Търсене и извличане на информация. Приложение на дълбоко машинно обучение

Стоян Михов

Лекция 4: Ранкирано търсене на документи. Езиков модел.

План на лекцията

1. **Формалности за курса (2 мин)**
2. Ранкирано търсене на информация (10 мин)
3. Документно представяне чрез вектори от тегла $\mathrm{tf}{\cdot}\mathrm{idf}$ (15 мин)
4. Ранкиране при документно представяне чрез вектори от тегла (15 мин)
5. Езикови модели (15 мин)
6. k-грамни езикови модели и изглаждане на езиков модел (15 мин)
7. Ранкиране чрез документни езикови модели (15 мин)

Формалности

- От днес лекциите ще се провеждат в зала 500
- Четвъртата лекция се базира на глави 6 и 12 от първия учебник и глава 9 от втория учебник.

План на лекцията

1. Формалности за курса (5 мин)
2. **Ранкирано търсене на информация (10 мин)**
3. Документно представяне чрез вектори от тегла $\mathrm{tf}\cdot\mathrm{idf}$ (15 мин)
4. Ранкиране при документно представяне чрез вектори от тегла (15 мин)
5. Езикови модели (15 мин)
6. k-грамни езикови модели и изглаждане на езиков модел (15 мин)
7. Ранкиране чрез документни езикови модели (15 мин)

Ранкирано търсене на информация

- В големи колекции от документи често резултатът от булевото търсене връща хиляди документи, отговарящи на заявката, или нито един.
- За удовлетворяване на информационната потребност най-често е достатъчен само един или няколко от документите.
- Проблемът се състои в намирането и извеждането само на най-релевантните документи по отношение на информационната потребност.
- Задачата се свежда до извеждането само на първите k документа подредени по ранк, който следва да отразява релевантността по отношение на информационната потребност.
- Основната цел при ранкираното търсене е да се спести времето на потребителя за задоволяването на неговата информацинна потребност.

Класически подход към задачата за ранкирано търсене

- Заявката е текст — въпрос, изречение или списък от ключови думи — свързани с информационната потребност.

- Извлича се списък от (всички) документи, които включват (един или повече) от съществените термове от заявката.

- За всеки от документите от извлечения списък се изчислява ранк спрямо заявката.

- Извеждат се най-високо ранкираните k документа от списъка.

- Ключовият проблем е реализирането на релевантна ранкираща функция.

Подходи за ранкиране

- **Базирано на зоните на документа:**
  - идея: ако повече термове от заявката се срещат в заглавието или резюмето на документа, то документът е по-релевантен (разглежда се в глава 6 от първия учебник).
- **Евристичен подход:**
  - ако в документа има повече броя срещания на термове от заявката и тези термове са по специфични, то документът е по-релевантен (ще разгледаме по-подробно).
- **Вероятностен модел за релевантността:**
  - (разглежда се в глава 11 от първия учебник).
- **Езиков модел:**
  - документите се ранкират по вероятността съответният езиков модел на документа да генерира заявката (ще разгледаме по-подробно).

План на лекцията

1. Формалности за курса (5 мин)
2. Ранкирано търсене на информация (10 мин)
3. **Документно представяне чрез вектори от тегла** $\mathrm{tf}\cdot\mathrm{idf}$ **(15 мин)**
4. Ранкиране при документно представяне чрез вектори от тегла (15 мин)
5. Езикови модели (15 мин)
6. k-грамни езикови модели и изглаждане на езиков модел (15 мин)
7. Ранкиране чрез документни езикови модели (15 мин)

Недостатъци на документно представяне в $\{0,1\}^M$ и $\mathbb{N}^M$

- При използването на биномен или мултиномен документен модел векторите, съответстващи на документите, отразяват наличието или броя на срещанията на термове.

- Тези представяния водят до следните недостатъци:

  - Не се отчита специфичността на съответните термове. Стоп думи и термини се третират по еднакъв начин.

  - Броят на срещанията расте линейно (при мултиномен модел), докато човешките сензорни възприятия са логаритмични.

  - Бройките са абсолютни и не зависят от дължината на документите.

Тегло на срещанията

- Дефинираме $\mathrm{tf}_{t,d}$ (term frequency), като броя на срещанията на терма t в документа d.
- Ако даден терм от заявката се среща 10 пъти в документа, то това не означава, че документът е 10 пъти по релевантен. Затова дефинираме теглото на срещанията $\mathrm{w}_{t,d}$ логаритмично:
- $$\mathrm{w}_{t,d} = \begin{cases} 1 + \log \mathrm{tf}_{t,d} & \text{if } \mathrm{tf}_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$$

Тегло на терм

- По-редките термове са по-специфични и носят повече информация.
- Релевантно е колкото по рядко се среща даден терм, толкова по-високо тегло да има. Освен това е по-релевантно да се разглежда не броят на срещанията, а броят на документите, в които се среща даденият терм.
- Нека df_t (document frequency) е броят на документите, в които се среща термът t. Дефинираме обратната документна честота idf_t като:
- $$\mathrm{idf}_t = \log \frac{N}{\mathrm{df}_t}$$

Тегло $\mathrm{tf}\cdot\mathrm{idf}$

- Дефинираме теглото $\mathrm{tf}\cdot\mathrm{idf}$ като произведението на теглото на срещанията с теглото на терма:

$$\mathrm{tf}\cdot\mathrm{idf}_{t,d} = \log(1 + \mathrm{tf}_{t,d}) \times \log \frac{N}{\mathrm{df}_t}$$

- Това е най-известното тегло за евристично ранкиране в търсенето на информация. Често се изписва като tf-idf или $\mathrm{tf} \times \mathrm{idf}$.

