CS 6501 Natural Language Processing

Statistical Language Modeling

Yangfeng Ji

Information and Language Processing Lab
Department of Computer Science
University of Virginia

Overview

Introduction

Word Prediction with Context

Consider the example, what words are likely to follow

```
Please turn your homework ...
```

[Jurafsky and Martin, 2019]

Word Prediction with Context

Consider the example, what words are likely to follow

`Please turn your homework ...`

Although we do not know the actual word in the original text, we have a good sense about what of these following words are likely to follow

- `in`
- `over`
- `refrigerator`
- `the`

[Jurafsky and Martin, 2019]

Mathematical Formulation

- Let $X_1, X_2, \ldots, X_{t-1}$ be the random variables representing the words in the context, and X_t be the next word that we would like the model to predict.

Mathematical Formulation

- Let $X_1, X_2, \ldots, X_{t-1}$ be the random variables representing the words in the context, and X_t be the next word that we would like the model to predict.
- Similarly, we can formulate this prediction as a classification problem, where $\boldsymbol{X}_{1:t-1} = X_1, X_2, \ldots, X_{t-1}$ are the input words and X_t is the output, we can write the classifier in a probabilistic form

$$P(X_t \mid X_1, \ldots, X_{t-1}) \quad \text{or} \quad P(X_t \mid \boldsymbol{X}_{1:t-1}) \tag{1}$$

Mathematical Formulation

- Let $X_1, X_2, \ldots, X_{t-1}$ be the random variables representing the words in the context, and X_t be the next word that we would like the model to predict.
- Similarly, we can formulate this prediction as a classification problem, where $\boldsymbol{X}_{1:t-1} = X_1, X_2, \ldots, X_{t-1}$ are the input words and X_t is the output, we can write the classifier in a probabilistic form

$$P(X_t \mid X_1, \ldots, X_{t-1}) \quad \text{or} \quad P(X_t \mid \boldsymbol{X}_{1:t-1}) \tag{1}$$

- The topics in this and the next lectures will offer two modeling methods in equation 1.

Mathematical Formulation

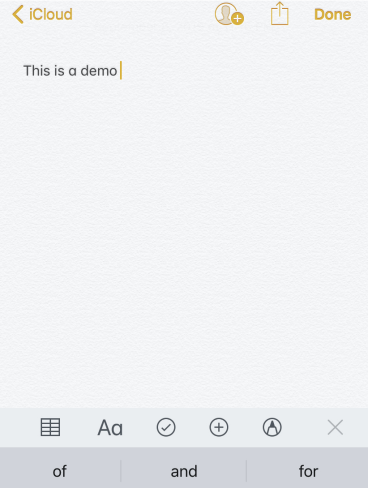
- Let $X_1, X_2, \ldots, X_{t-1}$ be the random variables representing the words in the context, and X_t be the next word that we would like the model to predict.
- Similarly, we can formulate this prediction as a classification problem, where $\boldsymbol{X}_{1:t-1} = X_1, X_2, \ldots, X_{t-1}$ are the input words and X_t is the output, we can write the classifier in a probabilistic form

$$P(X_t \mid X_1, \ldots, X_{t-1}) \quad \text{or} \quad P(X_t \mid \boldsymbol{X}_{1:t-1}) \tag{1}$$

- The topics in this and the next lectures will offer two modeling methods in equation 1.
- Difference with the word embedding methods discussed in the previous lecture
  - Skip-gram model: predicting the surrounding words
  - Language models: predicting the next word

Word Prediction in Input Methods

Input methods use language models to predict the next likely words, to speed up the typing



Writing a Poem?

