$$p(\mathrm{w} \mid d) = \frac{|d|}{|d| + \mu} \cdot \frac{c(w,d)}{|d|} + \frac{\mu}{|d| + \mu} \cdot p(\mathrm{w} \mid C)$$

محاسبه $p(\mathrm{q} \mid D_1)$:

$$p(\text{"بازیابی"} \mid D_1) = \frac{c(\text{"بازیابی"}, D_1) + \mu \cdot p(\text{"بازیابی"} \mid C)}{|D_1| + \mu}$$

$$p(\text{"بازیابی"} \mid D_1) = \frac{1 + 2000 \cdot 0.002}{8 + 2000} = 0.0024900$$

$$(\text{"یادگیری"} \mid D_1) = \frac{c(\text{"یادگیری"}, D_1) + \mu \cdot p(\text{"یادگیری"} \mid C)}{|D_1| + \mu}$$

$$p(\text{"یادگیری"} \mid D_1) = \frac{1 + 2000 \cdot 0.003}{8 + 2000} = 0.0034860$$

محاسبه $p(\mathrm{q} \mid D_3)$:

$$(\text{"بازیابی"} \mid D_3) = \frac{c(\text{"بازیابی"}, D_3) + \mu \cdot p(\text{"بازیابی"} \mid C)}{\mid D_3 \mid + \mu}$$

$$p(\text{"بازیابی"} \mid D_3) = \frac{1 + 2000 \cdot 0.002}{13 + 2000} = 0.0024839$$

$$(\text{"یادگیری"} \mid D_3) = \frac{c(\text{"یادگیری"}, D_3) + \mu \cdot p(\text{"یادگیری"} \mid C)}{\mid D_3 \mid + \mu}$$

$$p(\text{"یادگیری"} \mid D_3) = \frac{1 + 2000 \cdot 0.003}{13 + 2000} = 0.0034774$$

جایگذاری:

$$\log\ p(q \mid D_1) = \sum_{w \in V, c(w|d)>0} \mathrm{c(w,q)} \log\ p(\mathrm{w} \mid D_1)$$

$$\log p(q \mid D_1) = \mathrm{c}(\text{"بازیابی"}, \mathrm{q}) \log p(\text{"بازیابی"} \mid D_1) + \mathrm{c}(\text{"یادگیری"}, \mathrm{q}) \log\ p(\text{"یادگیری"} \mid D_1)$$

$$\log p(q \mid D_1) = 1.\log p(\text{"بازیابی"} \mid D_1) + 1.\log\ p(\text{"یادگیری"} \mid D_1)$$

$$\log p(q \mid D_1) = \log\ 0.0024900 + \log\ 0.0034860$$

$$p(q \mid D_1) = 0.0024900\ \times 0.0034860 = 0.0000086801$$

$$\log p(q \mid D_3) = \mathrm{c}(\text{"بازیابی"}, \mathrm{q}) \log p(\text{"بازیابی"} \mid D_3) + \mathrm{c}(\text{"یادگیری"}, \mathrm{q}) \log\ p(\text{"یادگیری"} \mid D_3)$$

$$\log p(q \mid D_3) = 1.\log p(\text{"بازیابی"} \mid D_3) + 1.\log\ p(\text{"یادگیری"} \mid D_3)$$

$$\log p(q \mid D_3) = \log 0.0024839 + \log 0.0034774$$

$$p(q \mid D_3) = 0.0024839 + 0.0034774 = 0.0000086375$$

$$p(q \mid D_1)\ >\ p(q \mid D_3)$$

چرا $p(q \mid D_1)\ >\ p(q \mid D_3)$ ؟

در Dirichlet پارامتر(μ=2000) خیلی بزرگ است، بنابراین سهم «شمارش واقعی در سند» نسبت به «احتمال زمینه(مدل زبانی مجموعه)» کوچک می‌شود.

$$p(\mathrm{w} \mid d) = \frac{1}{\mid d \mid + \mu} \cdot \underbrace{\frac{\mid d \mid . c(w,d)}{\mid d \mid}}_{\text{یکسان در } D_1 \text{ و } D_3} + \frac{\mu}{\mid d \mid + \mu} \cdot \underbrace{p(\mathrm{w} \mid C)}_{\text{یکسان در } D_1 \text{ و } D_3}$$

وقتی طول سند کوتاه‌تر باشد($|D_1|$=8) ، اثر مدل مجموعه و تعداد وقوع واژه در سند نسبت به مخرج |d|+μ کمی بزرگتر است؛ برای D_3 که طولش بیشتر است ($|D_3|$= 13)مدل مجموعه و تعداد وقوع واژه کم‌اثرتر می‌شود.

$$0.996016\ = \frac{2000}{2008} = \frac{\mu}{\mid D_1 \mid + \mu}\ \ >\ \ \frac{\mu}{\mid D_3 \mid + \mu} = \frac{2000}{2013} = 0.993541$$

$$0.0004980 = \frac{1}{2008} = \frac{1}{|D_1| + \mu} > \frac{1}{|D_3| + \mu} = \frac{1}{2013} = 0.0004968$$

یعنی تفاوت تنها در مخرج است. مخرج بزرگتر در D3 باعث کاهش اندکی در p(w|c) و $c(w,d)$ نسبت به D_1 شده.