Документно представяне чрез вектори от тегла

- На всеки документ съпоставяме вектор от $M = |V|$ тегла $\mathrm{tf}\cdot\mathrm{idf}$. Тогава $d \in \mathbb{R}^M$.

- Векторите са много разредени — повечето елементи са 0.

- Ключова идея:

  - Семантичната близост на два документа свеждаме до близост между съответните им вектори.

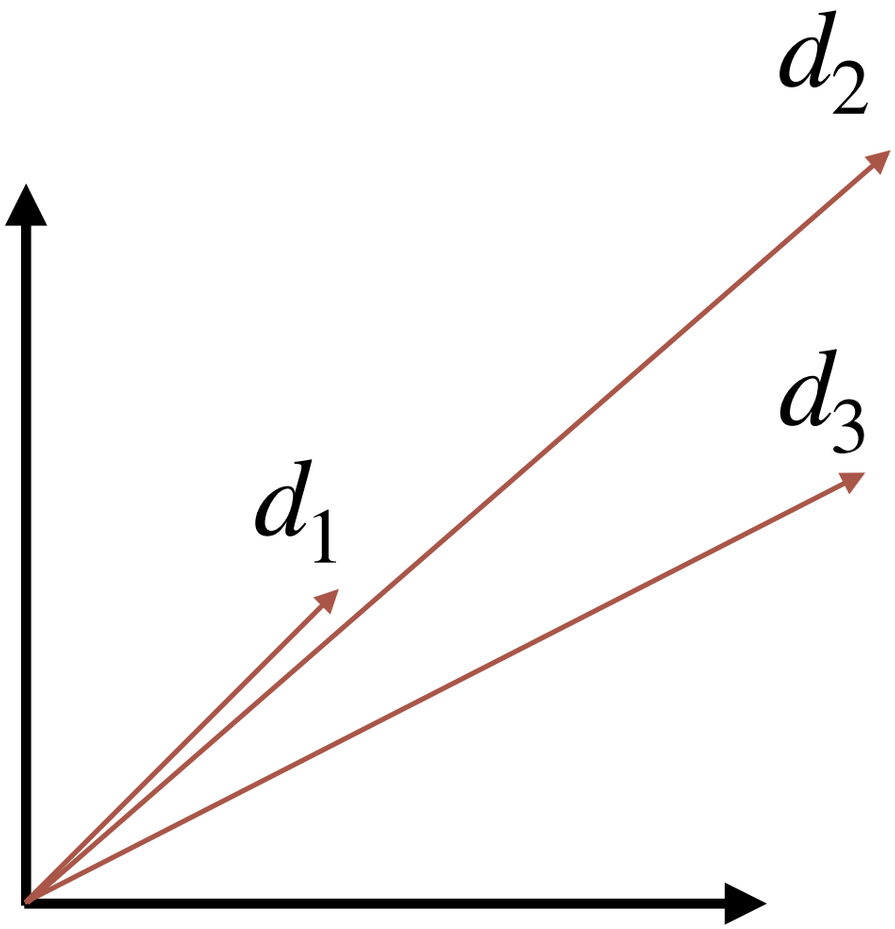
  - Ранкираме документите в зависимост от близостта им до вектора, представящ заявката.

План на лекцията

1. Формалности за курса (5 мин)
2. Ранкирано търсене на информация (10 мин)
3. Документно представяне чрез вектори от тегла $\mathrm{tf}\cdot\mathrm{idf}$ (15 мин)
4. **Ранкиране при документно представяне чрез вектори от тегла (15 мин)**
5. Езикови модели (15 мин)
6. k-грамни езикови модели и изглаждане на езиков модел (15 мин)
7. Ранкиране чрез документни езикови модели (15 мин)

Проблеми при Евклидово разстояние

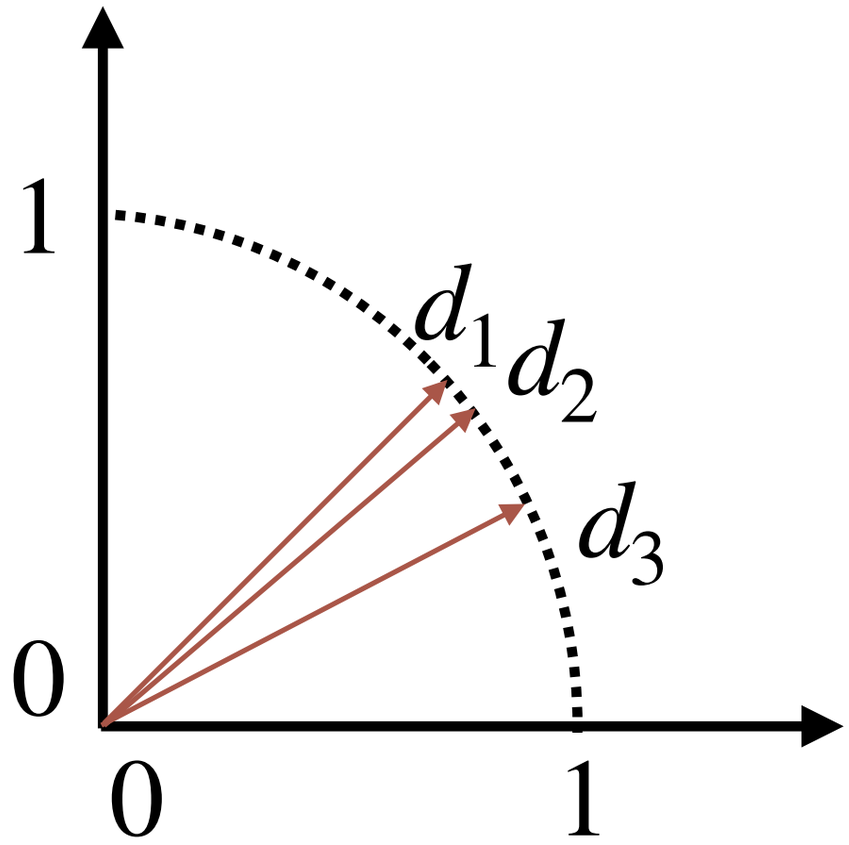
- Разстоянието зависи от броя на думите в документите.
- Семантично по-релевантно е да се използва за близост ъгълът между документите
- Малкият ъгъл между два документа съответства на близко честотно разпределение на съществените термове в двата документа.