Trevor Noah and Amanda Gorman writing poems with the input methods on their phones

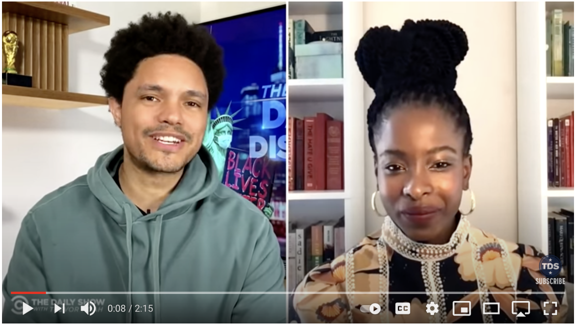


Figure: The Daily Social Distancing Show: Bonus Track feat. Amanda Gorman

Link

Joint Probability and Chain Rule

Given the conditional probability $P(X_t \mid \boldsymbol{X}_{1:t-1})$, to evaluate the qualty of a text, we need the chain rule in probability to factorize the joint probability $P(\boldsymbol{X}_{1:t})$ into a series of conditional probabilities.

$$P(X_1, X_2, \cdots, X_t) \quad = \quad P(X_1)P(X_2, \cdots, X_k \mid X_1) \tag{2}$$

Joint Probability and Chain Rule

Given the conditional probability $P(X_t \mid \boldsymbol{X}_{1:t-1})$, to evaluate the qualty of a text, we need the chain rule in probability to factorize the joint probability $P(\boldsymbol{X}_{1:t})$ into a series of conditional probabilities.

$$\begin{aligned} P(X_1, X_2, \cdots, X_t) &= P(X_1)P(X_2, \cdots, X_k \mid X_1) \\ &= P(X_1)P(X_2 \mid X_1)P(X_3, \cdots, X_t \mid X_1, X_2) \end{aligned} \tag{2}$$

Joint Probability and Chain Rule

Given the conditional probability $P(X_t \mid \boldsymbol{X}_{1:t-1})$, to evaluate the qualty of a text, we need the chain rule in probability to factorize the joint probability $P(\boldsymbol{X}_{1:t})$ into a series of conditional probabilities.

$$\begin{aligned} P(X_1, X_2, \cdots, X_t) &= P(X_1)P(X_2, \cdots, X_k \mid X_1) \\ &= P(X_1)P(X_2 \mid X_1)P(X_3, \cdots, X_t \mid X_1, X_2) \\ &= P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_1, X_2) \cdots \\ &\quad P(X_t \mid X_1, \cdots, X_{t-1}) \end{aligned} \tag{2}$$

Joint Probability and Chain Rule

Given the conditional probability $P(X_t \mid \boldsymbol{X}_{1:t-1})$, to evaluate the qualty of a text, we need the chain rule in probability to factorize the joint probability $P(\boldsymbol{X}_{1:t})$ into a series of conditional probabilities.

$$
\begin{aligned}
P(X_1, X_2, \cdots, X_t) &= P(X_1)P(X_2, \cdots, X_k \mid X_1) \\
&= P(X_1)P(X_2 \mid X_1)P(X_3, \cdots, X_t \mid X_1, X_2) \\
&= P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_1, X_2) \cdots \\
&\quad\; P(X_t \mid X_1, \cdots, X_{t-1}) \\
&= \prod_{i=1}^{k} P(X_i \mid X_1, \ldots, X_{i-1}) \qquad (2)
\end{aligned}
$$

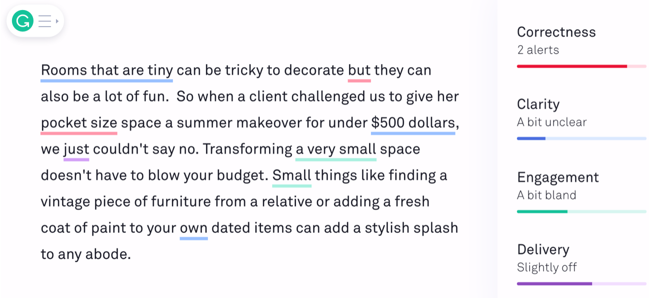
Given a voice signal, a language model in speech recognition will evaluate the likelihood of decoded texts

$$P(\text{I saw a van}) \gg P(\text{eyes awe of an}) \tag{3}$$

[Jurafsky and Martin, 2019]