پاسخ بخش ج

مدل ترکیبی:

$$P(w \mid d) = \alpha P_{JM}(w \mid d) + \beta P_{Dir}(w \mid d) + \gamma P_{AD}(w \mid d) + \theta P_{Add}(w \mid d)$$

به‌طوری‌که

$$\alpha + \beta + \gamma + \theta = 1 \qquad \alpha, \beta, \gamma, \theta \geq 0$$

۱. چه زمانی هر یک از مولفه‌ها نقش مهم‌تری دارن؟

- **Dirichlet:**

$$P_{Dir} = \frac{|d|}{|d| + \mu} \cdot \frac{c}{|d|} + \frac{\mu}{|d| + \mu} \cdot P_C$$

  - اگر $|d| \to \infty$ آنگاه: $\frac{|d|}{|d|+\mu} \to 1$ در نتیجه $P_{Dir} \to \frac{c}{|d|}$
  - اگر $|d| \ll \mu$ آنگاه: $\frac{|d|}{|d|+\mu} \approx 0$ درنتیجه $P_{Dir} \approx P_C$

هر چه $|d|$ نسبت به μ بزرگتر شود، Dirichlet بیشتر به مشاهدات و مدل سند تکیه می‌کند؛ در غیر این صورت به احتمال زمینه و مدل مجموعه تکیه می‌کند.

- **Jelinek-Mercer:**

$$P_{JM} = (1 - \lambda)\frac{c}{|d|} + \lambda P_C$$

  - اگر $\lambda \to 0$ آنگاه: $P_{JM} \to \frac{c}{|d|}$
  - اگر $\lambda \to 1$ آنگاه: $P_{JM} \to P_c$

در سندهای طولانی که $\frac{c}{|d|}$ قابل‌اطمینان است، انتخاب λ کوچک باعث می‌شود JM نقش مهم‌تری داشته باشد؛ تأثیر $|d|$ غیرمستقیم و از طریق نسبت $\frac{c}{|d|}$ است، ولی کنترل اصلی را پارامتر λ می‌کند.

- **Absolute Discounting :**

$$P_{AD} = \frac{max(c-\delta, 0) + \delta \mid d \mid_u P_C}{\mid d \mid}$$

- اگر $c \geq \delta$ آنگاه : $P_{AD} = \frac{c-\delta}{|d|} + \frac{\delta |d|_u}{|d|} P_C$

دو ترم داریم :

I. discounted ML estimate: $P_{Seen}(w|d) = \frac{c-\delta}{|d|}$

II. Collection Model: $\alpha_d P(w|C) = \frac{\delta |d|_u}{|d|} P_C$

✓ اگر $\mid d \mid = \text{constant}$ ، مشتق نسبت به $\mid d \mid_u$: $\frac{\partial P_{AD}}{\partial |d|_u} = \frac{\delta}{|d|} P_C$

چون $\delta > 0$ ، $\mid d \mid > 0$ و $P_C > 0$ ، داریم : $\frac{\partial P_{AD}}{\partial |d|_u} > 0$

(با افزایش $\mid d \mid_u$ و ثابت ماندن $\mid d \mid$) آنگاه : $P_{AD} \leftarrow$ افزایش

P_{AD} به‌صورت خطی و صعودی افزایش می‌یابد و AD به‌نفع واژه‌هایی که در سند کم‌تکرارند شانس بیشتری می‌دهد.

✓ اگر اگر $\mid d \mid \neq \text{constant}$ و نسبت $\frac{|d|_u}{|d|}$ افزایش: $\frac{\delta |d|_u}{|d|} P_C = \delta \left(\frac{|d|_u}{|d|}\right) P_C$

(با افزایش $\mid d \mid_u$ یا کاهش $\mid d \mid$) آنگاه : $P_{AD} \leftarrow$ افزایش

- اگر $c < \delta$ آنگاه : $P_{AD} = \frac{\delta |d|_u}{|d|} P_C$

باز هم استدلال بالا میتوانیم داشته باشیم.

- اگر $\delta \to 0$ آنگاه : $P_{AD} \to \frac{c}{|d|}$
- اگر $\delta \to 1$ آنگاه : $P_{AD} \to \frac{\delta |d|_u}{|d|} P_C$

P_{AD} زمانی مهم‌تر است که سند تعداد یکتاهای نسبتاً زیادی داشته باشد یا وقتی می‌خواهیم جرم احتمال discounted ML estimate را به شکل متناسب $\mid d \mid_u$ بازتوزیع کنیم.

- **Additive (Laplace):**

$$P_{Add} = \frac{c+\alpha}{|\,d\,| + \alpha.|V|}$$

- اگر $|V| \to \infty$ آنگاه: $P_{Add} \to \frac{1}{|V|}$
- اگر $|V| \to 0$ آنگاه: $\lim_{|V|\to 0} P_{Add}(w \mid d) = \frac{c+\alpha}{|d|}$
- اگر $\alpha \to \infty$ و $|V| > 0$ آنگاه: $\lim_{\alpha\to\infty} P_{Add}(w \mid d) = \lim_{\alpha\to\infty} \frac{\alpha(1+c/\alpha)}{\alpha(|V|+|d|/\alpha)} = \frac{1}{|V|}$
- اگر $\alpha \to \infty$ و $\alpha|V| \to \infty$ آنگاه: $\lim_{\alpha\to\infty} \frac{c+\alpha}{|d|+\alpha|V|} = \lim_{\alpha\to\infty} \frac{\alpha(1+o(1))}{\alpha(|V|+o(1))} = \frac{1}{\lim_{\alpha\to\infty}|V|}$
- اگر $\alpha \to \infty$ و $\alpha|V| \to 0$ آنگاه: $\lim_{\alpha|V|\to 0} P_{Add}(w \mid d) = \frac{c+\alpha}{|d|}$

پس اگر فضای واژگان کوچک باشد، Additive می‌تواند خوب باشد؛ ولی در IR عملی با |V| بزرگ این مؤلفه معمولاً رقیق‌کننده و کم‌تاثیر است.