Косинусова близост

- В интервала $\left[0, \frac{\pi}{2}\right]$ функцията косинус е монотонно намаляваща
- Вместо ъгъла е изчислително по-удобно да намираме косинуса между векторите:

$$\cos(\vec{q}, \vec{d}) = \frac{\vec{q} \,.\, \vec{d}}{|\vec{q}||\vec{d}|} = \frac{\sum_{i=1}^{M} q_i d_i}{\sqrt{\sum_{i=1}^{M} q_i^2} \sqrt{\sum_{i=1}^{M} d_i^2}}$$



- Алтернативно, ако документите са нормализирани, така че всички вектори да са с дължина 1, то косинусът е равен на декартовото произведение между двата вектора:

$$\cos(\vec{q}, \vec{d}) = \sum_{i=1}^{M} q_i d_i, \text{ ако } |\vec{q}| = |\vec{d}| = 1$$

Алгоритъм за ранкирано търсене

```
CosineScore(q)
1  float Scores[N] = 0
2  Initialize Length[N]
3  for each query term t do
4     calculate w_{t,q} and fetch postings list for t
5     for each pair(d, tf_{t,d}) in postings list do
6        Scores[d] += wf_{t,d} × w_{t,q}
7  Read the array Length[d]
8  for each d do
9     Scores[d] = Scores[d]/Length[d]
10 return Top K components of Scores[]
```

Забележки по ефективността

```
CosineScore(q)
1   float Scores[N] = 0
2   Initialize Length[N]
3   for each query term t do
4       calculate w_{t,q} and fetch postings list for t
5       for each pair(d, tf_{t,d}) in postings list do
6           Scores[d] += wf_{t,d} × w_{t,q}
7   Read the array Length[d]
8   for each d do
9       Scores[d] = Scores[d]/Length[d]
10  return Top K components of Scores[]
```

- Може да съхраняваме нормализирани тегла и да отпадне деленето на дължината.
- Вместо тегла (с плаваща запетая) може да съхраняваме брой срещания, за да пестим памет.
- Обратната документна честота можем да съхраняваме в речника за термовете.
- Най-високо ранкираните K документа можем да получим ефективно с приоритетна опашка (разглежда се в курса БАСД).

Варианти на теглото tf·idf

term frequency		document frequency		normalization	
n (natural)	$\mathrm{tf}_{t,d}$	n (no)	1	n (none)	1
l (logarithm)	$1+\log(\mathrm{tf}_{t,d})$	t (idf)	$\log \frac{N}{\mathrm{df}_t}$	c (cosine)	$\frac{1}{\sqrt{w_1^2+w_2^2+\ldots+w_M^2}}$
a (augmented)	$0.5+\frac{0.5\times \mathrm{tf}_{t,d}}{\max_t(\mathrm{tf}_{t,d})}$	p (prob idf)	$\max\{0, \log \frac{N-\mathrm{df}_t}{\mathrm{df}_t}\}$	u (pivoted unique)	$1/u$
b (boolean)	$\begin{cases}1 \text{ if } \mathrm{tf}_{t,d}>0\\0 \text{ otherwise}\end{cases}$			b (byte size)	$1/CharLength^{\alpha}$, $\alpha<1$
L (log ave)	$\frac{1+\log(\mathrm{tf}_{t,d})}{1+\log(\mathrm{ave}_{t\epsilon d}(\mathrm{tf}_{t,d}))}$				

ltc

- Заявката и документите може да имат различни схеми за претегляне
- Нотацията SMART — ddd.qqq — например ltc.lnn

Пример за евристично ранкиране

Терм	Заявка				Документ		Произведение
	tf	df	idf	$w_{t,q}$	tf	$w_{t,d}$	
автомобил	0	5000	2.3	0	1	0.41	0
добра	1	50000	1.3	1.3	0	0	0
застраховка	1	1000	3.0	3.0	2	0.82	2.46
кола	1	10000	2.0	2.0	1	0.41	0.82

- Пример за ранкиране при заявка:
  “добра застраховка кола” в колекция от N=1000000 документа.

