Language Models

Sandeep Soni

10/23/2023

Class Logistics

- Proposal feedback will be made available today
- Hw8 is due on Wednesday

Classification

x	y
The movie was fantastic	😀
The movie could have been better but wasn't bad	😐
That movie was a waste of time and money	😡

Classification

x	y
The movie was fantastic	😀
The movie could have been better but wasn't bad	😐
That movie was a waste of time and money	😡

- In classification problem, our objective is to model the probability of the label given the text i.e. P(y|x)

What is the probability of a given piece of text i.e. P(x)?

Why should we model P(x)?

- NLP perspective: many tasks are sequence-to-sequence tasks
  - Machine translation: text in source language -> text in target language
  - Summarization: long text -> short text

Why should we model P(x)?

- A system should be “fluent” in generating output sequences
- How can we quantify fluency?

El cafe negro me gusta mucho

Machine translation system I

The coffee black me pleases much

El cafe negro me gusta mucho

Machine translation system II

I love dark coffee

Machine translation system I

The coffee black me pleases much

P(“The coffee black me pleases much”)

<

P(“I love dark coffee”)

Machine translation system II

I love dark coffee

“When I look at an article in Russian, I say: `This is really written in English but it has been coded in some strange symbols. I will now proceed to decode.”

—Warren Weaver (1955)

Noisy channel model

$$P_{\texttt{engish}\leftarrow\texttt{spanish}}(\mathbf{w}^e \mid \mathbf{w}^s) \propto P_{\texttt{engish,spanish}}(\mathbf{w}^e, \mathbf{w}^s)$$

$$= \underbrace{\boxed{P_{\texttt{spanish}\leftarrow\texttt{english}}(\mathbf{w}^{(s)} \mid \mathbf{w}^{(e)})}}_{\text{Translation model}} \times \underbrace{\boxed{P_{\texttt{english}}(\mathbf{w}^{(e)})}}_{\text{Language model}}$$

We want the noisy channel model to have high fidelity to the source text and high fluency in the generated output

Language model

- $\mathscr{V}$ is a vocabulary of symbols that need to be generated
- $\mathscr{S}$ is a infinite set of sequences of symbols taken from $\mathscr{V}$; each sequence ends with a special token STOP
- A sequence $x \in \mathscr{S}$

Language model

- $P(x) = P(x_1, x_2, \ldots, x_n)$
- P("I will be back") = P("I", "will", "be", "back", STOP)
- $0 \leq P(x) \leq 1$
- $\sum_{x' \in \mathcal{S}} P(x') = 1$

Language model

- Language modeling is a task of estimating a probabilistic model over words i.e. P(x)
- This is not easy!
- What is the P(“to be or not to be”)

Chain Rule

$$P(x_1, x_2, \ldots, x_n) = P(x_1)$$

$$\times P(x_2 | x_1)$$

$$\times P(x_3 | x_1, x_2)$$

$$\ldots$$

$$\times P(x_n | x_1, \ldots, x_{n-1})$$

$$P(\texttt{to be or not to be}) = P(\texttt{to})$$

$$\times P(\texttt{be} | \texttt{to})$$

$$\times P(\texttt{or} | \texttt{to}, \texttt{be})$$

$$\ldots$$

$$\times P(\texttt{be} | \texttt{to}, \texttt{be}, \ldots, \texttt{to})$$

Chain Rule

$$P(\texttt{to be or not to be}) = P(\texttt{to})$$

$$\times P(\texttt{be}|\texttt{to})$$

$$\times P(\texttt{or}|\texttt{to}, \texttt{be})$$

$$\ldots$$

$$\times P(\texttt{be}|\texttt{to}, \texttt{be}, \ldots, \texttt{to})$$

P(“be” | “to be or not to”) is hard to estimate unless we make some assumptions

Markov assumptions

$$P(x_i \mid x_{i-1}, \ldots, x_1) \approx P(x_i)$$

zero-order

$$P(x_i \mid x_{i-1}, \ldots, x_1) \approx P(x_i \mid x_{i-1})$$

first-order

$$P(x_i \mid x_{i-1}, \ldots, x_1) \approx P(x_i \mid x_{i-2}, x_{i-1})$$

second-order

N-gram language models

$$P(x) = \prod_i P(x_i) \times P(\text{STOP})$$

unigram

$$P(x) = \prod_i P(x_i \mid x_{i-1}) \times P(\text{STOP} \mid x_n)$$

bigram

$$P(x) = \prod_i P(x_i \mid x_{i-2}, x_{i-1}) \times P(\text{STOP} \mid x_{n-1}, x_n)$$

trigram

Bigram Generation

$$P(x) = \prod_i P(x_i \mid x_{i-1}) \times P(\text{STOP} \mid x_n)$$

bigram

Previous word	Current word	Probability
START	I	0.2
START	like	0.0008
…	…	…
I	like	0.1
I	coffee	0.0003
…	…	…
coffee	STOP	0.06
I	STOP	0.0001