۲. چیکار کنیم $\alpha,\beta,\gamma,\theta$ خودکار تعریف بشن؟

باید وزن‌های ترکیبی چهار مؤلفهٔ مدل زبان را طوری یاد بگیریم که ترکیب آن‌ها بیشترین احتمال را روی مجموعهٔ نگهداری (validation) بدهد. وزن‌ها باید مثبت باشند و جمع‌شان برابر ۱ شود؛ برای این کار پارامترهای آزاد z را یاد می‌گیریم و آن‌ها را با softmax به α, β, γ, θ نگاشت می‌زنیم. در ادامه روش را با مسیرهای forward و backward حل میکنیم.

تعریف مدل و نگاشت وزن‌ها

- مدل ترکیبی:

$$P(w \mid d) = \alpha P_{JM}(w \mid d) + \beta P_{Dir}(w \mid d) + \gamma P_{AD}(w \mid d) + \theta P_{Add}(w \mid d)$$

- نگاشت وزن‌ها (پارامترهای آزاد z_1, z_2, z_3, z_4):

$$\omega_i = \frac{e^{z_i}}{\sum_{j=1}^{4} e^{z_j}} \text{ که } \omega \in \{\alpha,\beta,\gamma,\theta\}.$$

مسیر– Forward

برای هر جفت (q,d) در مجموعهٔ اعتبار و برای هر واژهٔ w در q ، چهار مؤلفهٔ $P_{JM}, P_{Dir}, P_{AD}, P_{Add}$ را محاسبه می‌کنیم. (پارامترهای داخلی هر مؤلفه مثل λ، μ، δ، α قبلاً تعیین شده‌اند).

۱. با z فعلی وزن‌ها را با softmax می‌سازم و برای هر w مقدار ترکیب $P(w \mid d)$ را محاسبه می‌کنم:

$$P(w \mid d; z) = \sum_{i=1}^{4} \omega_i(z)\, P_i(w \mid d)$$

۲. امتیاز پرس‌وجو در سند را با جمعِ لگاریتم احتمال کلمات حساب میکنیم:

$$\log P(q \mid d) = \sum_{w \in q} c(w,q) \log P(w \mid d; z)$$

۳. تابع هدف کل روی مجموعهٔ اعتبار برابر است با جمع این امتیازها:

$$\mathcal{L}(z) = \sum_{(q,d) \in D_{val}} \sum_{w \in q} (w,q) \log P(w \mid d; z)$$

۴. برای پایداری یک جملهٔ منظم‌ساز اضافه می‌کنم:

$$\mathcal{L}_{reg}(z) = \mathcal{L}(z) - \frac{\lambda}{2} \| z \|^2$$

عدد $\mathcal{L}_{reg}(z)$ نشان می‌دهد وزن‌های فعلی چقدر خوبند.

مسیر– Backward

اگر برای یک واژه-سند، $P_j(w \mid d)$ از میانگین ترکیب $P(w \mid d)$ بهتر باشد، باید وزن مؤلفهٔ ز را بیشتر کنیم؛ اگر بدتر باشد، باید کمتر کنیم. نسبتِ مؤلفه به ترکیب این شهود را مشخص می‌کند:

$$R_j(w,d) = \frac{P_j(w \mid d)}{P(w \mid d)}.$$

۲. مشتقِ لگاریتم برای هر z_j (نتیجهٔ عملی محاسبات گرادیان) چنین است:

$$\frac{\partial \mathcal{L}}{\partial z_j} = \sum_{(q,d)} \sum_{w \in q} c(w,q)\, \omega_j \left(\frac{P_j(w \mid d)}{P(w \mid d)} - 1\right)$$

۳. اضافه کردن منظم‌سازی به گرادیان:

$$\nabla_{z_j} \mathcal{L}_{reg} = \frac{\partial \mathcal{L}}{\partial z_j} - \lambda z_j$$

۴. به‌روزرسانی پارامترهای آزاد z با یک گام گرادیان یا با Adam/L-BFGS:

$$z_j \leftarrow z_j + \eta \nabla_{z_j} \mathcal{L}_{reg}$$

توضیح ساده: اگر عبارت داخل جمع مثبت باشد، وزن مؤلفهٔ ز را بیشتر می‌کنیم؛ اگر منفی باشد، کمتر می‌کنیم.

حلقهٔ یادگیری کامل:

۱. مقداردهی اولیه: $z = (0,0,0,0) \rightarrow \omega_i = 0.25$

۲. تکرار تا همگرایی:

- Forward: محاسبهٔ $P_i(w \mid d)$ و $P(w \mid d; z)$، سپس محاسبهٔ $\mathcal{L}_{reg}(z)$.
- Backward: محاسبهٔ $\nabla_z \mathcal{L}_{reg}$ با فرمول بالا.
- Update: به‌روزرسانی z با نرخ یادگیری η (یا استفاده از Adam برای خودتنظیم نرخ).

۳. شرط توقف: تغییرات $\mathcal{L}_{reg}$ یا z بسیار کوچک شود یا تعداد تکرارها به حد مشخص برسد.

۴. خروجی: وزن‌ها $\alpha, \beta, \gamma, \theta = \text{softmax}(z)$ که به‌صورت خودکار یاد گرفته میشوند.