- Използва се SMART схема **nnc.btn**

- При даден документ с две срещания на термовете “застраховка” и по едно срещане на “кола” и “ автомобил” се получава ранк:

$$\text{score}(q,d) = 0 \times 0.41 + 1.3 \times 0 + 3.0 \times 0.82 + 2.0 \times 0.41 = 3.28$$

План на лекцията

1. Формалности за курса (5 мин)
2. Ранкирано търсене на информация (10 мин)
3. Документно представяне чрез вектори от тегла $\mathrm{tf}{\cdot}\mathrm{idf}$ (15 мин)
4. Ранкиране при документно представяне чрез вектори от тегла (15 мин)
5. **Езикови модели (15 мин)**
6. k-грамни езикови модели и изглаждане на езиков модел (15 мин)
7. Ранкиране чрез документни езикови модели (15 мин)

Моделиране на езика

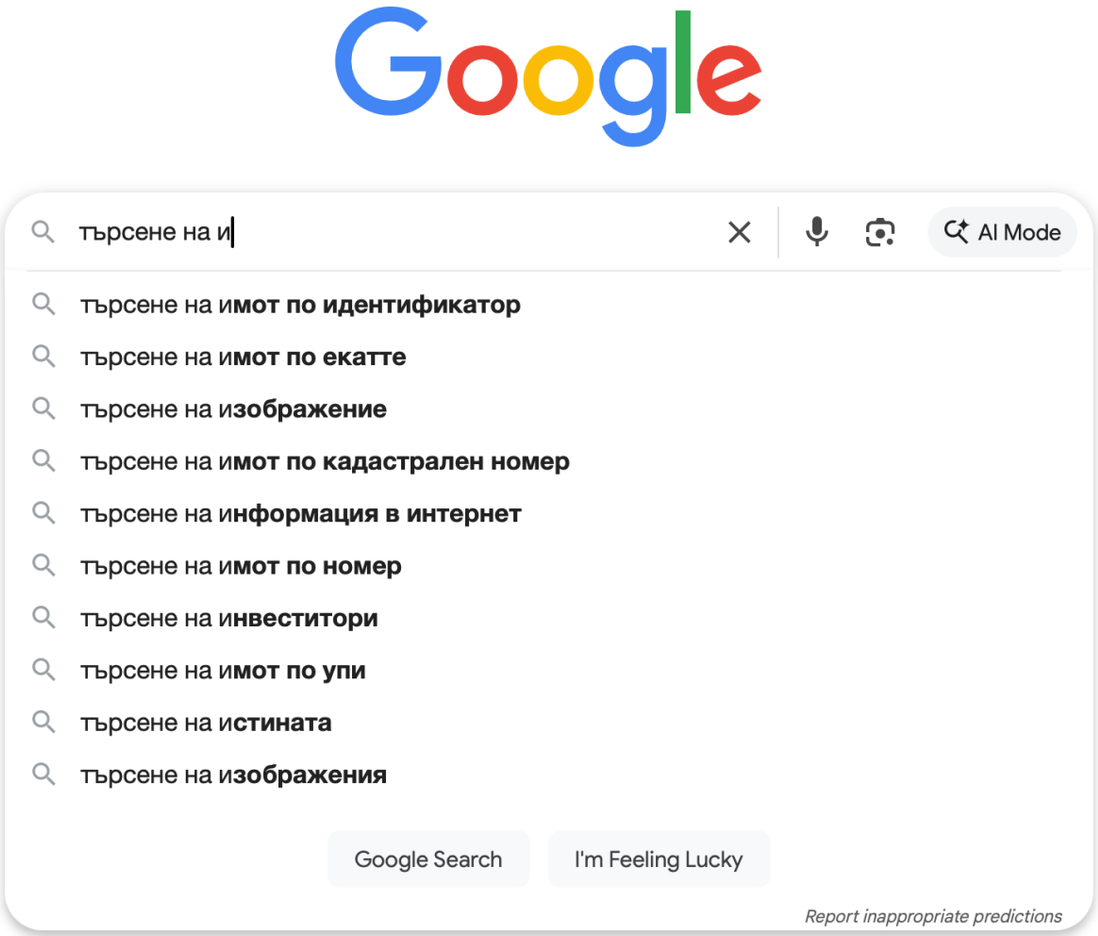
- Моделиране на езика е задачата да се определи вероятност на изреченията от езика, която да отразява вероятността да наблюдаваме даденото изречение.
    - Например, каква е вероятността да наблюдаваме изречението "*Черното куче подгони котката*"?
- **<u>Дефиниция:</u>** Нека е даден краен речник (азбука) от символи V. Тогава **езиков модел** наричаме дискретно (безкрайно) разпределение върху всички крайни последователности от символи. В такъв случай вероятностното пространство е $\Omega = V^*$. Вероятността на дадена последователност от символи $\alpha = x_1 x_2 \ldots x_n \in V^*$ означаваме с $\Pr[\alpha],\ \Pr[x_1 x_2 \ldots x_n]$.
- Следователно $\sum_{\alpha \in V^*} \Pr[\alpha] = 1.$
- **Забележка:** В зависимост от приложението символите от V могат да бъдат букви, срички, думи в естествен език или произволни други обекти, стига V да е крайно.
- **Проблем:** Как да моделираме **безрайно** разпределение?

