Language Model (LM)

LING 570

Fei Xia

Week 4: 10/21/2009

LM

- Word prediction: to predict the next word in a sentence.
  - Ex: I'd like to make a collect ___
- Statistical models of word sequences are called language models (LMs).
- Task:
  - Build a statistical model from the training data.
  - Now given a sentence w_1 w_2 … w_n, we want to estimate its probability $P(w_1 \ldots w_n)$?
- Goal: the model should prefer good sentences to bad ones.

Some Terms

- Corpus: a collection of text or speech
- Words: may or may not include punctuation marks.
- Types: the number of distinct words in a corpus
- Tokens: the total number of words in a corpus

Applications of LMs

- Speech recognition
  - Ex: I bought two/too/to books.
- Handwriting recognition
- Machine translation
- Spelling correction
- …

Outline

- N-gram LM
- Evaluation

N-gram LM

N-gram LM

- Given a sentence $w_1\ w_2\ \dots\ w_n$, how to estimate its probability $P(w_1 \dots w_n)$?

- The Markov independence assumption:
  $P(w_n \mid w_1, \dots, w_{n-1})$ depends only on the previous k words.

- $P(w_1 \dots w_n)$
  $= P(w_1) * P(w_2|w_1) * \dots P(w_n \mid w_1, \dots, w_{n-1})$
  $\approx P(w_1) * P(w_2|w_1) * \dots P(w_n \mid w_{n-k+1}, \dots, w_{n-1})$

- 0^{th} order Markov model: unigram model
- 1^{st} order Markov model: bigram model
- 2^{nd} order Markov model: trigram model
- …

Unigram LM

- $P(w_1 \ldots w_n)$

$\approx P(w_1)*P(w_2)* \ldots P(w_n)$

- Estimating P(w):
  - MLE: P(w) = C(w)/N, N is the num of tokens

- How many states in the FSA?
- How many model parameters?

Bigram LM

- $P(w_1 \dots w_n)$

  $= P(\text{BOS } w_1 \dots w_n \text{ EOS})$

  $\approx P(\text{BOS}) * P(w_1|\text{BOS}) * \dots *P(w_n \mid w_{n-1})*P(\text{EOS}|w_n)$

- Estimating $P(w_n|w_{n-1})$:
  - MLE: $P(w_n) = C(w_{n-1},w_n)/C(w_{n-1})$

- How many states in the FSA?
- How many model parameters?

Trigram LM

- $P(w_1\ldots w_n) = P(\text{BOS } w_1\ldots w_n \text{ EOS})$

$\approx P(\text{BOS}) * P(w_1|\text{BOS}) * P(w_2|\text{BOS}, w_1) * \ldots$

$* P(w_n \mid w_{n-2}, w_{n-1}) * P(\text{EOS} \mid w_{n-1}\ w_n)$

- Estimating $P(w_n|w_{n-2}, w_{n-1})$:
  - MLE: $P(w_n) = C(w_{n-2}, w_{n-1}, w_n)/C(w_{n-2}, w_{n-1})$

- How many states in the FSA?
- How many model parameters?

Text generation

Unigram: To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have

Bigram: What means, sir. I confess she? then all sorts, he is trim, captain.

Trigram: Sweet prince, Falstaff shall die. Harry of Monmouth’s grave.

4-gram: Will you not tell me who I am? It cannot be but so.

N-gram LM packages

- SRI LM toolkit
- CMU LM toolkit
- …

So far

- N-grams:
  – Number of states in FSA: $|V|^{N-1}$
  – Number of model parameters: $|V|^N$

- Remaining issues:
  – Data sparse problem ➔ smoothing
    - Unknown words: OOV rate
  – Mismatch between training and test data
    ➔ model adaptation
  – Other LM models: structured LM, class-based LM

Evaluation

Evaluation (in general)

- Evaluation is required for almost all CompLing papers.

- There are many factors to consider:
  - Data
  - Metrics
  - Results of competing systems
  - …

- You need to think about evaluation from the very beginning.

Rules for evaluation

- Always evaluate your system
- **Use standard metrics**
- **Separate training/dev/test data**
- Use standard training/dev/test data

- Clearly specify experiment setting
- Include baseline and results from competing systems
- Perform error analysis
- Show the system is useful for real applications (optional)

Division of data

- Training data
  - True training data: to learn model parameters
  - held-out data: to tune other parameters
- Development data: used when developing a system.
- Test data: used only once, for the final evaluation
- Dividing the data:
  - Common ratio: 80%, 10%, 10%.
  - N-fold validation

Standard metrics for LM

- Direct evaluation
  - Perplexity

- Indirect evaluation:
  - ASR
  - MT
  - …

Perplexity

- Perplexity is based on computing the probabilities of each sentence in the test set.

