Statistical Learning for Text Data Analytics
Language Models

Yangqiu Song

Hong Kong University of Science and Technology

yqsong@cse.ust.hk

Spring 2018

∗Contents are based on materials created by Hongning Wang, Julia Hockenmaier, Dan Jurafsky, Dan Klein, Noah Smith, Slav Petrov, Yejin Choi, and Michael Collins

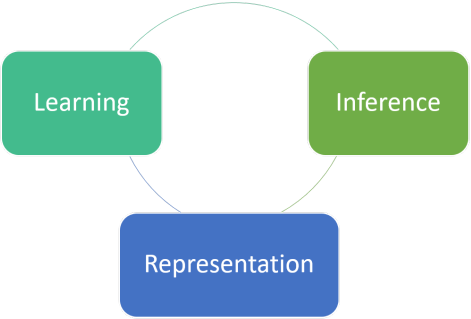
Reference Content

- Noah Smith. CSE 517: Natural Language Processing https://courses.cs.washington.edu/courses/cse517/16wi/
- Julia Hockenmaier. CS447: Natural Language Processing. http://courses.engr.illinois.edu/cs447
- Hongning Wang. CS6501 Text Mining. http://www.cs.virginia.edu/~hw5x/Course/Text-Mining-2015-Spring/_site/
- Dan Jurafsky. cs124/ling180: From Languages to Information. http://web.stanford.edu/class/cs124/
- Dan Klein. CS 288: Statistical Natural Language Processing. https://people.eecs.berkeley.edu/~klein/cs288/sp10/

Reference Content (Cont'd)

- Slav Petrov. Statistical Natural Language Processing. https://cs.nyu.edu/courses/fall16/CSCI-GA.3033-008/
- Chris Manning. CS 224N/Ling 237. Natural Language Processing. https://web.stanford.edu/class/cs224n/
- Yejin Choi. CSE 517 (Grad) Natural Language Processing. http://courses.cs.washington.edu/courses/cse517/15wi/
- Michael Collins. COMS W4705: Natural Language Processing. www.cs.columbia.edu/~mcollins/courses/nlp2011/

Course Topics



- Representation: language models, word embeddings, topic models
- Learning: supervised learning, semi-supervised learning, sequence models, deep learning, optimization techniques
- Inference: constraint modeling, joint inference, search algorithms

NLP applications: tasks introduced in Lecture 1

Overview

Language Model Evaluation

- Train the models on the same training set
  - Parameter tuning can be done by holding off some training set for validation
- Test the models on an unseen test set
  - This data set must be disjoint from training data
- Language model A is better than model B
  - If A assigns higher probability to the test data than B

Measuring Model Quality

- The goal isn't to pound out fake sentences!
  - Obviously, generated sentences get "better" as we increase the model order
  - More precisely: using ML estimators, higher order is always better likelihood on train, but not test
- What we really want to know is:
  - Will our model prefer good sentences to bad ones?
  - Bad $\neq$ ungrammatical!
  - Bad $\approx$ unlikely
  - Bad $=$ sentences that our model really likes but aren't the correct answer

Measuring Model Quality (Cont'd)

- The Shannon Game (by Claude Shannon, 1916–2001):
  - How well can we predict the next word?

When I eat pizza, I wipe off the ______

grease	0.5
sauce	0.4
dust	0.05
...	
mice	0.0001
...	
the	$1e-100$

  - Unigrams are terrible at this game.
- How good are we doing?
  - Compute per word log likelihood (N words, M test sentences S_i):
  - An intuitive way: $l = \frac{1}{N}\sum_i^N \log P(S_i)$

Perplexity

- Standard evaluation metric for language models
  - A function of the probability that a language model assigns to a data set
  - Rooted in the notion of cross-entropy in information theory

Perplexity

- Perplexity of a probability distribution

$$2^{H(P)} = 2^{-\sum_x P(x) \log_2 P(x)}$$

  - $H(P)$: entropy
  - Perplexity of a random variable X may be defined as the perplexity of the distribution over its possible values x
  - In the special case where P models a uniform distribution over k discrete events, its perplexity is k
- Perplexity of a probability model

$$2^{H(\hat{P},Q)} = 2^{-\sum_x \hat{P}(x) \log_2 Q(x)}$$

  - $H(\hat{P}, Q)$: cross entropy
  - $\hat{P}$ denotes the empirical distribution of the test sample (i.e., $\hat{P}(x) = n/N$ if x appeared n times in the test sample of size N)
  - Q: a proposed probability model
  - One may evaluate Q by asking how well it predicts a separate test sample $x_1, x_2, ..., x_N$ also drawn from unknown P

The Shannon Game Intuition for Perplexity

- How hard is the task of recognizing digits "0,1,2,3,4,5,6,7,8,9" at random
  - Perplexity 10
- How hard is recognizing (30,000) names at random
  - Perplexity 30,000
- If a system has to recognize
  - Operator (1 in 4)
  - Sales (1 in 4)
  - Technical Support (1 in 4)
  - 30,000 names (1 in 120,000 each)
  - Perplexity is 53
- Perplexity is weighted equivalent branching factor

Perplexity as Branching Factor

- Language with higher perplexity means the number of words branching from a previous word is larger on average
- The difference between the perplexity of a language model and the true perplexity of the language is an indication of the quality of the model

