ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

2. Mô hình ngôn ngữ

Lê Thanh Hương
Bộ môn Hệ thống Thông tin
Viện CNTT &TT – Trường ĐHBKHN
Email: huonglt@soict.hust.edu.vn

Mô hình ngôn ngữ

- Là phân bố xác suất trên các tập văn bản
- Cho biết xác suất của 1 câu (hoặc 1 cụm từ) thuộc 1 ngôn ngữ là bao nhiêu
- Mô hình ngôn ngữ tốt sẽ đánh giá đúng các câu đúng ngữ pháp, trôi chảy hơn các từ có thứ tự ngẫu nhiên.
- vd: P(“hôm nay trời đẹp”) > P(“trời đẹp nay hôm”)

Mô hình ngôn ngữ N-gram

- Mục tiêu: tính xác suất của 1 câu hoặc một cụm từ:

  $P(W) = P(w_1,w_2,w_3,w_4,w_5,\ldots,w_m)$

- Theo công thức Bayes:

  $P(AB) = P(B|A)*P(A)$

- Ta có:

  $P(w_1,w_2,w_3,w_4,w_5,\ldots,w_m) =$
  $P(w1)*P(w2|w1)*P(w3|w1w2)*\ldots*P(wm|w1w2w3 \ldots wm-1)$

  P("hôm nay trời đẹp") =

  P(hôm) * P(nay|hôm) * P(trời|hôm nay) * P(đẹp |hôm nay trời)

Mô hình ngôn ngữ N-gram

Cách tính xác suất: $P(\text{đẹp} \mid hôm\ nay\ trời) = \frac{P(hôm\ nay\ trời\ đẹp)}{P(hôm\ nay\ trời)}$

- không thể lưu hết các xác suất trên, đặc biệt với m là độ dài văn bản ngôn ngữ tự nhiên

➢ sử dụng chuỗi Markov bậc n với giả thiết 1 từ chỉ phụ thuộc n-1 từ đứng trước nó (mô hình n-gram)

$$P(w_m|w_1 w_2 w_3 .. w_{m-1}) = P(w_m|w_1, w_2, w_3, .., w_{m-1})$$

$$= P(w_m|w_{m-n} w_{m-n+1} w_{m-n+2} \cdots w_{m-1})$$

Các mô hình ngram

- Mô hình unigram: $P(w_1 w_2 \dots wn) \sim \prod_i P(w_i)$
- Mô hình bigram: $P(w_1 w_2 \dots wn) \sim \prod_i P(w_i | wi\_1)$
- Mô hình trigram: $P(w_1 w_2 \dots wn) \sim \prod_i P(w_i | wi\_1 w_i\_2)$

Tính xác suất bigram

- Đánh giá Maximum Likelihood

$$P(w_i \mid w_{i-1}) = \frac{count(w_{i-1}, w_i)}{count(w_{i-1})}$$

$$P(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$$

Tính xác suất bigram – ví dụ 1

$$P(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$$

<s> I am Sam </s>

<s> Sam I am </s>

<s> I do not like green eggs and ham </s>

$$P(\texttt{I} \mid \texttt{<s>}) = \frac{2}{3} = .67 \qquad P(\texttt{Sam} \mid \texttt{<s>}) = \frac{1}{3} = .33 \qquad P(\texttt{am} \mid \texttt{I}) = \frac{2}{3} = .67$$

$$P(\texttt{</s>} \mid \texttt{Sam}) = \frac{1}{2} = 0.5 \qquad P(\texttt{Sam} \mid \texttt{am}) = \frac{1}{2} = .5 \qquad P(\texttt{do} \mid \texttt{I}) = \frac{1}{3} = .33$$

Tính xác suất bigram – ví dụ 2
Berkeley Restaurant Project sentences

- can you tell me about any good cantonese restaurants close by
- mid priced thai food is what i'm looking for
- tell me about chez panisse
- can you give me a listing of the kinds of food that are available
- i'm looking for a good place to eat breakfast
- when is caffe venezia open during the day

Đếm các bigram

- Trên tổng số 9222 câu

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

Tính xác suất bigram

- Chuẩn hóa theo unigrams:

i	want	to	eat	chinese	food	lunch	spend
2533	927	2417	746	158	1093	341	278

- Kết quả:

	i	want	to	eat	chinese	food	lunch	spend
i	0.002	0.33	0	0.0036	0	0	0	0.00079
want	0.0022	0	0.66	0.0011	0.0065	0.0065	0.0054	0.0011
to	0.00083	0	0.0017	0.28	0.00083	0	0.0025	0.087
eat	0	0	0.0027	0	0.021	0.0027	0.056	0
chinese	0.0063	0	0	0	0	0.52	0.0063	0
food	0.014	0	0.014	0	0.00092	0.0037	0	0
lunch	0.0059	0	0	0	0	0.0029	0	0
spend	0.0036	0	0.0036	0	0	0	0	0

