Mô hình hóa ngôn ngữ

Giới thiệu về N-gram

Những mô hình ngôn ngữ xác suất

- Mục tiêu của buổi học: gán cho mỗi câu một xác suất

Why?

  - Dịch máy:
    - P(**high** winds tonite) > P(**large** winds tonite)
  - Sửa lỗi chính tả:
    - The office is about fifteen **minuets** from my house
      - P(about fifteen **minutes** from) > P(about fifteen **minuets** from)
  - Nhận dạng tiếng nói:
    - P(I saw a van) >> P(eyes awe of an)
  - Tóm tắt văn bản, hỏi đáp, …!!

Mô hình ngôn ngữ xác suất

- Mục tiêu: tính xác suất của một câu hay một chuỗi các từ:

  $P(W) = P(w_1,w_2,w_3,w_4,w_5...w_n)$

- Tác vụ có liên quan: xác suất của từ xuất hiện kế tiếp:

  $P(w_5|w_1,w_2,w_3,w_4)$

- Mô hình tính một trong hai thứ trên:

  $P(W)$ hay $P(w_n|w_1,w_2...w_{n-1})$ được gọi là mô hình ngôn ngữ (**language model)**.

- **Language model** hay **LM** là từ chuẩn

Tính P(W) bằng cách nào?

- Làm thế nào để tính xác suất kết hợp (joint probability):
  - P(its, water, is, so, transparent, that)
- Ý tưởng đầu tiên: hãy sử dụng Luật dây chuyền (Chain Rule) của xác suất

Nhắc lại: Chain Rule

- Nhắc lại định nghĩa xác suất
- Nhiều biến hơn:

  $P(A,B,C,D) = P(A)P(B|A)P(C|A,B)P(D|A,B,C)$

- Chain Rule trong trường hợp tổng quát

$$P(x_1,x_2,x_3,\ldots,x_n) = P(x_1)P(x_2|x_1)P(x_3|x_1,x_2)\ldots P(x_n|x_1,\ldots,x_{n-1})$$

Ứng dụng Chain Rule để tính xác suất kết hợp của các từ trong câu

$$P(w_1 w_2 \ldots w_n) = \prod_i P(w_i \mid w_1 w_2 \ldots w_{i-1})$$

P("its water is so transparent") =

P(its) × P(water|its) × P(is|its water)

× P(so|its water is) × P(transparent|its water is so)

Làm sao để ước lượng những xác suất này?

- Chúng ta có thể chỉ cần đếm và chia?

$$P(\text{the} \mid \text{its water is so transparent that}) = \frac{Count(\text{its water is so transparent that the})}{Count(\text{its water is so transparent that})}$$

- Không thể! Quá nhiều câu khả dĩ!
- Chúng ta không bao giờ có đủ dữ liệu để ước lượng.

Giả định Markov

Andrei Markov

- Giả định đơn giản hóa:

$$P(\text{the} \mid \text{its water is so transparent that}) \approx P(\text{the} \mid \text{that})$$

- Hoặc có thể

$$P(\text{the} \mid \text{its water is so transparent that}) \approx P(\text{the} \mid \text{transparent that})$$

Giả định Markov

$$P(w_1 w_2 \ldots w_n) \approx \prod_i P(w_i \mid w_{i-k} \ldots w_{i-1})$$

- Nói cách khác, chúng ta xấp xỉ mỗi thành phần trong tích:

$$P(w_i \mid w_1 w_2 \ldots w_{i-1}) \approx P(w_i \mid w_{i-k} \ldots w_{i-1})$$

Trường hợp đơn giản nhất: Mô hình Unigram

$$P(w_1 w_2 \ldots w_n) \approx \prod_i P(w_i)$$

Một số câu được tạo sinh tự động từ mô hình unigram

fifth, an, of, futures, the, an, incorporated, a, a, the, inflation, most, dollars, quarter, in, is, mass

thrift, did, eighty, said, hard, 'm, july, bullish

that, or, limited, the

Mô hình Bigram

- Điều kiện trên từ đứng trước:

$$P(w_i \mid w_1 w_2 \ldots w_{i-1}) \approx P(w_i \mid w_{i-1})$$

texaco, rose, one, in, this, issue, is, pursuing, growth, in, a, boiler, house, said, mr., gurria, mexico, 's, motion, control, proposal, without, permission, from, five, hundred, fifty, five, yen

outside, new, car, parking, lot, of, the, agreement, reached

this, would, be, a, record, november

Mô hình N-gram khác

- Chúng ta có thể mở rộng sang trigrams, 4-grams, 5-grams
- Nói chung đây là mô hình ngôn ngữ không đầy đủ
  - Bởi ngôn ngữ có những quan hệ phụ thuộc có khoảng cách xa **long-distance dependencies**:

  "The computer which I had just put into the machine room on the fifth floor crashed."

