Introdução a NLP e IR

Language Models

Alexandre Rademaker[1]

FGV/EMAp

August 23, 2022

[1]Olga Zamaraeva, University of Washington

Lecture goals

- Define what language models are
- LMs and knowledge
- Consider N-gram models as the simplest LM
  - Cheap but rigid
  - The simple math behind N-gram models
  - Intrinsic evaluation of N-grams (perplexity)
- Preview of neural models
  - Flexible but expensive
  - High-level architecture (input, output)
  - Mention word embeddings as the "byproduct" of neural models

Language Models: It's all about sequences

"A grammar is better, but in practice people use language models."

D. Jurafsky

"You are uniformly charming!" cried he, with a smile of associating and now and then I bowed and they perceived a chaise and four to wish for.

Generated by a trigram LM trained on Austen's books

"What comes out of a 4-gram model of Shakespeare looks like Shakespeare because it is Shakespeare."

D. Jurafsky

But wait!

Haven't we been studying "language models" all along? (FSA, FST?)

- Yes and no.
- We've been modeling languages with FSA:
  - Regular languages
  - Morphology and phonology
  - Based on linguistic knowledge
- *Language Models* is also a **term** denoting a particular kind of statistical models
  - We now begin to talk about modeling *syntax**
  - (though not really; it is about *word order* which is a shallower notion than syntax)

Language Models

London is the capital of ...

- Language models are programs which output the most probable word given some context

Language Models

London is the capital of ...

- Language models are programs which output the most probable word given some context
  - That's it!

Language Models

London is the capital of ...

- Language models are programs which output the most probable word given some context
  - That's it!
- They output a probability *distribution* over the vocabulary

Language Models

London is the capital of ...

- Language models are programs which output the most probable word given some context
  - That's it!
- They output a probability *distribution* over the vocabulary
- ... **N-grams** are the simplest LM: what's the most probable **next** word given a sequence?
  - E.g. how likely is it that the next word after *London is the capital of* is *England*? *fashion*? *swims*?
- How would a language model rank the probabilities?

Statistical Language Models (LM)

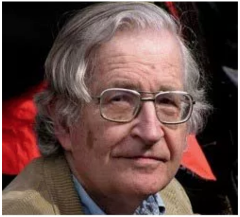
- A notion of modeling language based on e.g. word frequencies
- Count how many times you saw X follow Y
- Now, predict that X will follow Y *with some probability*
  - *London is the capital of* can be plausibly followed by *England, the, fashion...*
  - LM is somewhat creative (the seed *was the capital of* may result in a different highest ranked predicted word)
- LMs proved to be useful
- But are they meaningful?

LMs and linguistic knowledge

- Statistical and neural LMs are very successful in NLP
- They capture some **surface** information about the language (including the “world knowledge” that is on the surface)
- What about deeper structure, explanations, reasons of phenomena?

(Beyond LMs) The role of statistics

`https://www.tor.com/2011/06/21/norvig-vs-chomsky-and-the-fight-for-the-future-of-ai/` (Kevin Gold's overview)



Chomsky: *To produce a statistically based simulation of ... a [bee] dance without attempting to understand why the bee behaved that way... is ...a notion of [scientific] success that's very novel. I don't know of anything like it in the history of science.*

The role of statistics

https://www.tor.com/2011/06/21/norvig-vs-chomsky-and-the-fight-for-the-future-of-ai/ (Kevin Gold's overview)

Norvig: *Engineering success correlates with scientific success*

When are LMs useful?

- Close captioning (Automatic Speech Recognition)
- Easier communication during travel (Machine Translation)
- Spelling correction and predictive text
- Document classification

Main idea behind LMs

- The LM is *trained* on a corpus and can then assign probabilities to new, *test* sentences
- Train by estimating actual probabilities of word sequences from actual corpora
- Then, deploy the model to:
    - classify documents in terms of: topic, style, authorship...
        - (which is closer to which model? Is this more like Plato or more like Aristotle?)
        - (which model says the text is more *probable*, according to it?)
    - generate *new* text

N-grams: The (simplified) math behind the simplest LM

E.g. what probability will a LM trained on corpus TC assign to the sentence:

