CS565: Intelligent Systems and Interfaces

Lecture: Language Modeling

24th Jan, 2017

Semester: Jan – May 2017

Ashish Anand

IIT Guwahati

Recap and Moving Forward

- Until Now
  - Sentence segmentation, Tokenization
  - Collocation

- Next
  - Language Modeling: Generative model of language

Objective

- Understanding Language Model
- N-Gram Language Model
- Evaluating Language Model

Lets look at some examples

- Predicting next word
  - I am planning ……..
- Speech Recognition
  - I saw *a van*   vs I saw *a 7*

Example continued

- Spelling correction
  - Study was conducted *by* students vs study was conducted *be* students
  - *Their* are two exams for this course vs *There* are two exams for this course

- Machine Translation
  - I have asked him to do homework
  - मैंने उससे पूछा कि होमवर्क करने के लिए
  - मैंने उसे होमवर्क करने के लिए कहा

My objective was

- To find next probable word
- To find which sentence is more likely to be true

But it must be recognized that the notion “probability of a sentence” is an entirely useless one, under any known interpretation of this term.

Noam Chomsky

Anytime a linguist leaves the group the recognition rate goes up.

Fred Jelinek (then of the IBM speech group)

Language Models (LM)

- Models assigning probabilities to a sequence of words

- P(I saw a van) > P(I saw a 7)

- P(मैंने उससे पूछा कि होमवर्क करने के लिए) < P(मैंने उसे होमवर्क करने के लिए कहा)

Defining LM Formally

- We consider a *vocabulary,* a finite set denoted as $\mathcal{V}$, and a function $P(w_1, w_2, w_3, \ldots, w_n)$, such that
  - For any $<w_1, w_2, w_3, \ldots, w_n> \in \mathcal{V}^\dagger$, $p(w_1, w_2, w_3, \ldots, w_n) \geq 0$
  - $\Sigma\, p(w_1, w_2, w_3, \ldots, w_n) = 1,$

where $\mathcal{V}^\dagger : \{S = w_1 w_2 w_3 \ldots w_n \mid w_i \in \mathcal{V}\}$.

Estimating $P(w_1, w_2, .., w_n)$

- Our task is to compute
  P(I, am, fascinated, with, recent, advances, in, AI)

- Chain Rule

Estimating $P(w_1, w_2, .., w_n)$

- Chain Rule
  - $P(w_1, w_2, w_3, ... , w_n) = P(w_1)\ P(w_2|w_1)\ P(w_3|w_1, w_2)\ ....\ P(w_n|w_1,.., w_{n-1})$

Estimating $P(w_n | w_1,.., w_{n-1})$

- Could we just count and divide?

$$P(eat|I\ want\ to) = \frac{count(I\ want\ to\ eat)}{count(I\ want\ to)}$$

- What is the problem here?

Estimating $P(w_n | w_1,.., w_{n-1})$

- Too many possible sentences
- Data sparseness
- Poor *generalizability*

Markov Assumption

- Simplifying assumption:

$$P(eat \mid I\ want\ to) \sim P(eat \mid to)$$

or

$$P(eat \mid I\ want\ to) \sim P(eat \mid want\ to)$$

Markov Assumption

$$P(w_1, w_2, w_3, \ldots., w_n) \sim \prod_i P(w_i | \, w_{i-k}, \ldots., w_{i-1})$$

i.e., Each component in the product is getting approximated by Markov assumption

$$P(w_i | w_1, w_2, w_3, \ldots., w_{i-1}) \sim P(w_i \mid w_{i-k}, \ldots., w_{i-1})$$

N-gram Models

- Unigram: Simplest Model (does not depend on anything)

$$P(w_1, w_2, w_3, \ldots., w_n) \sim \prod_i P(wi)$$

- Bigram Model (1st Order Markov model)

$$P(w_1, w_2, w_3, \ldots., w_n) \sim \prod_i P(w_i | w_{i-1})$$

- Trigram Model (2nd order Markov model)

$$P(w_1, w_2, w_3, \ldots., w_n) \sim \prod_i P(w_i | w_{i-2}, w_{i-1})$$

N-gram Model: Issue

- Long-distance dependencies

“The computer which I had just put into the lab on the fifth floor crashed”

