Ling/CSE 472: Introduction to Computational Linguistics

4/20/17
N-grams

Overview

- What are N-grams? (high-level)
- When are they useful?
- Evaluation
- Simple (un-smoothed) N-grams
- Training and test sets
- Unknown words
- N-grams and linguistic knowledge
- Reading questions
- Next time: smoothing, back-off, interpolation

What are N-grams?

- A way of modeling the probability of a string of words.
- ... or the probability of the N+1st word being *w* given words 1-N.

$$P(w_n|w_1^{n-1}) \approx P(w_n|w_{n-1})$$

- We'll come back to this in more detail in a moment

When are they useful?

- When would you want to know the relative probability of two strings?
- ... the relative probability of two words given the previous N words?
- ... the absolute probability of a string?

When are they useful?

- When would you want to know the relative probability of two strings?
  - Choosing among ASR candidates
  - Choosing among MT candidates
- ... the relative probability of two words given the previous N words?
  - Choosing among spell correction candidates
  - Predictive text (T9, AAC): Does your cell phone store an n-gram model?
- ... the absolute probability of a string? (kind of)
  - Spell checking: relative to a threshold
  - Language ID, authorship ID: relative to probability from some other model

Evaluating N-gram models

- What kinds of extrinsic evaluation are possible?
- What kinds of intrinsic evaluation are possible?
- What different kinds of models could you compare?

Evaluating N-gram models

- What kinds of extrinsic evaluation are possible?
    - ASR, MT, ...
- What kinds of intrinsic evaluation are possible?
    - Perplexity: Given an n-gram model trained on some training set, how well does it predict the test set? (i.e., what probability does it assign to the test set?)
- What different kinds of models could you compare?
    - Different: training data, smoothing/back-off techniques, higher-level tokens

Training and test sets

- Training and test sets must be distinct, otherwise probabilities will be artificially high

- This is just a special case of a more general reason: The purpose of test sets is to see if the method generalizes to unseen data

- Training and test sets are typically drawn from the same or comparable corpora

  - Why?

  - What if they aren’t?

Simple (un-smoothed) N-grams

- What we'd really like to calculate:

$$P(w_1^n) = P(w_1)P(w_2|w_1)P(w_3|w_1^2)\dots P(w_n|w_1^{n-1})$$

$$= \prod_{k=1}^{n} P(w_k|w_1^{k-1})$$

- But we'd never find a corpus big enough.

- Why not?

So: An approximation

- Markov assumption: The probability of a given word only depends on the previous word

$$P(w_n|w_1^{n-1}) \approx P(w_n|w_{n-1})$$

- Is this assumption valid?

- (NB: That's a bigram model ... for a trigram model, we look at the previous two words, etc.)

Maximum Likelihood Estimates for bigram counts

- Bigram probability for a word *y* given a previous word *x*:

- Out of all the times you saw *x*, in what percentage was it followed by *y*?

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

Probability v. Frequency

- Probability: How likely something is to happen
- Frequency: How frequently something has happened in a set of observations
- Probability clearly influences frequency
- Frequency can be used to estimate probability (what else can?)
    - ... but they are not the same thing
- If a bigram never appears in a training corpus,
    - What is its observed frequency?
    - What is its probability?

Example

- What are the bigrams in the following mini corpus? What are their MLEs?

<s> How much wood would a wood chuck chuck if a wood chuck could chuck wood? </s> <s> As much wood as a wood chuck could if a wood chuck could chuck wood. </s>

- What probability does that bigram model assign to the following sentences?

<s> How much wood. </s>
<s> How much wood? </s>
<s> As much wood chuck chuck chuck wood. </s>
<s> How would a wood chuck chuck ? </s>

bigrams

- <s> How = 1/2
- How much = 1
- much wood = 1
- wood would = 1/8
- would a = 1
- a wood = 1
- wood chuck = 1/2
- chuck chuck = 1/7
- chuck if = 1/7
- if a = 1
- chuck could = 3/7
- could chuck = 2/3
- chuck wood = 2/7
- wood ? = 1/8
- ? </s> = 1
- <s> As = 1/2
- As much = 1/2
- wood as = 1/8
- as a = 1/2
- could if = 1/3
- wood . = 1/8
- . </s> = 1

Sentences

<s> How much wood. </s>
<s> How much wood? </s>
<s> As much wood chuck chuck chuck wood. </s>
<s> How would a wood chuck chuck ? </s>

1. 1/2 * 1 * 1 * 1/8 * 1 = 1/16
2. 1/2 * 1 * 1 * 1/8 * 1 = 1/16
3. 1/2 * 1/2 * 1 * 1/2 * 1/7 * 1/7 * 2/7 * 1 = 1/1372
4. 1/2 * 0 ... = 0

Counting things in a corpus

- Type/token distinction

- But what counts as a token? What are some cases where this is not obvious?

