Language Modeling

Michael Collins, Columbia University

Overview

- The language modeling problem
- Trigram models
- Evaluating language models: perplexity
- Estimation techniques:
  - Linear interpolation
  - Discounting methods

The Language Modeling Problem

- We have some (finite) vocabulary,
  say $\mathcal{V} = \{$the, a, man, telescope, Beckham, two, $\ldots\}$

- We have an (infinite) set of strings, $\mathcal{V}^{\dagger}$

  the STOP
  a STOP
  the fan STOP
  the fan saw Beckham STOP
  the fan saw saw STOP
  the fan saw Beckham play for Real Madrid STOP

The Language Modeling Problem (Continued)

- We have a *training sample* of example sentences in English

The Language Modeling Problem (Continued)

- We have a *training sample* of example sentences in English
- We need to "learn" a probability distribution p i.e., p is a function that satisfies

$$\sum_{x \in \mathcal{V}^\dagger} p(x) = 1, \quad p(x) \geq 0 \text{ for all } x \in \mathcal{V}^\dagger$$

The Language Modeling Problem (Continued)

- We have a *training sample* of example sentences in English
- We need to "learn" a probability distribution p i.e., p is a function that satisfies

$$\sum_{x \in \mathcal{V}^\dagger} p(x) = 1, \quad p(x) \geq 0 \text{ for all } x \in \mathcal{V}^\dagger$$

$p(\text{the STOP}) = 10^{-12}$
$p(\text{the fan STOP}) = 10^{-8}$
$p(\text{the fan saw Beckham STOP}) = 2 \times 10^{-8}$
$p(\text{the fan saw saw STOP}) = 10^{-15}$
$\ldots$
$p(\text{the fan saw Beckham play for Real Madrid STOP}) = 2 \times 10^{-9}$
$\ldots$

Why on earth would we want to do this?!

- **Speech recognition** was the original motivation. (Related problems are optical character recognition, handwriting recognition.)

Why on earth would we want to do this?!

- **Speech recognition** was the original motivation. (Related problems are optical character recognition, handwriting recognition.)
- The estimation techniques developed for this problem will be **VERY** useful for other problems in NLP

A Naive Method

- We have N training sentences
- For any sentence $x_1 \ldots x_n$, $c(x_1 \ldots x_n)$ is the number of times the sentence is seen in our training data
- A naive estimate:

$$p(x_1 \ldots x_n) = \frac{c(x_1 \ldots x_n)}{N}$$

Overview

- The language modeling problem
- Trigram models
- Evaluating language models: perplexity
- Estimation techniques:
  - Linear interpolation
  - Discounting methods

Markov Processes

- Consider a sequence of random variables $X_1, X_2, \ldots X_n$. Each random variable can take any value in a finite set $\mathcal{V}$. For now we assume the length n is fixed (e.g., $n = 100$).

- Our goal: model

$$P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n)$$

First-Order Markov Processes
一阶马尔科夫

$$P(X_1 = x_1, X_2 = x_2, \ldots X_n = x_n)$$

马尔可夫模型本质上是一个加权的有限状态机，它描述了不同状态之间的转换关系以及转换概率（这里的权重就是状态转移概率）。

常见的是一阶马尔科夫：当前状态出现的概率，只取决于上一个状态

First-Order Markov Processes

$$
\begin{aligned}
& P(X_1 = x_1, X_2 = x_2, \ldots X_n = x_n) \\
= \;& P(X_1 = x_1) \prod_{i=2}^{n} P(X_i = x_i | X_1 = x_1, \ldots, X_{i-1} = x_{i-1})
\end{aligned}
$$

First-Order Markov Processes

$$\begin{aligned}
& P(X_1 = x_1, X_2 = x_2, \ldots X_n = x_n) \\
= \; & P(X_1 = x_1) \prod_{i=2}^{n} P(X_i = x_i | X_1 = x_1, \ldots, X_{i-1} = x_{i-1}) \\
= \; & P(X_1 = x_1) \prod_{i=2}^{n} P(X_i = x_i | X_{i-1} = x_{i-1})
\end{aligned}$$