پاسخ سوال دوم

پاسخ بخش الف

جدول اسناد و امتیاز اولیه:

سند	محتوا	امتیاز اولیه
D_1	الگوریتم های یادگیری عمیق و شبکه های عصبی	12.5
D_2	یادگیری عمیق در بینایی ماشین	11.8
D_3	الگوریتم های ژنتیک و بهینه‌سازی	8.3
D_4	یادگیری تقویتی عمیق	10.2
D_5	معماری‌های عمیق در پردازش زبان	9.7

:E-STEP

$$Z_i \in \{1(background), 0(topic)\}$$

احتمال این که واژه از منبع عمومی باشه:

$$P^{(n)}(Z_i = 1 \mid w_i) = \frac{\lambda \cdot P(w_i \mid \mathrm{C})}{\lambda \cdot P(w_i \mid C) + (1-\lambda) \cdot P^{(n)}(w_i \mid \theta_F)}$$

LAMBDA: $\lambda = 0.3$

WORD: {عمیق,یادگیری,الگوریتم}

INITIAL FEEDBACK MODEL: $P^{(0)}(w \mid \theta_F)$ یکنواخت

$$P^{(0)}(w \mid \theta_F) = \frac{1}{3} \approx 0.333333$$

: $n = 0$

الگوریتم = W :

$$P^{(0)}(Z_i = 1 \mid \text{الگوریتم}) = \frac{0.3 \cdot P(\text{الگوریتم} \mid \mathrm{C})}{0.3 \cdot P(\text{الگوریتم} \mid C) + 0.7 \cdot P^{(0)}(\text{الگوریتم} \mid \theta_F)}$$

$$P^{(0)}(Z_i = 1 \mid \text{الگوریتم}) = \frac{0.3 \cdot 0.001}{0.3 \cdot 0.001 + 0.7 \cdot 0.334}$$

$$P^{(0)}(Z = 1 \mid \text{الگوریتم}) \approx \frac{0.0003}{0.2341} \approx 0.00128$$

یادگیری = W :

$$P^{(0)}(Z_i = 1 \mid \text{یادگیری}) = \frac{0.3 \cdot P(\text{یادگیری} \mid \mathrm{C})}{0.3 \cdot P(\text{یادگیری} \mid C) + 0.7 \cdot P^{(0)}(\text{یادگیری} \mid \theta_F)}$$

$$P^{(0)}(Z_i = 1 \mid \text{یادگیری}) = \frac{0.3 \cdot 0.002}{0.3 \cdot 0.002 + 0.7 \cdot 0.334}$$

$$P^{(0)}(Z = 1 \mid \text{یادگیری}) \approx \frac{0.0006}{0.2344} \approx 0.00256$$

عمیق = W :

$$P^{(0)}(Z_i = 1 \mid \text{عمیق}) = \frac{0.3 \cdot P(\text{عمیق} \mid \mathrm{C})}{0.3 \cdot P(\text{عمیق} \mid C) + 0.7 \cdot P^{(0)}(\text{عمیق} \mid \theta_F)}$$

$$P^{(0)}(Z_i = 1 \mid \text{عمیق}) = \frac{0.3 \cdot 0.0015}{0.3 \cdot 0.002 + 0.7 \cdot 0.334}$$

$$P^{(0)}(Z = 1 \mid \text{عمیق}) \approx \frac{0.00045}{0.23425} \approx 0.00192$$

:M-step

با استفاده از تخمین‌های مرحله‌ی قبل، مدل موضوعی رو به‌روزرسانی می‌کنیم:

$$P^{(n+1)}(w_i \mid \theta_F) = \frac{c(w_i, F) \cdot (1 - P^{(n)}(z_i = 1 \mid w_i))}{\sum_{w_j} c(w_j, F) \cdot (1 - P^{(n)}(z_j = 1 \mid w_j))}$$

که:

$$\sum_{w_j} c(w_j, F) \cdot \left(1 - P^{(n)}(z_j = 1 \mid w_j)\right) = c(\text{الگوریتم}, F) \cdot (1 - 0.00128) + c(\text{یادگیری}, F) \cdot (1 - 0.00256) + c(\text{عمیق}, F) \cdot (1 - 0.00192)$$

طبق جدول اسناد داریم:

$$\sum_{w_j} c(w_j, F) \cdot \left(1 - P^{(n)}(z_j = 1 \mid w_j)\right) = 2 \cdot (0.99872) + 3 \cdot (0.99744) + 4 \cdot (0.99808)$$

$$\sum_{w_j} c(w_j, F) \cdot \left(1 - P^{(n)}(z_j = 1 \mid w_j)\right) = 1.99744 + 2.99232 + 3.99232$$

$$\sum_{w_j} c(w_j, F) \cdot \left(1 - P^{(n)}(z_j = 1 \mid w_j)\right) = 8.98208$$

$$P^{(1)}(\text{الگوریتم} \mid \theta_F) = \frac{c(\text{الگوریتم}, F) \cdot (1 - P^{(0)}(z_i = 1 \mid \text{الگوریتم}))}{\sum_{w_j} c(w_j, F) \cdot (1 - P^{(0)}(z_j = 1 \mid w_j))}$$

$$P^{(1)}(\text{الگوریتم} \mid \theta_F) = \frac{1.99744}{8.98208} \approx 0.22238$$