- And what counts as the same type? What are some cases where this is not obvious?

- Is there a single right answer?

Counting things in a corpus

- Type/token distinction
- But what counts as a token? What are some cases where this is not obvious?
    - Contracted forms, punctuation, hyphenated forms, words with spaces (*New York*), ...
- And what counts as the same type? What are some cases where this is not obvious?
    - Caps/non-caps, word-form/lemma, homographs, ...
- Is there a single right answer?
    - No: It depends on the application context

Unknown words

- What would a n-gram model trained as described so far say about the probability of a sentence with an unknown word in it?

- What could be done about that?

Unknown words

- What would a n-gram model trained as described so far say about the probability of a sentence with an unknown word in it? -- 0

- What could be done about that?

- Choose a vocabulary smaller than the actual V, and replace all other words with <UNK> -- Any words you might want to be sure to include in V?

- Or: Replace the first occurrence of every word in the training set with <UNK>

- Then: Estimate probabilities of <UNK> just like any other word

N-grams and linguistic knowledge

- Is an n-gram model a grammar?
- What kinds of information about a language does it capture?
- What kinds of information about a language does it miss?

N-grams and linguistic knowledge

- N-gram models are supposed to be “language-independent” in that they don’t require specific knowledge about a language to create --- just text.

- Would you expect them to work equally well across languages? Why or why not?

Overview

- What are N-grams? (high-level)
- When are they useful?
- Evaluation
- Simple (un-smoothed) N-grams
- Training and test sets
- Unknown words
- N-grams and linguistic knowledge
- Reading questions
- Next time: smoothing, back-off, interpolation

Reading questions

- It is necessary to have dev set, which is to validate and check our model, when we are developing our N-gram models? Can we look at the test set for validation occasionally as validation, or we shuffle everything to do a cross-validation?

- Why do we bother generating a matrix that contains the word counts/prob for n-grams if most of the matrix is so sparse? Is there ever a situation where you care about the probability functions of arbitrary P\left(w_a|w_1w_2...w_n \right)P(wa|w1w2...wn)? It seems like for most applications, such as generating facsimile sentences, you would only care about n-grams which you have encountered before.

Reading questions

- I'm intrigued by the different perplexity values of the WSJ test set given on page 97. Is there a general estimate for how the perplexity corresponding to an N-gram model changes with N? And what features of a natural language or corpus might cause it to have a high or low perplexity according to some N-gram model?

- Is the perplexity calculated for subsets of text (such as for an individual sentence), or for the entire test set at once (since it's normalized anyway). Can you take the average of the perplexities of a subset of sentences to approximate the perplexity of the entire test set?

- What does "normalization" mean?

Reading questions

- How does knowing the perplexity vs the probability (which are inverse to each other, right?) benefit us in NLP? Also, can you clarify the difference between branching factor and perplexity? I understand how entropy would be related, but the example they give on page 97 about numbers didn't clear it up for me.

- To evaluate a language model, it sounds like we often test on a body of coherent, sensible text. Is testing ever done with on a large scale with incoherent gibberish?

Reading questions

- In the reading we are introduced to N-grams which look back at a "history string" to predict the probability of the single next string. This "history string" varies with respect to the size of the N-gram. Is there ever the case in which an n-gram model endeavors to predict beyond one single element? This seems unwieldy, but could conceivably operate well if limited to function words.

- Also, are n-gram models ever dynamic in size? That is, would there ever be a case in which an adjustment of scope could take place automatically to facilitate higher accuracy?

- In some cases, the probability of a word showing up might be more dependent on the word two(or some other number > 1) before it instead of right before it. Is it possible/useful to try and learn "weights" for words// positions, so that we get this information?

Reading questions

- With the MLE methods mentioned, MLE is a way to maximize the likelihood of the model. Is this then saying that we are trying to maximize the likelihood that the surrounding words appear near the word of interest?

- This section mentions that any unknown words are replaced by the token <UNK> and then the probabilities are estimated from the number of times <UNK> appears. If this OOV word is just one of many, by replacing it and others with the same token wouldn't that cause an issue?

Reading questions

- Also, what are the different meanings of uh and um (pg 85)?
- Why does treating uh as a word improves next-word prediction?
- In practice, what is the degree of N-grams used in modern day spell checking and auto-completion (e.g. 4-grams, 5-grams)?