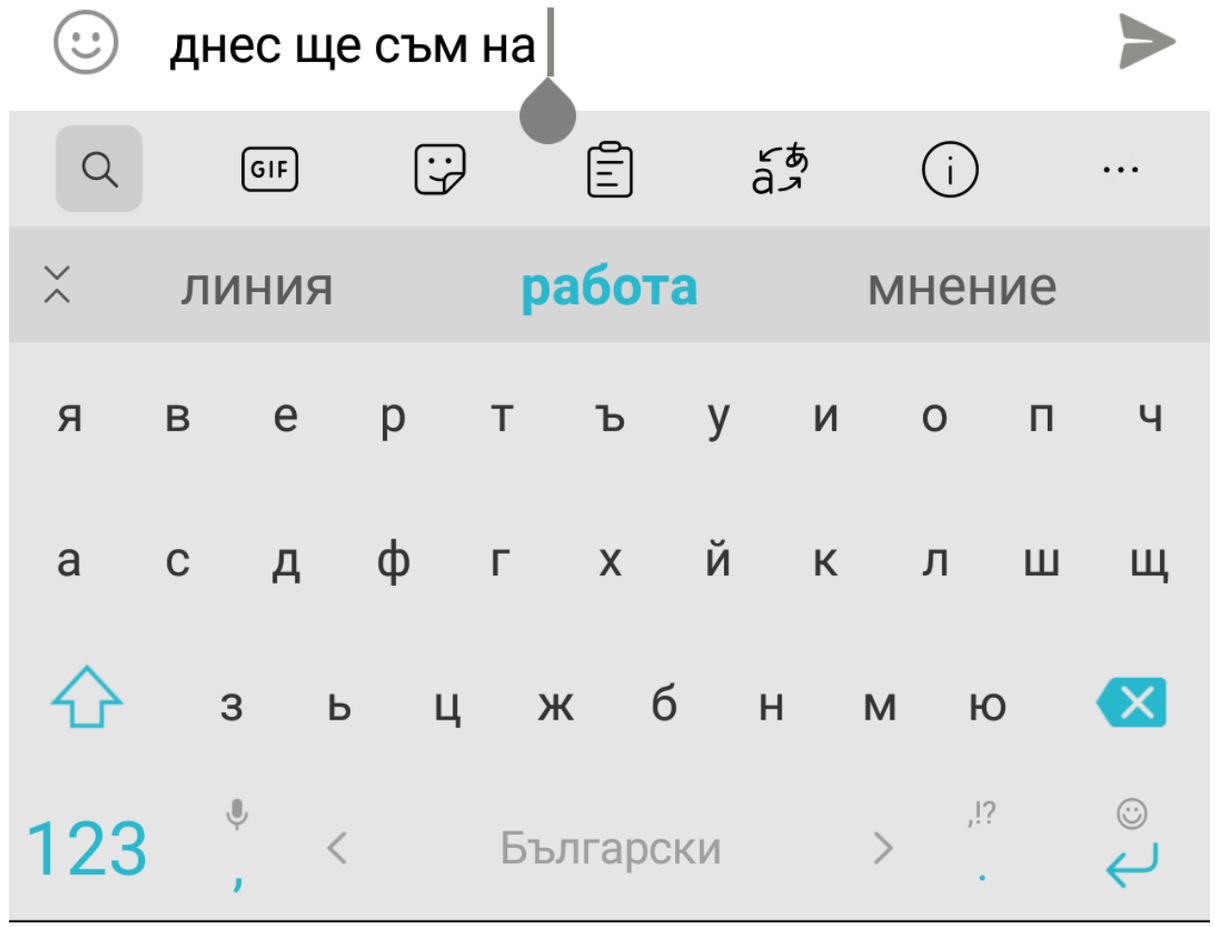
Моделиране на езика с локални разпределения

- Ще покажем, че всеки езиков модел може да се представи с фамилия от локални разпределения — разпределение върху символите от речника да следват след дадена начална последователност.
  - Например, каква е вероятността да наблюдаваме думата “*самолет*” след начало “*Черното куче подгони*”?

- За целта ще разгледаме условната вероятности символа x_{n+1} да следва началната последователност от символи $x_1 x_2 \ldots x_n$ — означаваме по-нататък с $\Pr[x_{n+1} \mid x_1 x_2 \ldots x_n]$.

Приложения на локално разпределение на езиков модел





Локални разпределения на езиков модел

- Нека е даден езиков модел, дефиниращ за последователност $\alpha \in V^*$ вероятност $\Pr[\alpha]$ и $\sum_{\alpha \in V^*} \Pr[\alpha] = 1$.

- Нека разгледаме вероятностното събитие A дадена последователност да има начало $x_1 x_2 \ldots x_n$. Тогава $A = \bigcup_{\alpha \in V^*} \{x_1 x_2 \ldots x_n \alpha\}$ и ще означаваме $A = x_1 x_2 \ldots x_n V^*$. В такъв случай получаваме
$$\Pr[A] = \Pr[x_1 x_2 \ldots x_n V^*] = \sum_{\alpha \in V^*} \Pr[x_1 x_2 \ldots x_n \alpha].$$

- Нека разгледаме вероятностното събитие B дадена последователност да има начало $x_1 x_2 \ldots x_n x_{n+1}$. Тогава вероятността $\Pr[x_{n+1} \mid x_1 x_2 \ldots x_n]$ т.е. x_{n+1} да следва началната последователност $x_1 x_2 \ldots x_n$ всъщност е $\Pr[B \mid A]$. Следователно:
$$\Pr[x_{n+1} \mid x_1 x_2 \ldots x_n] := \Pr[B \mid A] = \frac{\Pr[B \cap A]}{\Pr[A]} = \frac{\Pr[B]}{\Pr[A]} = \frac{\Pr[x_1 x_2 \ldots x_n x_{n+1} V^*]}{\Pr[x_1 x_2 \ldots x_n V^*]}.$$

- Нека разгледаме вероятностното събитие при условие, че последователността има начало $x_1 x_2 \ldots x_n$, то тя да свърши. Това събитие ще означаваме с $\Pr[\$ \mid x_1 x_2 \ldots x_n]$, където $\$$ е нов символ, така че $\$ \notin V$. Тогава:
$$\Pr[\$ \mid x_1 x_2 \ldots x_n] := \frac{\Pr[x_1 x_2 \ldots x_n]}{\Pr[x_1 x_2 \ldots x_n V^*]}.$$

Локални разпределения на езиков модел

- **Свойство:** Нека $V^{\$} = V \cup \{\$\}$ и $x_1x_2\ldots x_n \in V^*$ е начална последователност. Тогава $\Pr[x \mid x_1x_2\ldots x_n]$ за $x \in V^{\$}$ дефинира вероятностно разпределение върху $V^{\$}$.