Perplexity Per Word for Language Models

- Given a test corpus with N tokens, $w_1, \ldots, w_N$, and an n-gram model $P(w_i|w_{i1}, \ldots, w_{in+1})$ the perplexity $PP(w_1, \ldots, w_N)$ is defined as follows (Brown et al. (1992)):
- The inverse of the likelihood of the test set as assigned by the language model, normalized by the number of words

$$
\begin{aligned}
PP(w_1, \ldots, w_N) &= P(w_1, \ldots, w_N)^{-\frac{1}{N}} \\
&= \sqrt[N]{\frac{1}{P(w_1, \ldots, w_N)}} \\
&= \sqrt[N]{\frac{1}{\prod_{i=1}^{N} P(w_i|w_1, \ldots, w_{i-1})}} \; (chain\ rule) \\
&= \sqrt[N]{\frac{1}{\prod_{i=1}^{N} P(w_i|w_{i-1}, \ldots, w_{i-n+1})}} \; (n-gram\ model)
\end{aligned}
$$

- Minimizing perplexity = maximizing probability!
- Language model LM_1 is better than LM_2 if LM_1 assigns lower perplexity (= higher probability) to the test corpus $w_1, \ldots, w_N$
- Note: the perplexity of LM_1 and LM_2 can only be directly compared if both models use the same vocabulary.

Practical Issues

- Since language model probabilities are very small, multiplying them together often yields to underflow
- It is often better to use logarithms instead, so replace

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

with

$$PP(w_1, \ldots, w_N) = \exp\left(-\frac{1}{N} \sum_{i=1}^{N} \log P(w_i | w_{i-1}, \ldots, w_{i-n+1}) \right)$$

An Experiment

- Models
  - Unigram, bigram, trigram models (with proper smoothing)
- Training data
  - 38M words of WSJ text (vocabulary: 20K types)
- Test data
  - 1.5M words of WSJ text
- Results

	Unigram	Bigram	Trigram
Perplexity	962	170	109

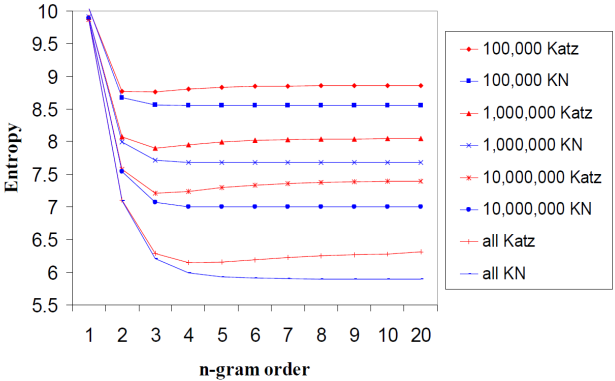
- Conclusion: The bigram is much better than the unigram, and the trigram is even better

What Actually Works?

- Trigrams and beyond
  - Unigrams, bigrams generally useless for speech or machine translation
  - Trigrams much better (when there's enough data)
  - 4-, 5-grams really useful in MT, but not so much for speech
- Discounting
  - Absolute discounting, Good-Turing, held-out estimation, Witten-Bell, etc.
- See Chen and Goodman (1996) reading for tons of graphs

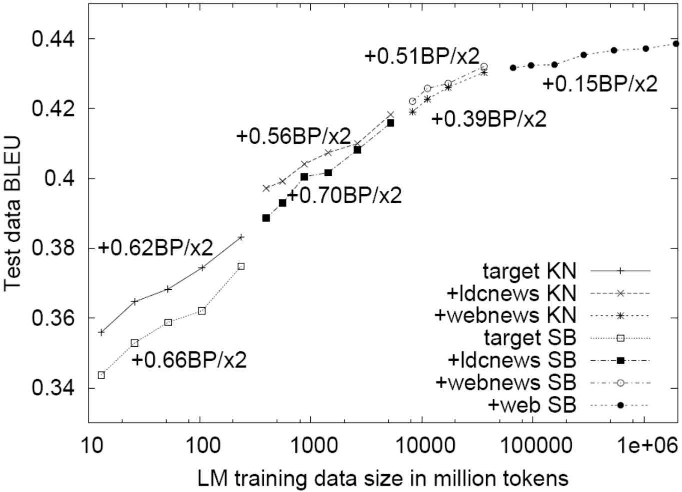
Data vs. Method?

- Having more data is better...
- ...but so is using a better estimator
- Another issue: $n > 3$ has huge costs in speech recognizers



Tons of Data?

- Tons of data closes gap, for extrinsic MT evaluation



Further Reading

- Manning et al. (2008). Introduction to information retrieval. Chapter 12: Language models for information retrieval.
- Jurafsky and Martin (2017). Speech and Language Processing. Chapter 4: N-Grams. `https://web.stanford.edu/~jurafsky/slp3/`
- Chen and Goodman (1996). An empirical study of smoothing techniques for language modeling.
- Collins (2011). Course notes for COMS w4705: Language modeling, 2011. `http://www.cs.columbia.edu/~mcollins/courses/nlp2011/notes/lm.pdf`
- Zhu (2010). Course notes for cs769: Language modeling, 2011. `http://pages.cs.wisc.edu/~jerryzhu/cs769/lm.pdf`

References

Brown, P. F., Pietra, V. J. D., Mercer, R. L., Pietra, S. A. D., and Lai, J. C. (1992). An estimate of an upper bound for the entropy of english. *Comput. Linguist.*, 18(1):31–40.

Chen, S. F. and Goodman, J. (1996). An empirical study of smoothing techniques for language modeling. In *ACL*, pages 310–318.

Collins, M. (2011). Course notes for coms w4705: Language modeling. Technical report, Columbia University.

Jurafsky, D. and Martin, J. H. (2017). *Speech and Language Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Zhu, X. J. (2010). Course notes for cs769: Language modeling. Technical report, University of Wisconsin-Madison.