"London is the capital of England"
In corpus TC, how many times did we see *England* after *London is the capital of*?

$$\frac{C(London, is, the, capital, of, England)}{C(London, is, the, capital, of)}$$

N-grams: The simplest LM

London is the capital of England

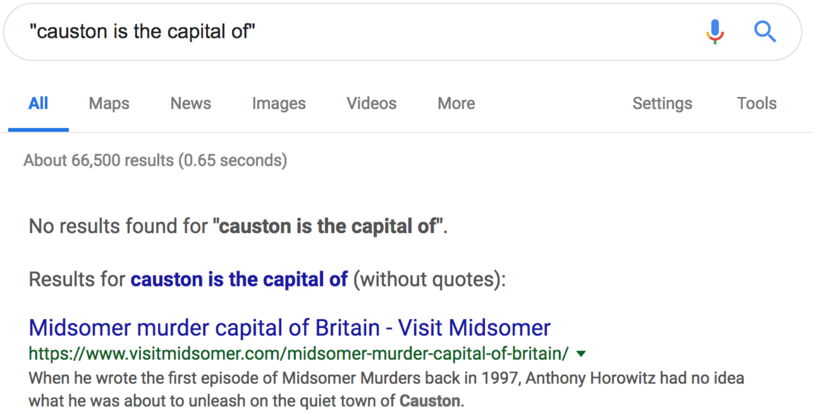
- What we'd like to calculate:

$$\frac{C(London, is, the, capital, of, England)}{C(London, is, the, capital, of)}$$

- In some cases, it is possible (using e.g. the web)
- But in most cases, we'd never find a corpus big enough
  - E.g. What if I want to know the probability of the sentence *Causton is the capital of murder in England*?

N-grams: Zero counts

Often not **possible** to compute joint probabilities directly:



Markov assumption

Andrey Markov (1856-1922)
(Not-so-fun-fact: In 1908, Markov was fired from the University for refusing to spy on his students)

- Markov assumption: The probability of a word given a sequence only depends on **a few** previous words, not the entire sequence
- *Approximate* the history given the last (few) word(s)
  - P(murder|of), P(of|capital) instead of P(murder|capital of)
  - Will it help me if my corpus does not contain the word *Causton*?

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

N-gram: bigger N means closer approximation

- P(England |London is the capital of)
  - P(England |of) – **bigram**
  - P(England |capital of) – **trigram**
  - P(England |the capital of)
  - P(England |is the capital of)

- Imagine new texts generated by the different models
- Is it more useful to be stuck with *capital* or *is the capital of*?

Small N = "silly" model, big N = rigid model

N-gram: bigger N means closer approximation

Consider *generating* from such models:

- P(him |Alas poor Yorick I knew)
  - P(him |knew) – **bigram**
  - P(I |knew him) – **trigram**
  - P(Yorik |I knew him)
  - P(poor |Yorick I knew him)
  - P(Alas |poor Yorick I knew him)

Small N = "silly" model, big N = rigid model (how interesting is it to generate exact strings from Shakespeare's *Hamlet*?)

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?

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

Small example (from Jurafsky&Martin 2008)

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

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

¡s¿ I am Sam ¡/s¿
¡s¿ Sam I am ¡/s¿
¡s¿ I do not like green eggs and ham ¡/s¿

P(do | I) =

(Respond at: `https://pollev.com/olgazamaraev657`)

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?

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?
  - To not allow 0 probabilities, anticipate an *UNK* word in the vocabulary, assign it some small probability, redistribute the rest of the probabilities so that all probabilities still sum to 1
    - "Smoothing" (and it does not come for free)
    - Next lecture

Probability vs. 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
  - ... 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?

Exercise: Bigger 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 = 1/13
4. 1/2 * 0 ... = 0

Generating from a N-gram model

"i had called upon my friend , mr . sherlock holmes , which i should ever communicate to the public ."

- Start with a **seed** sequence of length N
- The model outputs the most probable word given the seed
- Now the last N-1 words from the seed plus the freshly output word become the **history**
- The model outputs the most probable word given history
- etc.