Tính xác suất câu dựa trên các bigram

P(<s> I want english food </s>) =

P(I|<s>)

× P(want|I)

× P(english|want)

× P(food|english)

× P(</s>|food)

= .000031

Các xác suất đã tính được

- P(english|want) = .0011
- P(chinese|want) = .0065
- P(to|want) = .66
- P(eat | to) = .28
- P(food | to) = 0
- P(want | spend) = 0
- P (i | <s>) = .25

Triển khai thực tế

- Sử dụng log thay cho phép nhân
  - Tránh được kết quả về 0
  - Nhanh hơn nhân

$$\log(p_1 \cdot p_2 \cdot p_3 \cdot p_4) = \log p_1 + \log p_2 + \log p_3 + \log p_4$$

Các mô hình ngôn ngữ có sẵn

- Google Book N-grams
  - http://ngrams.googlelabs.com/
- KenLM
  - https://kheafield.com/code/kenlm/

Google N-Gram Release

- serve as the incoming 92
- serve as the incubator 99
- serve as the independent 794
- serve as the index 223
- serve as the indication 72
- serve as the indicator 120
- serve as the indicators 45
- serve as the indispensable 111
- serve as the indispensible 40
- serve as the individual 234

http://googleresearch.blogspot.com/2006/08/all-our-n-gram-are-belong-to-you.html

Đánh giá các mô hình ngôn ngữ

- Gán xác suất cao cho các câu thực hoặc các câu có tần suất xuất hiện lớn
  - Hơn các câu sai ngữ pháp hoặc các câu ít xuất hiện?
- Huấn luyện mô hình trên một tập huấn luyện (training set)
- Đánh giá trên một tập dữ liệu mới (test set)
- Sử dụng ma trận độ đo để đánh giá mức độ tốt của mô hình trên tập test

Đánh giá mô hình N-gram

- So sánh 2 mô hình A và B
  - Sử dụng mỗi mô hình cho một nhiệm vụ cụ thể:
    - sửa lỗi chính tả, nhận dạng tiếng nói, dịch máy, …
  - Thử nghiệm (chạy) nhiệm vụ đó, tính độ chính xác khi sử dụng mô hình A và B
    - Bao nhiêu từ sai được sửa đúng
    - Bao nhiêu từ được dịch đúng
  - So sánh độ chính xác khi sử dụng A và B

Đánh giá mô hình N-gram – Đánh giá trong

- Đánh giá trong sử dụng độ đo **perplexity (độ phức tạp)**
  - Đánh giá xấp xỉ không tốt
    - Chỉ khi dữ liệu test giống dữ liệu train (về bộ từ vựng)
    - Tốt cho thí nghiệm nhưng không tốt cho thực tế

Ý tưởng của Perplexity

- Shannon Game:
  - Ta có thể tiên đoán từ tiếp theo không?

    I always order pizza with cheese and ____

    The 33rd President of the US was ____

    I saw a ____

  - Có thể dùng unigram không?
- Mô hình tố sẽ gán xác suất cao cho từ thường xuyên xuất hiện ở vị trí dự đoán

mushrooms 0.1
pepperoni 0.1
anchovies 0.01
….
fried rice 0.0001
….
and 1e-100

Độ phức tạp (Perplexity)

Mô hình tốt nhất là mô hình dự đoán từ chưa nhìn thấy tốt nhất

- Cho xác suất câu cao nhất P(sentence)
- Perplexity là nghịch đảo xác suất trên tập test, chuẩn hóa theo số từ

$$PP(W) = P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}}$$

$$= \sqrt[N]{\frac{1}{P(w_1 w_2 \ldots w_N)}}$$

Luật chuỗi:

$$\mathrm{PP}(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i | w_1 \ldots w_{i-1})}}$$

Với bigrams:

$$\mathrm{PP}(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i | w_{i-1})}}$$

Độ phức tạp (Perplexity)

- Độ phức tạp tương đương số trường hợp rẽ nhánh
- Giả thiết 1 câu gồm các chữ số ngẫu nhiên. Khi đó độ phức tạp của câu dựa trên 1 mô hình sẽ gán P=1/10

$$
\begin{aligned}
\mathrm{PP}(W) &= P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}} \\
&= \left(\frac{1}{10}^{N}\right)^{-\frac{1}{N}} \\
&= \frac{1}{10}^{-1} \\
&= 10
\end{aligned}
$$

Hiện tượng quá khớp dữ liệu (overfitting)

- N-grams chỉ tiên đoán từ tốt nếu tập test giống tập train.
  - Ta cần tạo ra mô hình có tính tổng quát, nghĩa là có thể xử lý các trường hợp xác suất = 0 (những TH không có trong tập train nhưng có trong tập test)