$$P^{(1)}(\text{یادگیری} \mid \theta_F) = \frac{c(\text{یادگیری}, F) \cdot (1 - P^{(0)}(z_i = 1 \mid \text{یادگیری}))}{\sum_{w_j} c(w_j, F) \cdot (1 - P^{(0)}(z_j = 1 \mid w_j))}$$

$$P^{(1)}(\text{یادگیری} \mid \theta_F) = \frac{2.99232}{8.98208} \approx 0.33314$$

$$P^{(1)}(\text{عمیق} \mid \theta_F) = \frac{c(\text{عمیق}, F) \cdot (1 - P^{(0)}(z_i = 1 \mid \text{عمیق}))}{\sum_{w_j} c(w_j, F) \cdot (1 - P^{(0)}(z_j = 1 \mid w_j))}$$

$$P^{(1)}(\text{عمیق} \mid \theta_F) = \frac{3.99232}{8.98208} \approx 0.44448$$

برای مجموعه بازخورد F به‌صورت زیر تعریف می‌شود:

$$\log p(F^n \mid \theta) = \sum_i \sum_w c(w; d_i) \log \left[(1-\lambda)\, P^{(n)}(w \mid \theta) + \lambda\, p(w \mid C)\right]$$

طبق **فرمول MIXTURE MODEL FOR FEEDBACK** $P(w) = \lambda P(w \mid C) + (1-\lambda) P(w \mid \theta_F)$:

$$\log p(F^n \mid \theta) = \sum_i \sum_w c(w; d_i) \log\ P^{(n)}(w)$$

$$P^{(1)}(\text{الگوریتم}) = \lambda P(\text{الگوریتم} \mid C) + (1-\lambda) P^{(1)}(\text{الگوریتم} \mid \theta_F)$$

$$P^{(1)}(\text{الگوریتم}) = 0.3\,.\,0.001 + 0.7\,.\,0.22238 \;=\; 0.155966 \;\approx\; 0.16$$

$$P^{(1)}(\text{یادگیری}) = \lambda P(\text{یادگیری} \mid C) + (1-\lambda) P^{(1)}(\text{یادگیری} \mid \theta_F)$$

$$P^{(1)}(\text{یادگیری}) = 0.3\,.\,0.002 + 0.7\,.\,0.33314 \;=\; 0.233798 \;\approx\; 0.23$$

$$P^{(1)}(\text{عمیق}) = \lambda P(\text{عمیق} \mid C) + (1-\lambda) P^{(1)}(\text{عمیق} \mid \theta_F)$$

$$P^{(1)}(\text{عمیق}) = 0.3\,.\,0.0015 + 0.7\,.\,0.44448 \;=\; 0.311586 \;\approx\; 0.31$$

$$\begin{aligned}\log p(F^1 \mid \theta) = &\sum_w c(w; d_1) \log\ P^{(1)}(w) + \sum_w c(w; d_2) \log\ P^{(1)}(w) + \sum_w c(w; d_3) \log\ P^{(1)}(w)\\ &+ \sum_w c(w; d_4) \log\ P^{(1)}(w) + \sum_w c(w; d_5) \log\ P^{(1)}(w)\end{aligned}$$

$$\begin{aligned}\log p(F^1 \mid \theta) \;= &\sum_i c(\text{الگوریتم}; d_i) \log\ P^{(1)}(\text{الگوریتم})\\ &+ \sum_i c(\text{یادگیری}; d_i) \log\ P^{(1)}(\text{یادگیری})\\ &+ \sum_i c(\text{عمیق}; d_i) \log\ P^{(1)}(\text{عمیق})\end{aligned}$$

$$\begin{aligned}\log p(F^1 \mid \theta) \;= &\; c(\text{الگوریتم}, F)\ log\ P^{(1)}(\text{الگوریتم})\\ &+ c(\text{یادگیری}, F)\ log\ P^{(1)}(\text{یادگیری})\\ &+ c(\text{عمیق}, F)\ log\ P^{(1)}(\text{عمیق})\end{aligned}$$

$$\sum_{w} c(w;d_1)\log P^{(1)}(w) = c(\text{الگوریتم};d_1)\log P^{(1)}(\text{الگوریتم})$$
$$+\ c(\text{یادگیری};d_1)\log P^{(1)}(\text{یادگیری}) + c(\text{عمیق};d_1)\log P^{(1)}(\text{عمیق})$$

$$\sum_{w} c(w;d_1)\log P^{(1)}(w) = \log 0.16 + \log 0.23 + \log 0.31 = -1.943$$

$$\sum_{w} c(w;d_2)\log P^{(1)}(w) = 0 + \log 0.23 + \log 0.31 = -1.147$$

$$\sum_{w} c(w;d_3)\log P^{(1)}(w) = \log 0.16 + 0 + 0 = -0.796$$

$$\sum_{w} c(w;d_4)\log P^{(1)}(w) = 0 + \log 0.23 + \log 0.31 = -1.147$$

$$\sum_{w} c(w;d_5)\log P^{(1)}(w) = 0 + 0 + \log 0.31 = -0.509$$

$$log\, p(F^1 \mid \theta) = (-1.943) + (-1.147) + (-0.796) + (-1.147) + (-0.509)$$

$$\log p(F^1 \mid \theta) = -5.542$$

$$\log p(F^0 \mid \theta) = 9\log 0.334 = -4.286$$

نتیجه نهایی:

word	#	P(w|C)	Iteration 0		Iteration 1	
			$p(w \mid \theta)$	P(z=1)	$p(w \mid \theta)$	P(z=1)
الگوریتم	2	0.001	0.334	0.00128	0.22238	
یادگیری	3	0.002	0.334	0.00256	0.33314	
عمیق	4	0.0015	0.334	0.00192	0.44448	
Log-Likelihood			-4.286		-5.542	