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

Evaluating N-gram models

- What kinds of extrinsic evaluation are possible?
- What kinds of intrinsic evaluation are possible?

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?)

Perplexity (intrinsic evaluation)

- Which model assigns the **highest probability** to the test set?
- *Perplexity (PP)* is the inverse probability normalized by word count
  - Informally, how "surprised" is the model by the test set?
  - Information theory
- E.g. for a test set $W = w_1 w_2 w_3 \ldots w_N$

$$PP(W) = P(w_1 w_2 w_3 \ldots w_N)^{\frac{-1}{N}} = \left(\prod_{i=1}^{N} P(w_i | w_1 \ldots w_{i-1})\right)^{\frac{-1}{N}}$$

$$\approx \left(\prod_{i=1}^{N} P(w_i | w_{i-1})\right)^{\frac{-1}{N}}$$

Perplexity

- Perplexity can bee seen as an average *branching factor* of a language
- e.g. consider a language of digits where each digit has a probability of 0.1 of following another digit

Is this high perplexity?

Perplexity

- Perplexity can bee seen as an average *branching factor* of a language
- e.g. consider a language of digits where each digit has a probability of 0.1 of following another digit

$$
\begin{aligned}
\mathrm{PP}(W) &= P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}} \\
&= \left(\frac{1}{10}^{N}\right)^{-\frac{1}{N}} \\
&= \frac{1}{10}^{-1} \\
&= 10
\end{aligned}
$$

Is this high perplexity?

Other varieties of statistical LMs

- Hidden Markov Models
  - were widely used in ASR
- Probabilistic CFGs
  - Assign probabilities to sequences of "constituents"
- ...all of these have similar limitations as n-grams
  - (either approximate too little or too much)

Desireable: Generalizing over contexts

- *London* is the capital of...
- *Causton* is the capital of...

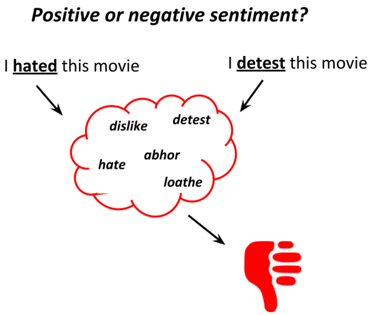


Figure from Allyson Ettinger's tutorial at SCiL 2019

Neural* language models

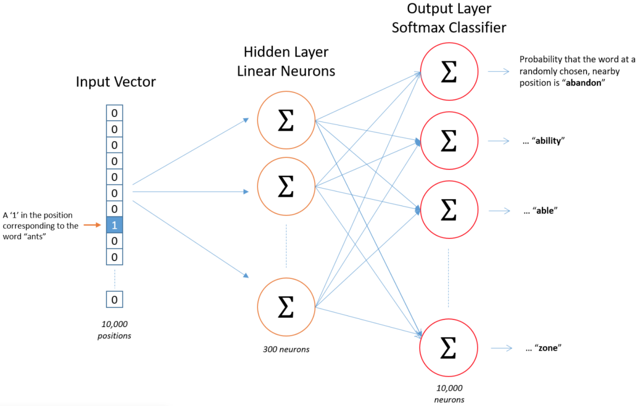
- Predict the word given context (or vice versa)
- Generalize over contexts, are more "creative" than n-grams:
  - Learn which words occur in similar contexts
  - It is possible to build a neural model that creates representations for unknown words "on the fly"**
- But:
  - Are more complex to train
  - Require lots of training data to start working well
  - Learn the training data biases

*These are *simplified* neural architectures

**Not the same architecture as in the lecture

(Simplified) neural models architecture

- The *feed-forward* SkipGram model (Mikolov et al)
- Input: a word from the vocabulary
- Middle: two matrices and some matrix multiplication
- Output: a probability for each word in the vocabulary occurring *somewhere nearby* the input word



Pic from: `http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/`

What you need to know

- What are N-grams?
- When are they useful?
- Simple (un-smoothed) N-grams
- Perplexity (relationship to probability and what for)
- (Next time) Unknown words, Smoothing, back-off, interpolation