START _____

Bigram Generation

$$P(x) = \prod_i P(x_i \mid x_{i-1}) \times P(\text{STOP} \mid x_n)$$

Previous word	Current word	Probability
START	I	0.2
START	like	0.0008
…	…	…
I	like	0.1
I	coffee	0.0003
…	…	…
coffee	STOP	0.06
I	STOP	0.0001

bigram

START I

START I _____

Bigram Generation

$$P(x) = \prod_i P(x_i \mid x_{i-1}) \times P(\text{STOP} \mid x_n)$$

Previous word	Current word	Probability
START	I	0.2
START	like	0.0008
…	…	…
I	like	0.1
I	coffee	0.0003
…	…	…
coffee	STOP	0.06
I	STOP	0.0001

bigram

START I

START I like

START I like _____

Bigram Generation

$$P(x) = \prod_i P(x_i \mid x_{i-1}) \times P(\text{STOP} \mid x_n)$$

Previous word	Current word	Probability
START	I	0.2
START	like	0.0008
…	…	…
I	like	0.1
I	coffee	0.0003
…	…	…
coffee	STOP	0.06
I	STOP	0.0001

bigram

START I

START I like

START I like black

START I like black _____

Bigram Generation

$$P(x) = \prod_i P(x_i \mid x_{i-1}) \times P(\text{STOP} \mid x_n)$$

Previous word	Current word	Probability
START	I	0.2
START	like	0.0008
…	…	…
I	like	0.1
I	coffee	0.0003
…	…	…
coffee	STOP	0.06
I	STOP	0.0001

bigram

START I

START I like

START I like black

START I like black coffee

START I like black coffee ____

Bigram Generation

$$P(x) = \prod_i P(x_i \mid x_{i-1}) \times P(\text{STOP} \mid x_n)$$

Previous word	Current word	Probability
START	I	0.2
START	like	0.0008
…	…	…
I	like	0.1
I	coffee	0.0003
…	…	…
coffee	STOP	0.06
I	STOP	0.0001

bigram

START I

START I like

START I like black

START I like black coffee

START I like black coffee STOP

ESTIMATION

Maximum likelihood estimation

$$P(x) = \prod_i P(x_i) \times P(\text{STOP})$$

unigram

$$P(x_i) = \frac{c(x_i)}{N}$$

$$P(x) = \prod_i P(x_i \mid x_{i-1}) \times P(\text{STOP} \mid x_n)$$

bigram

$$P(x_i \mid x_{i-1}) = \frac{c(x_i, x_{i-1})}{c(x_{i-1})}$$

$$P(x) = \prod_i P(x_i \mid x_{i-2}, x_{i-1}) \times P(\text{STOP} \mid x_{n-1}, x_n)$$

trigram

$$P(x_i \mid x_{i-2}, x_{i-1}) = \frac{c(x_i, x_{i-1}, x_{i-2})}{c(x_{i-1}, x_{i-2})}$$

Data sparsity

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

Figure 4.1 Bigram counts for eight of the words (out of $V = 1446$) in the Berkeley Restaurant Project corpus of 9332 sentences. Zero counts are in gray.

SLP 4.3

If probabilities are estimated from a fixed finite sample, we'll encounter situations where we won't see some terms, bigrams, etc?

Data sparsity

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

Figure 4.1 Bigram counts for eight of the words (out of $V = 1446$) in the Berkeley Restaurant Project corpus of 9332 sentences. Zero counts are in gray.

SLP 4.3

How to calculate P(x)? How to estimate P(x_i)?

Smoothing

Maximum likelihood estimate

$$P(x_i) = \frac{c(x_i)}{N}$$

Smoothed estimate

$$P(x_i) = \frac{c(x_i) + \alpha}{N + V\alpha}$$

Various different ways to do smoothing; other solutions include interpolation

Evaluation

- How do we know if our language model is good?
- Ideally, we should evaluate a language model extrinsically (e.g., how much a language model improves the performance of a machine translation model)
- More often we evaluate a language model intrinsically

Evaluation

- A good language model should assign high probability to unseen sequences that are plausible
- A good language model should be less perplexed or surprised
- Perplexity = inverse probability of test data, averaged over words

Perplexity

- Perplexity = $2^{-\frac{l(w)}{M}}$, where M is the total number of unseen tokens
- $$l(w) = \sum_{m=1}^{M} \log P(w_m \mid w_{m-1}, \ldots, w_1)$$
- Based on the language model of your choice, you'll calculate l(w)
- Smaller perplexity is better!

Skipgram

- In one of the most sensational comebacks in the champions league Liverpool defeated Barcelona by mounting a strong performance in the second half

Skipgram

- In one of the most sensational comebacks in the champions league Liverpool defeated Barcelona by mounting a strong _____ in the second half

Skipgram

- In one of the most sensational comebacks in the champions league Liverpool defeated Barcelona by mounting a strong _____ in the second half

Word2vec learned word vectors using the skipgram language model

In class

- lm exploration