پاسخ بخش ب

کدام کلمات وزن بیشتری می‌گیرند و چرا؟

در میان واژه‌های بررسی‌شده، کلمه «عمیق» بیشترین وزن را به دست آورده است. دلیل این موضوع این است که این واژه بیشترین تعداد تکرار را در اسناد بازخورد داشته و در محاسبات مرحله‌ی E نیز مقدار $1-P(Z=1 \mid w)$ برای آن بزرگتر بوده است، بنابراین سهم بیشتری در مدل موضوعی پیدا کرده است. پس از آن واژه «یادگیری» قرار می‌گیرد که با سه بار تکرار و مقدار نسبتاً بزرگ $1-P(Z=1 \mid w)$ وزن متوسطی گرفته است. در نهایت واژه «الگوریتم» کمترین وزن را دارد زیرا تنها دو بار در

اسناد دیده شده و احتمال تعلق آن به مدل زمینه بیشتر بوده است، در نتیجه سهم کمتری در مدل موضوعی دارد.

اگر λ افزایش یابد، تأثیر آن چیست؟

اگر مقدار λ افزایش یابد، وزن مدل زمینه بیشتر می‌شود و در نتیجه احتمال تعلق واژه‌ها به مدل زمینه افزایش پیدا می‌کند. این تغییر باعث می‌شود که مقدار

$1 - P(Z = 1 \mid w)$ کاهش می یابد و در مرحله‌ی M وزن واژه‌ها در مدل بازخورد کمتر می شود. پس ، هر چه λ بزرگتر باشد، مدل بازخورد محافظه‌کارتر عمل می‌کند و بیشتر به توزیع مجموعه‌ی زمینه نزدیک می‌شود. این یعنی واژه‌های عمومی‌تر تقویت می‌شوند و واژه‌های خاصتر که در اسناد بازخورد اهمیت دارند، ممکن است نادیده گرفته شوند.

آیا D_3 باعث نویز می‌شود؟ چگونه می‌توان آن را شناسایی کرد؟

سند D_3 فقط جمله‌ی «الگوریتم‌های ژنتیک و بهینه‌سازی» است، در عمل فقط کلمه‌ی «الگوریتم» رو تقویت می‌کنه و هیچ کمکی به واژه‌های اصلی موضوع یعنی «یادگیری» و «عمیق» نمی‌کنه. چون این دو واژه در اون سند وجود ندارن، سند از نظر محتوایی با موضوع اصلی یعنی «یادگیری عمیق» هم‌راستا نیست و به همین دلیل می‌تونه نویز باشه.

برای اینکه بفهمیم نویزه یا نه، می‌تونیم چند چیز رو نگاه کنیم: اول اینکه ببینیم چقدر با واژه‌های کلیدی موضوع هم‌پوشانی داره؛ اگر خیلی کم باشه، احتمال نویز بودن زیاده. دوم اینکه بررسی کنیم سهم این سند در محاسبات کلی چقدره؛ اگر تأثیرش روی احتمال‌ها و لاگ‌لایکلیهود خیلی کم یا حتی منفی باشه، نشونه‌ی نویزه. سوم اینکه امتیاز اولیه‌ی سند رو با بقیه مقایسه کنیم؛ اگر پایین‌تر باشه، باز هم نشونه‌ی کم‌ربط بودنشه.

پاسخ سوال سوم

پاسخ بخش الف

QUERY LIKELIHOOD MODEL (مدل احتمال پرس‌وجو):

این مدل فرض می‌کنه که هر سند یک مدل زبانی داره، و بررسی می‌کنه که احتمال تولید پرس‌وجو توسط مدل زبانی سند چقدر هست. یعنی:

$$P(Q \mid D) = \prod_{w \in Q} P(w \mid D)$$

که در اون $P(w \mid D)$ با استفاده از فراوانی واژه در سند و SMOOTH کردن (مثل JELINEK-MERCER) محاسبه می‌شه.

KL-DIVERGENCE MODEL (مدل واگرایی کولبک-لایبلر):

این مدل فاصله‌ی آماری بین توزیع واژه‌های پرس‌وجو و سند رو اندازه‌گیری می‌کنه. فرمول اصلی:

$$D_{KL}(P(Q) \mid\mid P(D)) = \sum_{w \in V} P(w \mid Q) \cdot \mathrm{LOG} \frac{P(w \mid Q)}{P(w \mid D)}$$

هرچی این فاصله کمتر باشه، سند به پرس‌وجو نزدیک‌تره و رتبه‌ی بالاتری می‌گیره.

در مدل‌های **شباهت توزیعی** مثل KL-DIVERGENCE و QUERY LIKELIHOOD ، هدف اینه که شباهت آماری بین توزیع واژه‌ها در پرس‌وجو و سند بررسی بشه نه صرفاً تطابق معنایی.

این شباهت باعث می‌شه سندهایی که از نظر آماری به پرس‌وجو نزدیک‌ترن، رتبه‌ی بالاتری بگیرن.

اما چون این مدل‌ها فقط به توزیع واژه‌ها نگاه می‌کنن، ممکنه اسناد بلندتر یا کلی‌تر که واژه‌های بیشتری دارن، امتیاز بیشتری بگیرن حتی اگر سند کوتاه‌تری وجود داشته باشه که دقیقاً به مفهوم پرس‌وجو اشاره کرده باشه.