- Nhưng thường các mô hình N-gram được dùng rất phổ biến

Mô hình hóa ngôn ngữ

Ước lượng xác suất N-gram

Ước lượng xác suất bigram

- Ước lượng Maximum Likelihood

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

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

Ví dụ

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

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

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

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

Ví dụ:
Một số câu trong Dự án Berkeley Restaurant

- 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

- 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

Đếm các bigram

- Chuẩn hóa bằng số lượng unigram:

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

Ước lượng bigram cho xác suất của câu

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

P(I|<s>)

× P(want|I)

× P(english|want)

× P(food|english)

× P(</s>|food)

= .000031

Những loại tri thức gì?

- 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

Những vấn đề thực tế

- Chúng ta làm tất cả trong không gian log
  - Tránh tràn số underflow
  - Phép cộng nhanh hơn phép nhân

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

Bộ công cụ mô hình hóa ngôn ngữ

- LM Toolkit - SRILM
  - http://www.speech.sri.com/projects/srilm/

Google N-Gram Release, 8/2006

AUG
3

All Our N-gram are Belong to You

Posted by Alex Franz and Thorsten Brants, Google Machine Translation Team

Here at Google Research we have been using word n-gram models for a variety of R&D projects,

...

That's why we decided to share this enormous dataset with everyone. We processed 1,024,908,267,229 words of running text and are publishing the counts for all 1,176,470,663 five-word sequences that appear at least 40 times. There are 13,588,391 unique words, after discarding words that appear less than 200 times.

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

Google Book N-grams

- http://ngrams.googlelabs.com/

Mô hình hóa ngôn ngữ

Đánh giá và perplexity

Đánh giá: Mô hình của chúng ta tốt cỡ nào?

- Mô hình ngôn ngữ của ta có đánh giá cao câu đúng hơn câu sai?
  - Gắn xác suất cao hơn cho câu “thực” hay “thường thấy”
    - Hơn câu “không đúng ngữ pháp” hay “ít thấy”?
- Chúng ta huấn luyện các tham số của mô hình dựa trên tập dữ liệu huấn luyện (**training set**).
- Chúng ta kiểm thử hoạt động của mô hình trên dữ liệu chưa thấy.
  - Tập kiểm thử (**test set**) is là một tập dữ liệu chưa thấy, khác với tập huấn luyện, không được sử dụng cho các mục đích khác.
  - Độ đo đánh giá (**evaluation metric**) cho chúng ta biết mô hình tốt cỡ nào trên tập kiểm thử.

Đánh giá ngoài (Extrinsic evaluation) mô hình N-gram

- Là cách đánh giá tốt nhất để so sánh 2 mô hình A và B
  - Đặt mô hình vào trong một tác vụ cụ thể
    - Sửa lỗi chính tả, nhận dạng tiếng nói, dịch máy
  - Chạy tác vụ, đo độ chính xác tương ứng với A và B
    - Bao nhiêu từ sai chính tả được sửa lại cho đúng
    - Bao nhiêu từ được dịch đúng
  - So sánh độ chính xác (accuracy) của A và B

Khó khăn của đánh giá ngoài mô hình N-gram

- Đánh giá ngoài (extrinsic, in-vivo)
  - Mất thời gian; có thể mất vài ngày đến vài tuần
- Do đó
  - Thỉnh thoảng sử dụng đánh giá nội tại (**intrinsic** evaluation): **perplexity**
  - Là một xấp xỉ tồi
    - Nếu như dữ liệu kiểm thử giống như dữ liệu huấn luyện
    - Do đó thường hữu ích trong các thử nghiệm
  - Nhưng cũng nên biết vì nó hữu ích.

Mô hình hóa ngôn ngữ

Những số không

Những số không

- Tập huấn luyện:
  … denied the allegations
  … denied the reports
  … denied the claims
  … denied the request

  P(“offer” | denied the) = 0

- Tập kiểm thử
  … denied the offer
  … denied the loan

Xác suất bigram bằng không

- Những bigram có xác suất bằng không
  - Nghĩa là chúng ta gán xác suất 0 cho tập kiểm thử!
- Và do đó chúng ta không thể tính perplexity (không thể chia cho 0)!