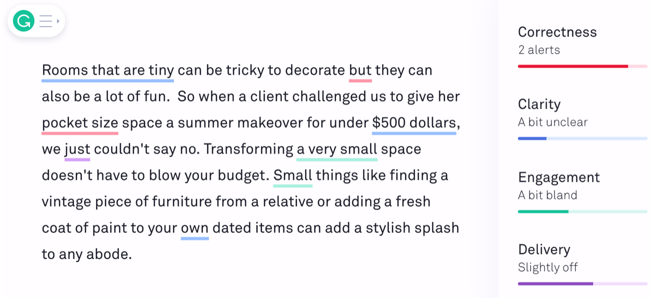
Writing Assistant

Grammarly:



Writing Assistant

Grammarly:



A good writing assistant system involves two tasks

- evaluate the quality of a text
- generate revision suggestions

A language model cannot provide support to all functions directly, but is a critical component in the backend system

Applications

- Generative tasks: predicting the next word given a context
  - Word prediction
  - Text generation
  - ...

Applications

- Generative tasks: predicting the next word given a context
  - Word prediction
  - Text generation
  - ...
- Discriminative tasks: evaluating the quality of texts
  - Speech recognition
  - Machine translation
  - Document summarization
  - ...

N-gram Language Models

Problem Definition

Given a vocab $\mathcal{V}$ that contains all the possible word types, then the prediction of X_t can be formulated as

$$P(X_t \mid X_1, \ldots, X_{t-1}) = ? \tag{4}$$

Problem Definition

Given a vocab $\mathscr{V}$ that contains all the possible word types, then the prediction of X_t can be formulated as

$$P(X_t \mid X_1, \ldots, X_{t-1}) = ? \tag{4}$$

The challenges of modeling $P(X_t \mid X_1, \ldots, X_{t-1})$

- it is a categorical distribution defined on the vocab $\mathscr{V}$
- it consider the entire context from the very first word X_1 to the previous word X_{t-1}

Problem Definition

Given a vocab $\mathcal{V}$ that contains all the possible word types, then the prediction of X_t can be formulated as

$$P(X_t \mid X_1, \ldots, X_{t-1}) = ? \tag{4}$$

The challenges of modeling $P(X_t \mid X_1, \ldots, X_{t-1})$

- it is a categorical distribution defined on the vocab $\mathcal{V}$
  - Unfortunately, we cannot do much on this problem, other than using hiarchical structures in softmax function (e.g., hierarchical softmax and class-factored softmax)
- it consider the entire context from the very first word X_1 to the previous word X_{t-1}

Problem Definition

Given a vocab $\mathcal{V}$ that contains all the possible word types, then the prediction of X_t can be formulated as

$$P(X_t \mid X_1, \ldots, X_{t-1}) = ? \tag{4}$$

The challenges of modeling $P(X_t \mid X_1, \ldots, X_{t-1})$

- it is a categorical distribution defined on the vocab $\mathcal{V}$
  - Unfortunately, we cannot do much on this problem, other than using hiarchical structures in softmax function (e.g., hierarchical softmax and class-factored softmax)
- it consider the entire context from the very first word X_1 to the previous word X_{t-1}
  - The main topic of this section

Parameter Estimation

With a collection of texts as training examples, the simple method of estimating the probabilities is using maximum likelihood estimation.

- In the first lecture, we discussed the MLE of a Bernoulli distribution

$$\hat{P}(X = 1) = \frac{\sum_{i=1}^{N} \delta(x_i, 1)}{N} = \frac{c(X = 1)}{N} \tag{5}$$

  where $c(X = 1)$ is the number of observations with value 1

[Collins, 2017]

Parameter Estimation

With a collection of texts as training examples, the simple method of estimating the probabilities is using maximum likelihood estimation.