First-Order Markov Processes

$$
\begin{aligned}
& P(X_1 = x_1, X_2 = x_2, \ldots X_n = x_n) \\
= \;& P(X_1 = x_1) \prod_{i=2}^{n} P(X_i = x_i | X_1 = x_1, \ldots, X_{i-1} = x_{i-1}) \\
= \;& P(X_1 = x_1) \prod_{i=2}^{n} P(X_i = x_i | X_{i-1} = x_{i-1})
\end{aligned}
$$

The first-order Markov assumption: For any $i \in \{2 \ldots n\}$, for any $x_1 \ldots x_i$,

$$P(X_i = x_i | X_1 = x_1 \ldots X_{i-1} = x_{i-1}) = P(X_i = x_i | X_{i-1} = x_{i-1})$$

Second-Order Markov Processes

$$P(X_1 = x_1, X_2 = x_2, \ldots X_n = x_n)$$

Second-Order Markov Processes

$$
\begin{aligned}
& P(X_1 = x_1, X_2 = x_2, \ldots X_n = x_n) \\
= \;& P(X_1 = x_1) \times P(X_2 = x_2 | X_1 = x_1) \\
& \times \prod_{i=3}^{n} P(X_i = x_i | X_{i-2} = x_{i-2}, X_{i-1} = x_{i-1})
\end{aligned}
$$

Second-Order Markov Processes

$$
\begin{aligned}
& P(X_1 = x_1, X_2 = x_2, \ldots X_n = x_n) \\
= & P(X_1 = x_1) \times P(X_2 = x_2 | X_1 = x_1) \\
& \times \prod_{i=3}^{n} P(X_i = x_i | X_{i-2} = x_{i-2}, X_{i-1} = x_{i-1}) \\
= & \prod_{i=1}^{n} P(X_i = x_i | X_{i-2} = x_{i-2}, X_{i-1} = x_{i-1})
\end{aligned}
$$

(For convenience we assume $x_0 = x_{-1} = *$, where * is a special "start" symbol.)

Modeling Variable Length Sequences

- We would like the length of the sequence, n, to also be a random variable
- A simple solution: always define $X_n = \text{STOP}$ where STOP is a special symbol

Modeling Variable Length Sequences

- We would like the length of the sequence, n, to also be a random variable
- A simple solution: always define $X_n = \text{STOP}$ where STOP is a special symbol
- Then use a Markov process as before:

$$
\begin{aligned}
& P(X_1 = x_1, X_2 = x_2, \ldots X_n = x_n) \\
= & \prod_{i=1}^{n} P(X_i = x_i | X_{i-2} = x_{i-2}, X_{i-1} = x_{i-1})
\end{aligned}
$$

(For convenience we assume $x_0 = x_{-1} = *$, where $*$ is a special "start" symbol.)

Trigram Language Models

- A trigram language model consists of:
  1. A finite set $\mathcal{V}$
  2. A parameter $q(w|u, v)$ for each trigram u, v, w such that $w \in \mathcal{V} \cup \{\text{STOP}\}$, and $u, v \in \mathcal{V} \cup \{*\}$.

Trigram Language Models

- A trigram language model consists of:
  1. A finite set $\mathcal{V}$
  2. A parameter $q(w|u,v)$ for each trigram u, v, w such that $w \in \mathcal{V} \cup \{\text{STOP}\}$, and $u, v \in \mathcal{V} \cup \{*\}$.

- For any sentence $x_1 \dots x_n$ where $x_i \in \mathcal{V}$ for $i = 1 \dots (n-1)$, and $x_n = \text{STOP}$, the probability of the sentence under the trigram language model is

$$p(x_1 \dots x_n) = \prod_{i=1}^{n} q(x_i | x_{i-2}, x_{i-1})$$

where we define $x_0 = x_{-1} = *$.

The dog barks STOP

An Example

For the sentence

the dog barks STOP

we would have

$$
\begin{aligned}
p(\text{the dog barks STOP}) &= q(\text{the}|*, *) \\
&\quad \times q(\text{dog}|*, \text{the}) \\
&\quad \times q(\text{barks}|\text{the}, \text{dog}) \\
&\quad \times q(\text{STOP}|\text{dog}, \text{barks})
\end{aligned}
$$

The Trigram Estimation Problem

Remaining estimation problem:

$$q(w_i \mid w_{i-2}, w_{i-1})$$

For example:

$$q(\text{laughs} \mid \text{the, dog})$$

怎么算?