- **Доказателство:** $x_1x_2\ldots x_nV^* = \left(\bigcup_{x\in V} x_1x_2\ldots x_nxV^*\right) \cup \{x_1x_2\ldots x_n\}$. Следователно:

$$\sum_{x\in V^{\$}} \Pr[x \mid x_1x_2\ldots x_n] = \frac{\sum_{x\in V} \Pr[x_1x_2\ldots x_nxV^*] + \Pr[x_1x_2\ldots x_n]}{\Pr[x_1x_2\ldots x_nV^*]} = \frac{\Pr[x_1x_2\ldots x_nV^*]}{\Pr[x_1x_2\ldots x_nV^*]} = 1$$

- **Дефиниция:** Разпределението $\Pr[x \mid x_1x_2\ldots x_n]$ за $x \in V^{\$}$ наричаме **локално разпределение на езиковия модел** за началния контекст $x_1x_2\ldots x_n$.

- **Свойство:**

$$\begin{aligned}
\Pr[x_1x_2\ldots x_n] &= \Pr[x_1x_2\ldots x_nV^*] \Pr[\$ \mid x_1x_2\ldots x_n] = \\
&= \Pr[x_1x_2\ldots x_{n-1}V^*] \Pr[x_n \mid x_1x_2\ldots x_{n-1}] \Pr[\$ \mid x_1x_2\ldots x_n] = \\
&= \Pr[x_1x_2\ldots x_{n-2}V^*] \Pr[x_{n-1} \mid x_1x_2\ldots x_{n-2}] \Pr[x_n \mid x_1x_2\ldots x_{n-1}] \Pr[\$ \mid x_1x_2\ldots x_n] = \\
&\vdots \\
&= \Pr[V^*] \Pr[x_1 \mid \varepsilon] \Pr[x_2 \mid x_1] \ldots \Pr[x_{n-1} \mid x_1x_2\ldots x_{n-2}] \Pr[x_n \mid x_1x_2\ldots x_{n-1}] \Pr[\$ \mid x_1x_2\ldots x_n] = \\
&= \Pr[x_1 \mid \varepsilon] \Pr[x_2 \mid x_1] \ldots \Pr[x_{n-1} \mid x_1x_2\ldots x_{n-2}] \Pr[x_n \mid x_1x_2\ldots x_{n-1}] \Pr[\$ \mid x_1x_2\ldots x_n] =
\end{aligned}$$

- **Извод:** Всеки езиков модел дефинира фамилия от контекстни локални разпределения $\{\Pr[x \mid x_1x_2\ldots x_n]\}_{x_1x_2\ldots x_n\in V^*}$.

От фамилия локални разпределения към езиков модел

- **Дефиниция:** Фамилията от контекстни локални разпределения $\{\Pr[x \mid x_1 x_2 \ldots x_n]\}_{x_1 x_2 \ldots x_n \in V^*}$ наричаме **точна** (tight) ако е в сила:

$$\sum_{x_1 x_2 \ldots x_n \in V^*} \Pr[x_1 \mid \varepsilon] \Pr[x_2 \mid x_1] \ldots \Pr[x_n \mid x_1 x_2 \ldots x_{n-1}] \Pr[\$ \mid x_1 x_2 \ldots x_n] = 1.$$

- **Свойство:** Всяка точна фамилия от контекстни локални разпределения дефинира езиков модел върху V^* като:

$$\Pr[x_1 x_2 \ldots x_n] = \Pr[x_1 \mid \varepsilon] \Pr[x_2 \mid x_1] \ldots \Pr[x_n \mid x_1 x_2 \ldots x_{n-1}] \Pr[\$ \mid x_1 x_2 \ldots x_n].$$

- **Задача**:
  а) Да се докаже, че ако за фамилията от контекстни локални разпределения $\{\Pr[x \mid x_1 x_2 \ldots x_n]\}_{x_1 x_2 \ldots x_n \in V^*}$ е дадено фиксирано $\delta \in (0,1)$, така че за всяка последователност $x_1 x_2 \ldots x_n \in V^*$ е в сила $\Pr[\$ \mid x_1 x_2 \ldots x_n] = \delta$, то тя е точна.
  б) ** Покажете, че съществува фамилия от контекстни локални разпределения $\{\Pr[x \mid x_1 x_2 \ldots x_n]\}_{x_1 x_2 \ldots x_n \in V^*}$, такава че за всяка последователност $x_1 x_2 \ldots x_n \in V^*$ е в сила $\Pr[\$ \mid x_1 x_2 \ldots x_n] > 0$ и не е точна.

Използване на еквивалентното представяне

- По-нататък често под езиков модел ще разбираме система (функция, алгоритъм, метод), която ни представя точна фамилия от контекстни локални разпределения $\{\Pr[x \mid x_1 x_2 \ldots x_n]\}_{x_1 x_2 \ldots x_n \in V^*}$.

- Т.е. за всяка начална последователност $x_1 x_2 \ldots x_n \in V^*$ системата ни връща вероятностно разпределение дефинирано върху $V^\$$, отразяващо условната вероятност $\Pr[x \mid x_1 x_2 \ldots x_n]$ за всяко $x \in V^\$$.