- In the first lecture, we discussed the MLE of a Bernoulli distribution
$$\hat{P}(X = 1) = \frac{\sum_{i=1}^{N} \delta(x_i, 1)}{N} = \frac{c(X = 1)}{N} \quad (5)$$
where $c(X = 1)$ is the number of observations with value 1
- Similarly, to estimate the conditional probability $P(X_t \mid \boldsymbol{X}_{1:t-1})$, we have
$$\hat{P}(X_t = x_t \mid \boldsymbol{X}_{1:t-1} = \boldsymbol{x}_{1:t-1}) = \frac{c(\boldsymbol{x}_{1:t})}{c(\boldsymbol{x}_{1:t-1})} \quad (6)$$
where $c(\boldsymbol{x}_{1:t-1})$ is the number that text $\boldsymbol{x}_{1:t-1}$ appears in the training examples, and $c(\boldsymbol{x}_{1:t})$ is the number that text $\boldsymbol{x}_{1:t}$ appears in the training examples.

[Collins, 2017]

Parameter Estimation (Cont.)

Imagine we have a huge collection of texts for parameter estimation

- With the sentence "`the dog barks`"

$$\hat{P}(X_3 = \texttt{barks} \mid \boldsymbol{X}_{1:2} = \texttt{the dog}) = \frac{c(\boldsymbol{X}_{1:3} = \texttt{the dog barks})}{c(\boldsymbol{X}_{1:2} = \texttt{the dog})} \quad (7)$$

Parameter Estimation (Cont.)

Imagine we have a huge collection of texts for parameter estimation

- With the sentence "`the dog barks`"

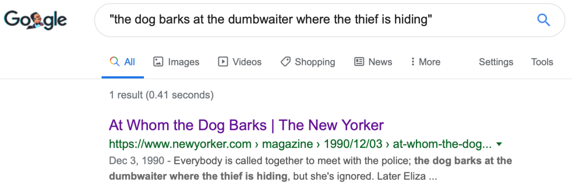
$$\hat{P}(X_3 = \texttt{barks} \mid \boldsymbol{X}_{1:2} = \texttt{the dog}) = \frac{c(\boldsymbol{X}_{1:3} = \texttt{the dog barks})}{c(\boldsymbol{X}_{1:2} = \texttt{the dog})} \quad (7)$$

- With the sentence "`the dog barks at the dumbwaiter where the thief is hiding`"

$$\begin{aligned}\hat{P}(X_{11} &= \texttt{hiding} \mid \boldsymbol{X}_{1:10} = \texttt{the dog} \cdots \texttt{is}) \\ &= \frac{c(X_{1:11} = \texttt{the dog} \cdots \texttt{is hiding})}{c(X_{1:10} = \texttt{the dog} \cdots \texttt{is})}\end{aligned} \quad (8)$$

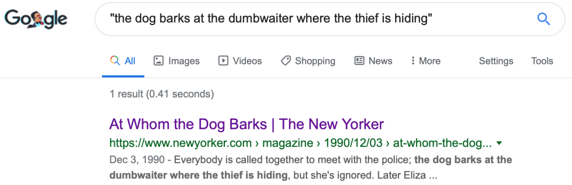
Parameter Estimation (Cont.)

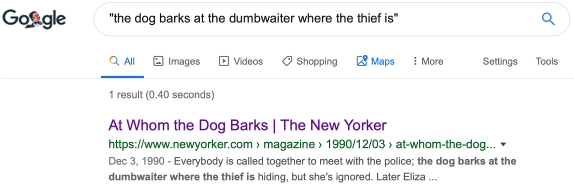
For this specific sentence, we only one training example even if we collect all the texts from the Internet



Parameter Estimation (Cont.)