- در **QUERY LIKELIHOOD** ، سند بلند احتمال بیشتری داره که واژه‌های پرس‌وجو رو شامل بشه، حتی اگر اون واژه‌ها در زمینه‌ی اصلی سند نباشن.
- در **KL-DIVERGENCE** ، سند بلندتر می‌تونه توزیع واژه‌ی متنوع‌تری داشته باشه که به توزیع پرس‌وجو نزدیک‌تر بشه، حتی اگر معنای دقیق نداشته باشه.
- **SMOOTH کردن** باعث می‌شه واژه‌های عمومی‌تر (از مدل زمینه‌ای) وارد محاسبه بشن و به نفع سندهای کلی‌تر تموم بشه.

پاسخ بخش ب

در مدل‌های آماری مثل **Query Likelihood** و **KL-Divergence**، هدف مقایسه‌ی توزیع واژه‌ها بین پرس‌وجو و سند هست. اما یه مشکل رایج اینه که بعضی واژه‌های پرس‌وجو ممکنه اصلاً در سند ظاهر نشده باشن، و این باعث می‌شه احتمال $P(w \mid D)$ صفر بشه. برای حل این مشکل، از **هموارسازی (Smoothing)** استفاده می‌کنیم.

شرایطی که هموارسازی باعث افت عملکرد می‌شود:

۱. وقتی سند کوتاه باشه:

- تعداد واژه‌ها کم‌تره، پس $P_{ML}(w \mid D)$ برای واژه‌های پرس‌وجو ممکنه صفر یا خیلی کم باشه.
- هموارسازی باعث می‌شه $P(w \mid C)$ وارد محاسبه بشه، که معمولاً واژه‌های عمومی‌تر رو تقویت می‌کنه.
- نتیجه: سند کوتاه که واژه‌های خاص داره، ممکنه امتیاز پایین‌تری بگیره چون مدل به واژه‌های عمومی‌تر وزن می‌ده.

۲. وقتی سند خاص باشه:

- اگر واژه‌های سند خیلی تخصصی باشن و در مجموعه کمتر دیده شده باشن، $P(w \mid C)$ برای اون‌ها کم می‌شه.

- هموارسازی باعث می‌شه این واژه‌های خاص ضعیف بشن و سند از نظر مدل کمتر مرتبط به نظر بیاد حتی اگر معنایی دقیق باشه.

۳. تنظیم نادرست پارامتر هموارسازی:

- اگر مقدار λ یا μ خیلی زیاد باشه، مدل بیشتر به مجموعه کلی تکیه می‌کنه و سندهای خاص رو نادیده می‌گیره.
- اگر خیلی کم باشه، مدل حساسیت زیادی به داده‌های سند پیدا می‌کنه و ممکنه ناپایدار بشه.

پاسخ بخش ج

Query Drift به پدیده‌ای گفته می‌شه که در اون توزیع زبانی پرس‌وجو به‌طور ناخواسته تغییر می‌کنه، طوری که تمرکز مدل بازیابی از موضوع اصلی پرس‌وجو منحرف می‌شه. این اتفاق معمولاً در فرآیند **بازخورد مرتبط (Relevance Feedback)** یا **گسترش پرس‌وجو (Query Expansion)** رخ می‌ده، وقتی واژه‌هایی به پرس‌وجو اضافه می‌شن که از نظر آماری رایج ولی از نظر معنایی بی‌ربط هستن.

در مدل‌های شباهت توزیعی، مثل **Query Likelihood**، شباهت بین پرس‌وجو و سند بر اساس احتمال تولید پرس‌وجو توسط مدل زبانی سند محاسبه می‌شه.

اگر واژه‌های عمومی مثل "data" ، "system"، "information"به پرس‌وجو اضافه بشن، چون این واژه‌ها در اکثر اسناد وجود دارن، مقدار $P(w \mid D)$ برای اون‌ها بالاست. در نتیجه، سندهایی که این واژه‌های عمومی رو دارن حتی اگر به موضوع اصلی پرس‌وجو ربطی نداشته باشن، امتیاز بیشتری می‌گیرن. این باعث می‌شه اسناد دقیق‌تر و تخصصی‌تر رتبه‌ی پایین‌تری بگیرن.

در مدل **KL-Divergence**هم همین اتفاق می‌افته. شباهت بین توزیع واژه‌های پرس‌وجو و سند با فرمول زیر سنجیده می‌شه:

$$D_{KL}(P(Q) \,||\, P(D)) = \sum_{w \in V} P(w \mid Q) \cdot \log \frac{P(w \mid Q)}{P(w \mid D)}$$

اگر توزیع $P(Q)$ به سمت واژه‌های عمومی متمایل بشه، فاصله KL با سندهایی که اون واژه‌ها رو دارن کمتر می‌شه، حتی اگر اون سندها به موضوع اصلی پرس‌وجو بی‌ربط باشن. این یعنی رتبه‌بندی به نفع اسناد عمومی‌تر تغییر می‌کنه.

۳ روش برای جلوگیری:

۱. مدل ترکیبی با کنترل نویز(Generative Mixture Model)

در این مدل، واژه‌ها از دو منبع تولید می‌شن: مدل زمینه‌ای $P(w \mid C)$و مدل موضوعی $P(w \mid \theta)$. با تنظیم مناسب پارامتر λ، می‌تونیم تأثیر واژه‌های عمومی رو کاهش بدیم:

$$P(w) = (1-\lambda) \cdot P(w \mid \theta) + \lambda \cdot P(w \mid C)$$

هرچی λ کمتر باشه، تمرکز مدل روی واژه‌های موضوعی بیشتر می‌شه و Drift کمتر اتفاق می‌افته.