Mô hình hóa ngôn ngữ

Smoothing: Add-one (Laplace) smoothing

Ước lượng Thêm 1 (Add-one estimation)

- Còn gọi là Laplace smoothing
- Giả vờ như chúng ta thấy mỗi từ nhiều hơn số thực tế 1 lần
- Chỉ cần cộng 1 vào tất cả số lượng đếm được!
- Ước lượng MLE: $$P_{MLE}(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$$
- Ước lượng Add-1: $$P_{Add-1}(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i)+1}{c(w_{i-1})+V}$$

Ngữ liệu Berkeley Restaurant: Đếm bigram đã được làm mịn Laplace

	i	want	to	eat	chinese	food	lunch	spend
i	6	828	1	10	1	1	1	3
want	3	1	609	2	7	7	6	2
to	3	1	5	687	3	1	7	212
eat	1	1	3	1	17	3	43	1
chinese	2	1	1	1	1	83	2	1
food	16	1	16	1	2	5	1	1
lunch	3	1	1	1	1	2	1	1
spend	2	1	2	1	1	1	1	1

Xác suất bigram đã được làm mịn Laplace

$$P^*(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)+1}{C(w_{n-1})+V}$$

	i	want	to	eat	chinese	food	lunch	spend
i	0.0015	0.21	0.00025	0.0025	0.00025	0.00025	0.00025	0.00075
want	0.0013	0.00042	0.26	0.00084	0.0029	0.0029	0.0025	0.00084
to	0.00078	0.00026	0.0013	0.18	0.00078	0.00026	0.0018	0.055
eat	0.00046	0.00046	0.0014	0.00046	0.0078	0.0014	0.02	0.00046
chinese	0.0012	0.00062	0.00062	0.00062	0.00062	0.052	0.0012	0.00062
food	0.0063	0.00039	0.0063	0.00039	0.00079	0.002	0.00039	0.00039
lunch	0.0017	0.00056	0.00056	0.00056	0.00056	0.0011	0.00056	0.00056
spend	0.0012	0.00058	0.0012	0.00058	0.00058	0.00058	0.00058	0.00058

Đếm bigram được tính lại theo xác suất làm mịn

$$c^*(w_{n-1}w_n) = \frac{[C(w_{n-1}w_n)+1] \times C(w_{n-1})}{C(w_{n-1})+V}$$

	i	want	to	eat	chinese	food	lunch	spend
i	3.8	527	0.64	6.4	0.64	0.64	0.64	1.9
want	1.2	0.39	238	0.78	2.7	2.7	2.3	0.78
to	1.9	0.63	3.1	430	1.9	0.63	4.4	133
eat	0.34	0.34	1	0.34	5.8	1	15	0.34
chinese	0.2	0.098	0.098	0.098	0.098	8.2	0.2	0.098
food	6.9	0.43	6.9	0.43	0.86	2.2	0.43	0.43
lunch	0.57	0.19	0.19	0.19	0.19	0.38	0.19	0.19
spend	0.32	0.16	0.32	0.16	0.16	0.16	0.16	0.16

So sánh với đếm bigram nguyên gốc

	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

	i	want	to	eat	chinese	food	lunch	spend
i	3.8	527	0.64	6.4	0.64	0.64	0.64	1.9
want	1.2	0.39	238	0.78	2.7	2.7	2.3	0.78
to	1.9	0.63	3.1	430	1.9	0.63	4.4	133
eat	0.34	0.34	1	0.34	5.8	1	15	0.34
chinese	0.2	0.098	0.098	0.098	0.098	8.2	0.2	0.098
food	6.9	0.43	6.9	0.43	0.86	2.2	0.43	0.43
lunch	0.57	0.19	0.19	0.19	0.19	0.38	0.19	0.19
spend	0.32	0.16	0.32	0.16	0.16	0.16	0.16	0.16

Ước lượng Thêm 1 (Add-1) là một công cụ thô sơ

- Do đó add-1 thực tế không được dùng cho mô hình N-gram:
  - Chúng ta sẽ thấy những phương pháp tốt hơn
- Nhưng add-1 được sử dụng để làm mịn những mô hình XLNNTN khác
  - Cho phân loại văn bản
  - Trong những lĩnh vực dữ liệu (domains) mà số lượng 0 không quá nhiều.

Mô hình hóa ngôn ngữ khác

Interpolation, Backoff, và Web-Scale LMs

Backoff và Interpolation

- Đôi khi nên dùng ít ngữ cảnh hơn (**less** context)
  - Điều kiện trên ít ngữ cảnh hơn cho những ngữ cảnh mà bạn không được học nhiều về chúng
- **Backoff:**
  - Sử dụng trigram nếu có evidence tốt,
  - Nếu không thì dùng bigram, không thì dùng unigram
- **Interpolation:**
  - Trộn lẫn unigram, bigram, trigram
- Interpolation cho thấy tốt hơn trong thực tế

Tóm tắt N-gram Smoothing

- Add-1 smoothing:
  - Dùng ổn cho phân loại văn bản, không tốt cho mô hình hóa ngôn ngữ
- Mô hình sử dụng phổ biến nhất:
  - Extended Interpolated Kneser-Ney (đọc sách)
- Đối với những dữ liệu N-gram lớn như Web:
  - Stupid backoff