План на лекцията

1. Формалности за курса (5 мин)
2. Ранкирано търсене на информация (10 мин)
3. Документно представяне чрез вектори от тегла $\mathbf{tf{\cdot}idf}$ (15 мин)
4. Ранкиране при документно представяне чрез вектори от тегла (15 мин)
5. Езикови модели (15 мин)
6. **k-грамни езикови модели и изглаждане на езиков модел (15 мин)**
7. Ранкиране чрез документни езикови модели (15 мин)

Марковско свойство

- Марковско свойство от ред k:
  Вероятността за следващата дума зависи само от предишните най-много $k-1$ думи. Т.е. за всяко $x \in V^{\$}$, всеки $x_1, x_2, \ldots, x_{k-1} \in V$ и всяко начало $\alpha \in V^*$ е изпълнено $\Pr[x \mid \alpha x_1 x_2 \ldots x_{k-1}] = \Pr[x \mid x_1 x_2 \ldots x_{k-1}]$.

- Например, ако приемем марковско свойство от ред 3, получаваме:

$$\Pr[x_1 x_2 \ldots x_n] =$$
$$= \Pr[x_1 \mid \varepsilon] \Pr[x_2 \mid x_1] \Pr[x_3 \mid x_1 x_2] \Pr[x_4 \mid x_1 x_2 x_3] \ldots \Pr[x_n \mid x_1 x_2 \ldots x_{n-1}] \Pr[\$ \mid x_1 x_2 \ldots x_n] =$$
$$= \Pr[x_1 \mid \varepsilon] \Pr[x_2 \mid x_1] \Pr[x_3 \mid x_1 x_2] \Pr[x_4 \mid x_2 x_3] \ldots \Pr[x_n \mid x_{n-2} x_{n-1}] \Pr[\$ \mid x_{n-1} x_n]$$

Марковски езиков модел

- Марковски езиков модел от ред k наричаме езиков модел, в който е сила Марковското свойство от ред k.

- Марковският езиков модел от ред k се определя от условните вероятности $\Pr[x \mid x_1 x_2 \ldots x_m]$ за всички последователности $x_1 x_2 \ldots x_m$ на думи от V за $m < k$. Т.е. моделът се определя от крайната фамилия от контекстни локални разпределения $\{\Pr[x \mid x_1 x_2 \ldots x_m]\}_{x_1 x_2 \ldots x_m \in V^*, m<k}$.

- Марковските езикови модели са широко използвани в по-ранните системи поради тяхната простота и изчислителна ефективност.

- По-модерните методи базирани на дълбоки невронни мрежи могат да представят не-марковски езикови модели — ще разгледаме по-нататък в курса.

- Ефективни изчислителни методи за представяне на марковски езиков модел се базират на претеглени крайни преобразуватели (WFST — Weighted Finite-State Transducers), които се разглеждат в курса по ПКА.

Обучение на Марковски езиков модел

- Принцип за максимизиране на правдоподобието:
- $$\hat{\Pr}^k_{MLE}[x \mid x_1 x_2 \ldots x_m] = \begin{cases} \dfrac{\#(x_1 x_2 \ldots x_m x)}{\#(x_1 x_2 \ldots x_m)} & \text{if } m = k-1 \\ \dfrac{\#(\hat{} x_1 x_2 \ldots x_m x)}{\#(\hat{} x_1 x_2 \ldots x_m)} & \text{if } m < k-1 \end{cases},$$

  където $\#(x_1 x_2 \ldots x_m x)$ е броят срещания на последователността $x_1 x_2 \ldots x_m x$ в корпуса, а $\#(\hat{} x_1 x_2 \ldots x_m x)$ е броят срещания на последователността $x_1 x_2 \ldots x_m x$ в началото на последователност от корпуса.

- Свеждаме обучението до броене на срещания на k-орки в корпус от последователности.

- **<u>Недостатък</u>**: Ако дадена k-орка не се е срещнала в корпуса, то съответната вероятност $= 0$.

Изглаждане на Марковски езиков модел

- Броят на различните k-орки расте експоненциално с k. При речник от 30000 елемента има 900 000 000 биграми и 27 000 000 000 000 триграми.

- Наивно и некоректно е да предполагаме, че всички k-орки са се срещнали в корпуса.

- Най-просто изглаждане — add α изглаждане:

$$\hat{\Pr}_{\text{add}\alpha}[x \mid x_1 x_2 \ldots x_m] = \frac{\#(x_1 x_2 \ldots x_m x) + \alpha}{\#(x_1 x_2 \ldots x_m) + \alpha |V'|}.$$

- При $\alpha = 1$ получаваме изглаждане на Лаплас

- **Проблем**: изглаждането add α дава една и съща вероятност на всички k -орки, които не са се срещнали в корпуса.

Изглаждане на езиков модел

- **Идея**: Вероятностите на k-орките, които не са се срещнали, ще се определят от вероятностите на съответните $k-1$-орки.

- Изглаждане с интерполация на Йелинек-Мерсер (Jelinek-Mercer interpolated smoothing):

$$\hat{\Pr}^{k}_{\text{int}}[x \mid x_1 x_2 \ldots x_m] = \lambda \hat{\Pr}^{k}_{\text{MLE}}[x \mid x_1 x_2 \ldots x_m] + (1-\lambda)\hat{\Pr}^{k-1}_{\text{int}}[x \mid x_2 x_3 \ldots x_m]$$

- Параметърът $\lambda \in (0,1)$ се настройва, така че да се получи "най-добър" езиков модел. Как да оценим колко е добър даден езиков модел ще видим на следващата лекция.

- Съществуват много други техники за изглаждане. Една от най-успешните техника за изглаждане на k-грамен езиков модел е модифицираното изглаждане на Кнесер-Ней (modified Knesser-Ney smoothing).