۲. فیلتر واژه‌های عمومی با وزن‌دهی معکوس(IDF Filtering)

قبل از اعمال مدل، می‌تونیم واژه‌هایی که در مجموعه زیاد دیده می‌شن (دارای IDF پایین) رو حذف یا وزن‌شون رو کم کنیم. این باعث می‌شه واژه‌های خاص و مهم‌تر تأثیر بیشتری در رتبه‌بندی داشته باشن و Drift کنترل بشه.

۳. Query Term Weighting (وزن‌دهی پویا به واژه‌های پرس‌وجو)

به جای اینکه همه‌ی واژه‌های پرس‌وجو وزن برابر داشته باشن، می‌تونیم وزن واژه‌ها رو بر اساس اهمیت آماری یا معنایی تنظیم کنیم. واژه‌های عمومی وزن کمتر می‌گیرن و واژه‌های خاص وزن بیشتر. این کار باعث می‌شه Drift کنترل بشه چون واژه‌های عمومی نمی‌تونن توزیع پرس‌وجو رو منحرف کنن.

پاسخ بخش د

Distribution Shift به وضعیتی گفته می‌شه که در اون توزیع آماری داده‌هایی که سیستم بازیابی باهاش مواجه می‌شه، با توزیعی که مدل زبانی بر اساس اون آموزش دیده متفاوت باشه. این تغییر می‌تونه ناشی از موارد زیر باشه:

- ورود واژه‌ها یا اصطلاحات جدید (مثلاً "ChatGPT" یا "LLM")
- تغییر سبک نگارش (مثلاً از رسمی به محاوره‌ای)
- تغییر موضوعات رایج (مثلاً از پزشکی به هوش مصنوعی)
- تغییر زبان یا ترکیب زبان‌ها (مثلاً پرس‌وجوی انگلیسی در مجموعه فارسی)

در بازیابی اطلاعات، این یعنی پرس‌وجوها یا اسناد جدید ممکنه شامل واژه‌ها، ساختارها یا مفاهیمی باشن که مدل قبلاً ندیده یا به‌ندرت دیده. در نتیجه، مدل نمی‌تونه به‌درستی احتمال واژه‌ها رو تخمین بزنه یا شباهت معنایی رو تشخیص بده.

مدل‌های زبانی مثل Query Likelihood یا KL-Divergence برای تخمین $P(w \mid D)$ یا $P(w \mid C)$ به داده‌های قبلی وابسته‌ان. وقتی واژه‌ای جدید وارد می‌شه:

- **در مدل‌های کلاسیک**: احتمال واژه صفر می‌شه یا باید با smoothing تخمین زده بشه، که معمولاً مقدار خیلی کمی می‌ده.
- **در مدل‌های یادگیری عمیق** embedding : واژه ممکنه وجود نداشته باشه یا ضعیف باشه، و مدل نمی‌تونه ارتباط معنایی رو درک کنه.

نتیجه: سندهایی که واژه‌های جدید دارن، حتی اگر مرتبط باشن، ممکنه رتبه‌ی پایینی بگیرن چون مدل نمی‌تونه شباهت آماری یا معنایی رو درست تشخیص بده.

دو راهکار مهندسی یا طراحی برای افزایش پایداری در برابرDistribution Shift

۱. استفاده از مدل‌های تطبیقی(Adaptive Language Models)

مدل‌های تطبیقی به‌طور مداوم خودشون رو با داده‌های جدید هماهنگ می‌کنن. دو روش مهم:

Fine-tuning: مدل زبانی به‌صورت منظم با داده‌های جدید (مثلاً اسناد تازه‌وارد یا پرس‌وجوهای اخیر) آموزش داده می‌شه. این باعث می‌شه مدل واژه‌ها و ساختارهای جدید رو یاد بگیره.

Continual Learning: مدل به‌جای آموزش مجدد کامل، به‌صورت افزایشی یاد می‌گیره. یعنی دانش قبلی حفظ می‌شه و داده‌های جدید بهش اضافه می‌شن. این روش برای محیط‌هایی که داده‌ها دائماً تغییر می‌کنن خیلی مؤثره.

مثال :اگر در یک موتور جستجو، کاربران شروع به استفاده از واژه‌ی جدیدی مثل "LLM" کنن، مدل تطبیقی می‌تونه با استفاده از پرس‌وجوهای اخیر و اسناد مرتبط، معنای این واژه رو یاد بگیره و در رتبه‌بندی لحاظ کنه.

۲. افزایش پوشش واژگانی با استفاده از مدل‌های زمینه‌ای (Contextual Embedding)

مدل‌هایی مثل BERT، RoBERTa ، یا T5 به جای اینکه برای هر واژه یک embedding ثابت بسازن، بردار معنایی واژه رو بر اساس جمله یا زمینه تولید می‌کنن. این باعث می‌شه حتی واژه‌های جدید یا نادر، در زمینه‌ی مناسب معنا پیدا کنن.

مثال :واژه‌ی "prompt" ممکنه در زمینه‌ی "writing prompt" معنای آموزشی داشته باشه، ولی در جمله‌ی"prompt engineering in LLMs" معنای فنی پیدا کنه. مدل‌های زمینه‌ای می‌تونن این تفاوت رو تشخیص بدن حتی اگر واژه قبلاً دیده نشده باشه.