- Intuitively, whichever model assigns a higher probability to the test set is a better model.

Definition of perplexity

Test data T = $s_0 \ldots s_m$
Let N be the total number of words in T

$$P(T) = \prod_{i=0}^{m} P(s_i)$$

$$PPL(T) = P(T)^{-\frac{1}{N}} = \frac{1}{\sqrt[N]{P(T)}}$$

Lower values mean that the model is better.

Perplexity

$$PPL(T) = P(T)^{-\frac{1}{N}} = \frac{1}{\sqrt[N]{P(T)}}$$

$$= 2^{-\frac{1}{N} log_2 P(T)}$$

$$= 2^{H(L,P)}$$

$$PPL(T) = 10^{-\frac{1}{N} log_{10} P(T)}$$

Calculating Perplexity

$$PPL(T) = 10^{-\frac{1}{N} lgP(T)}$$

Suppose T consists of m sentences: s_1, …, s_m

$$lgP(T) = lg \prod_{i=1}^{m} P(s_i) = \sum_{i=1}^{m} lgP(s_i)$$

N = word_num + sent_num – oov_num

Calculating P(s)

- Let $s = w_1 \dots w_n$

$P(w_1 \dots w_n) = P(BOS\ w_1 \dots w_n\ EOS)$
$= P(w_1|BOS)*P(w_2\ |BOS,w_1)* \dots$
$P(w_n \mid w_{n-2}, w_{n-1})*P(EOS \mid w_{n-1}\ w_n)$

If a n-gram contains a unknown word,

skip the n-gram (i.e., remove it from the Eq)

oov_num ++;

Some intuition about perplexity

- Given a vocabulary V and assume uniform distribution; i.e., P(w) = 1/ |V|

- The perplexity of any test data T with unigram LM is:

$$PPL(T) = P(T)^{-\frac{1}{N}} = \frac{1}{|V|}^{N*(-\frac{1}{N})} = | V |$$

- Perplexity is a measure of effective "branching factor".

Standard metrics for LM

- Direct evaluation
  - Perplexity

- **Indirect evaluation:**
  - ASR
  - MT
  - …

ASR

- Word error rate (WER):
  - System: And he saw apart of the movie
  - Gold: Andy saw a part of the movie
  - WER = 3/7

Summary

- N-gram LMs:

- Evaluation for LM:
  - Perplexity = $10^{- 1/N * \lg P(T)} = 2^{H(L,P)}$
  - Indirect measures: WER for ASR, BLEU for MT, etc.

Next time

- Smoothing: J&M 4.5-4.9
- Other LMs: class-based LM, structured LM

Additional slides

Entropy

- Entropy is a measure of the uncertainty associated with a distribution.

$$H(X) = -\sum_{x} p(x) \log p(x)$$

- The lower bound on the number of bits it takes to transmit messages.

- An example:
  - Display the results of horse races.
  - Goal: minimize the number of bits to encode the results.

An example

- Uniform distribution: p_i=1/8.

$$H(X) = -8 * (\frac{1}{8}\log_2\frac{1}{8}) = 3\, bits$$

- Non-uniform distribution: (1/2,1/4,1/8, 1/16, 1/64, 1/64, 1/64, 1/64)

$$H(X) = -(\frac{1}{2}\log\frac{1}{2} + \frac{1}{4}\log\frac{1}{4} + \frac{1}{8}\log\frac{1}{8} + \frac{1}{16}\log\frac{1}{16} + 4 * \frac{1}{64}\log\frac{1}{64}) = 2\, bits$$

(0, 10, 110, 1110, 111100, 111101, 111110, 111111)

➔Uniform distribution has higher entropy.

➔MaxEnt: make the distribution as “uniform” as possible.

Cross Entropy

- Entropy: $$H(X) = -\sum_x p(x) \log p(x)$$
- Cross Entropy: $$H_c(X) = -\sum_x p(x) \log q(x)$$
- Cross entropy is a distance measure between p(x) and q(x): p(x) is the true probability; q(x) is our estimate of p(x).

$$H_c(X) \geq H(X)$$

Cross entropy of a language

- The cross entropy of a language L:

$$H(L,q) = -\lim_{n\to\infty} \frac{\sum_{x_{1n}} p(x_{1n}) \log q(x_{1n})}{n}$$

- If we make certain assumptions that the language is “nice”, then the cross entropy can be calculated as:

$$H(L,q) = -\lim_{n\to\infty} \frac{\log q(x_{1n})}{n} \approx -\frac{\log q(x_{1n})}{n}$$

Perplexity

$$PPL(T) = P(T)^{-\frac{1}{N}} = \frac{1}{\sqrt[N]{P(T)}}$$

$$= 2^{-\frac{1}{N} log_2 P(T)}$$

$$= 2^{H(L,P)}$$