Estimating the Probabilities

Data

- Training
- Development
- Test

Maximum Likelihood Estimate

- Unigram

$$P(w_i) = \frac{c(w_i)}{K} \;;$$

$$K\text{:}\, Total\ number\ of\ \boldsymbol{tokens}\ in\ training\ set$$

- Bigram

$$P(w_i|\, w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$$

- N-Gram

$$P(w_n|w_{n-N+1}^{n-1}) = \frac{c(w_{n-N+1}^{n-1}\, w_n)}{c(w_{n-N+1}^{n-1})}$$

Bigram Probabilities

eat on	.16	eat Thai	.03
eat some	.06	eat breakfast	.03
eat lunch	.06	eat in	.02
eat dinner	.05	eat Chinese	.02
eat at	.04	eat Mexican	.02
eat a	.04	eat tomorrow	.01
eat Indian	.04	eat dessert	.007
eat today	.03	eat British	.001

A fragment of bigram probabilities from the *Berkeley Restaurant Project* showing most likely word to follow *eat*

Source: Figure 6.2: Page 225, SLP

Computing probability of a sentence

<s> I	.25	I want	.32	want to	.65	to eat	.26	British food	.60
<s> I'd	.06	I would	.29	want a	.05	to have	.14	British restaurant	.15
<s> Tell	.04	I don't	.08	want some	.04	to spend	.09	British cuisine	.01
<s> I'm	.02	I have	.04	want thai	.01	to be	.02	British lunch	.01

Figure 6.3 More fragments from the bigram grammar from the Berkeley Restaurant Project.

P (<s> I want to eat British food </s>) = P(I|<s>) P(want|I) P(to|want) P(eat|to) P(British|eat) P(food|British) P(</s>|food)

Language Model Evaluation

Two paradigms

- Intrinsic evaluation

- Extrinsic evaluation

Intrinsic Evaluation: Perplexity

- Given a test data of *m* sentences: $s_1, s_2, \ldots\ldots, s_m$
- Probability of a sentence under this model $p(s_i)$
- Log-Probability of all sentences: $\log \prod p(s_i) = \sum \log p(s_i)$

Perplexity: Alternate definitions

- Perplexity = 2^{-l} , where l = 1/M($\sum \log p(s_i)$)
- Perplexity = $P(s_1 s_2 \ldots\ldots s_n)^{-(1/M)}$
- Smaller the value of perplexity, better the language model is.

Interpreting Perplexity

- Weighted average branching factor

- Branching factor: number of possible next words that can follow any word.

One specific example

- Training: 38 million words from *Wall Street Journals* [vocab size: 19,979]

- Test: 1.5 million words

	Unigram	Bigram	Trigram
Perplexity	962	170	109

Generalization

- 1 gram: Hill he late speaks; or! a more to leg less first you enter
- 2 gram: What means, sir. I confess she? then all sorts, he is trim, captain
- 3 gram: This shall forbid it should be branded, if renown made it empty
- 4 gram: It cannot be but so.

Source: SLP (3rd Ed.), Figure 4.3. Training data on Shakespeare's works. V = 29, 066.

Generalization

- 1 gram: Months the my and issue of year foreign …..
- 2 gram: Last December through the way to preserve the Hudson ….
- 3 gram: They also point to ninety nine point six billion dollars from two ……

Source: SLP (3rd ed.), Figure 4.4. Training data on 40 million words of Wall Street Journal

Unknown words

- Fix vocabulary and words within training data not appearing in vocabulary are mapped to <UNK>
- Less frequent words mapped to <UNK>

Sparsity

- Works well if test corpus is very similar to training, which is not generally the case.
- Training Set

  …… denied the allegations

  …… denied the reports

  …… denied the claims

  …… denied the request
- Test Set

  …. denied the offer

  …. denied the loan

P(“offer” | denied the) = 0

Smoothing

- Next Lecture

Reference

- SLP (3^{rd} Ed.) , Chapter 4
- Collin's lecture-notes on Language Modeling