TH xác suất = 0

- Tập train:
  … denied the allegations
  … denied the reports
  … denied the claims
  … denied the request

- Tập test
  … denied the offer
  … denied the loan

P(“offer” | denied the) = 0

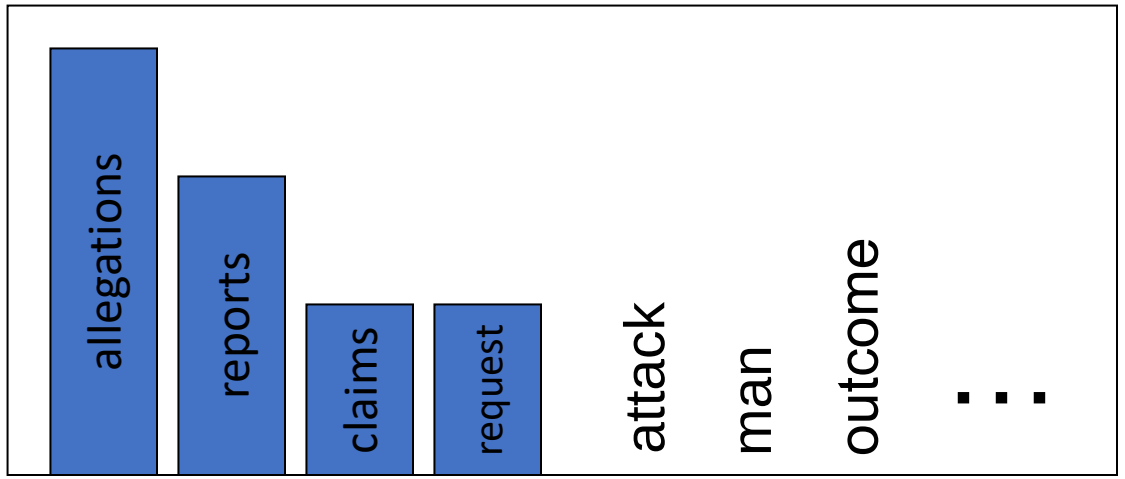
→ xác suất của 1 câu hoặc một cụm từ về 0

➢ Sử dụng các phương pháp làm mịn

Ý tưởng của phương pháp làm mịn

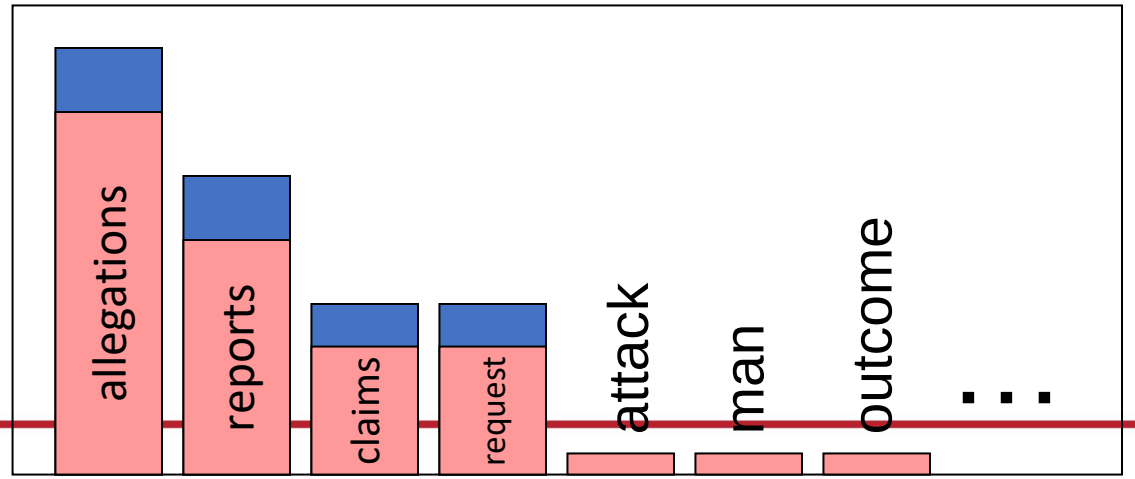
- Xác suất trên tập train:

P(w | denied the)
3 allegations
2 reports
1 claims
1 request
7 total



- Giảm xác suất các n-gram có xác suất lớn hơn 0 để bù cho các n-gram có xác suất bằng 0.

P(w | denied the)
2.5 allegations
1.5 reports
0.5 claims
0.5 request
2 other
7 total



Đánh giá kiểu add-one

- Gọi là phép làm mịn Laplace
- Giả thiết mỗi từ xuất hiện nhiều hơn 1 lần so với thực tế
- Cộng 1 vào tất cả các giá trị đếm

- Đánh giá MLE :

$$P_{MLE}(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$$

- Đánh giá add-1:

$$P_{Add\text{-}1}(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i)+1}{c(w_{i-1})+V}$$