The Trigram Estimation Problem

Remaining estimation problem:

$$q(w_i \mid w_{i-2}, w_{i-1})$$

For example:

$$q(\text{laughs} \mid \text{the, dog})$$

A natural estimate (the "maximum likelihood estimate"):

$$q(w_i \mid w_{i-2}, w_{i-1}) = \frac{\text{Count}(w_{i-2}, w_{i-1}, w_i)}{\text{Count}(w_{i-2}, w_{i-1})}$$

$$q(\text{laughs} \mid \text{the, dog}) = \frac{\text{Count(the, dog, laughs)}}{\text{Count(the, dog)}}$$

Sparse Data Problems

A natural estimate (the "maximum likelihood estimate"):

$$q(w_i \mid w_{i-2}, w_{i-1}) = \frac{\text{Count}(w_{i-2}, w_{i-1}, w_i)}{\text{Count}(w_{i-2}, w_{i-1})}$$

$$q(\text{laughs} \mid \text{the, dog}) = \frac{\text{Count}(\text{the, dog, laughs})}{\text{Count}(\text{the, dog})}$$

Say our vocabulary size is $N = |\mathcal{V}|$, then there are N^3 parameters in the model.

e.g., $N = 20,000 \quad \Rightarrow \quad 20,000^3 = 8 \times 10^{12}$ parameters

Overview

- The language modeling problem
- Trigram models
- Evaluating language models: perplexity
- Estimation techniques: PPL 困惑度
    - Linear interpolation
    - Discounting methods

Evaluating a Language Model: Perplexity

- We have some test data, m sentences

$$s_1, s_2, s_3, \ldots, s_m$$

Evaluating a Language Model: Perplexity

- We have some test data, m sentences

$$s_1, s_2, s_3, \ldots, s_m$$

- We could look at the probability under our model $\prod_{i=1}^{m} p(s_i)$. Or more conveniently, the *log probability*

$$\log \prod_{i=1}^{m} p(s_i) = \sum_{i=1}^{m} \log p(s_i)$$

Evaluating a Language Model: Perplexity

混淆度 (Perplexity) 用来衡量一个语言模型在未见过的的字符串S上的表现。

- We have some test data, m sentences

$$s_1, s_2, s_3, \ldots, s_m$$

- We could look at the probability under our model $\prod_{i=1}^{m} p(s_i)$. Or more conveniently, the *log probability*

$$\log \prod_{i=1}^{m} p(s_i) = \sum_{i=1}^{m} \log p(s_i)$$

- In fact the usual evaluation measure is *perplexity*

$$\text{Perplexity} = 2^{-l} \quad \text{where} \quad l = \frac{1}{M} \sum_{i=1}^{m} \log p(s_i)$$

and M is the total number of words in the test data.

Some Intuition about Perplexity

- Say we have a vocabulary $\mathcal{V}$, and $N = |\mathcal{V}| + 1$ and model that predicts

$$q(w|u,v) = \frac{1}{N}$$

for all $w \in \mathcal{V} \cup \{\text{STOP}\}$, for all $u, v \in \mathcal{V} \cup \{*\}$.
- Easy to calculate the perplexity in this case:

$$\text{Perplexity} = 2^{-l} \quad \text{where} \quad l = \log \frac{1}{N}$$

$\Rightarrow$

$$\text{Perplexity} = N$$

Perplexity is a measure of effective "branching factor"

平均分支系数

示例:训练好的bigram语言模型的困惑度为?

- $p(w_1|BOS) = 0, p(w_2|BOS) = 1, p(w_3|BOS) = 0$;
- $p(w_1|w_1) = \frac{1}{3}, p(w_2|w_1) = \frac{1}{3}, p(w_3|w_1) = \frac{1}{3}$;
- $p(w_1|w_2) = \frac{1}{3}, p(w_2|w_2) = \frac{1}{3}, p(w_3|w_2) = \frac{1}{3}$;
- $p(w_1|w_3) = \frac{1}{3}, p(w_2|w_3) = \frac{1}{3}, p(w_3|w_3) = \frac{1}{3}$;
- $p(EOS|w_1) = \frac{1}{3}, p(EOS|w_2) = \frac{1}{3}, p(EOS|w_3) = \frac{1}{3}$;

计算 $perplexity(w_2, w_1, w_3)$ 的值