Пример за двуграмен Марковски езиков модел и изглаждане с интерполация

Корпус

Иван кара кола $
Мария кара $
Иван гони Мария $
Мария купи кола $
Мария кара колело $

Монограми	Брой	Биграми	Брой
Иван	2	Иван кара	1
Мария	4	Мария кара	2
кара	3	Иван гони	1
купи	1	Мария купи	1
гони	1	кара кола	1
кола	2	кара колело	1
колело	1	гони Мария	1
		купи кола	1
$	5	^ Иван	2
Общо	**19**	^ Мария	3
		кара $	1
		кола $	2
		Мария $	1
		колело $	1
		Общо	**19**

При $\lambda = 0.75$:

$$\Pr[\text{ Мария кара кола \$}] = \Pr[\text{Мария} \mid \varepsilon] \Pr[\text{кара} \mid \text{Мария}] \Pr[\text{кола} \mid \text{кара}] \Pr[\$ \mid \text{кола}] =$$

$$= \left(\lambda\frac{3}{5} + (1-\lambda)\frac{4}{19}\right)\left(\lambda\frac{2}{4} + (1-\lambda)\frac{3}{19}\right)\left(\lambda\frac{1}{3} + (1-\lambda)\frac{2}{19}\right)\left(\lambda\frac{2}{2} + (1-\lambda)\frac{5}{19}\right) =$$

$$= 0.04696026422$$

План на лекцията

1. Формалности за курса (5 мин)
2. Ранкирано търсене на информация (10 мин)
3. Документно представяне чрез вектори от тегла $\mathrm{tf}{\cdot}\mathrm{idf}$ (15 мин)
4. Ранкиране при документно представяне чрез вектори от тегла (15 мин)
5. Езикови модели (15 мин)
6. k-грамни езикови модели и изглаждане на езиков модел (15 мин)
7. **Ранкиране чрез документни езикови модели (15 мин)**

Ранкиране с езиков модел

- **ИДЕЯ**:
  - Всеки документ дефинира езиков модел.
  - Използвайки езиковия модел на даден документ намираме вероятността заявката да бъде генерирана от този езиков модел.
- Ранкираме документите по вероятността да генерират дадената заявка.

Формализация на ранкирането с езиков модел

. Търсим $\hat{d} = \arg\max_{d} \Pr[d \mid q]$

. $\Pr[d \mid q] = \dfrac{\Pr[q \mid d] \Pr[d]}{\Pr[q]}$

- Вероятността $\Pr[q]$ е фиксирана и не зависи от d, затова я игнорираме.

- Априорната вероятност $\Pr[d]$ или приемаме за константа — т.е. всички документи са равновероятни и съответно я игнорираме, или я приближаваме, като използваме вероятности, базирани на критерии като авторитетност на документа, дължина, жанр, кога е създаден, брой ползватели, които са го достъпвали и др.

- За да намерим $\Pr[q \mid d]$, ще използваме езиков модел M_d извлечен от документа d.

- Индивидуалните документи в дадена колекция обикновено са сравнително къси (например около 1000 думи). Поради това обикновено се използва Марковски модел от ред 1 (монограмен модел).

- Монограмният езиков модел е еквивалентен на мултиномния наивен Бейсов модел (от предишната лекция), като всеки документ се третира като отделен клас. При този модел имаме:

$$\Pr[q \mid M_d] = \prod_{t \in V^{\$}} \Pr[t \mid \varepsilon; M_d]^{\mathrm{tf}_{t,q}},$$

Оценяване на документен монограмен езиков модел

- За да оценим параметрите на модела, използваме принципа за максимизиране на правдоподобието: $\hat{\Pr}_{MLE}[t \mid \varepsilon; M_d] = \frac{\mathrm{tf}_{t,d}}{L_d}$

- За да изгладим разпределението, ще използваме линейна интерполация между два езикови модела: $\Pr[t \mid \varepsilon; M_d] = \lambda \Pr_{MLE}[t \mid \varepsilon; M_d] + (1 - \lambda)\Pr_{MLE}[t \mid \varepsilon; M_C]$, където M_C е монограмният езиков модел извлечен от цялата колекция.

- Параметърът $\lambda \in (0,1)$ следва да бъде внимателно настроен. Често λ се настройва да зависи от дължината на q. При по-къси заявки се избира по-висока стойност, за да се засили значението всички термове от заявката да се срещат в документа.

- Ролята на изглаждането не е само за избягване на нулеви вероятности, но води и до подобряване качеството на модела.

Обобщение

1. Извличаме всички документи от колекцията, в които се среща някой от термовете на заявката

2. Изчисляваме за всеки документ d:
$$\Pr[d \mid q] \propto \Pr[d] \Pr[q \mid d] = \Pr[d] \prod_{t \in q} (\lambda \Pr[t \mid \varepsilon; M_d] + (1 - \lambda) \Pr[t \mid \varepsilon; M_C])$$

3. Априорната вероятност $\Pr[d]$ приближаваме да отчита авторитетността на документа.

4. Извеждаме първите k на брой документа подредени по вероятността да генерират заявката.

Забележка: В някои системи вместо да се ранкират документите по $\Pr[q \mid d]$, се получават по-добри резултати като се ранкират обратно пропорционално на релативната ентропия — ще разгледаме понятието ентропия на следващата лекция:

$$D(M_q \mid\mid M_d) = \sum_{t \in V} \Pr[t \mid \varepsilon; M_q] \log_2 \frac{\Pr[t \mid \varepsilon; M_q]}{\Pr[t \mid \varepsilon; M_d]}$$