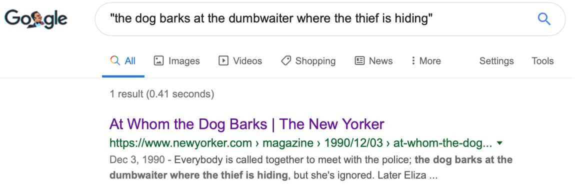
For this specific sentence, we only one training example even if we collect all the texts from the Internet

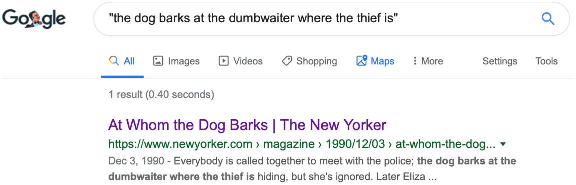




Parameter Estimation (Cont.)

For this specific sentence, we only one training example even if we collect all the texts from the Internet





$$\hat{P}(X_{11} = \texttt{hiding} \mid \boldsymbol{X}_{1:10} = \texttt{the dog} \cdots \texttt{is}) = 1.0$$

Simplification: Uni-gram

- The main challenge of parameter estimation is the long-term dependence between X_t and $\boldsymbol{X}_{1:t-1}$

Simplification: Uni-gram

- The main challenge of parameter estimation is the long-term dependence between X_t and $\boldsymbol{X}_{1:t-1}$
- Uni-gram: assume all words are independent with each other. With this assumption, we only need to estimate the probability of each individual word (no conditional probability involved)

$$P(X_t \mid X_1, \ldots, X_{t-1}) \approx P(X_t) \tag{9}$$

Simplification: Uni-gram

- The main challenge of parameter estimation is the long-term dependence between X_t and $\boldsymbol{X}_{1:t-1}$
- Uni-gram: assume all words are independent with each other. With this assumption, we only need to estimate the probability of each individual word (no conditional probability involved)

$$P(X_t \mid X_1, \ldots, X_{t-1}) \approx P(X_t) \tag{9}$$

- For example

$$P(\text{barks} \mid \text{the dog}) \approx P(\text{barks}) \tag{10}$$

Simplification: Uni-gram

- The main challenge of parameter estimation is the long-term dependence between X_t and $\boldsymbol{X}_{1:t-1}$
- Uni-gram: assume all words are independent with each other. With this assumption, we only need to estimate the probability of each individual word (no conditional probability involved)

$$P(X_t \mid X_1, \ldots, X_{t-1}) \approx P(X_t) \tag{9}$$

- For example

$$P(\text{barks} \mid \text{the dog}) \approx P(\text{barks}) \tag{10}$$

- Comments: the tradeoff between prediction power and number of parameters
  - It has extremely limited prediction power
  - Number of parameters: $V = |\mathcal{V}|$

Bi-gram Models

- To find a good balance between the prediction power and parameter estimation challenge, we can limit the contextual information used in a language modeling.
- Bi-gram model: uses only one word X_{t-1} from the previous context to predict the current word X_t

$$P(X_t \mid \boldsymbol{X}_{1:t-1}) \approx P(X_t \mid X_{t-1}) \tag{11}$$

Bi-gram Models

- To find a good balance between the prediction power and parameter estimation challenge, we can limit the contextual information used in a language modeling.
- Bi-gram model: uses only one word X_{t-1} from the previous context to predict the current word X_t

$$P(X_t \mid \boldsymbol{X}_{1:t-1}) \approx P(X_t \mid X_{t-1}) \tag{11}$$

- For example, given the text "`the dog barks`", the prediction of the last word `barks` in a bi-gram model is formulated as

$$P(\texttt{barks} \mid \texttt{the dog}) \approx P(\texttt{barks} \mid \texttt{dog}) \tag{12}$$

Bi-gram Models

- To find a good balance between the prediction power and parameter estimation challenge, we can limit the contextual information used in a language modeling.
- Bi-gram model: uses only one word X_{t-1} from the previous context to predict the current word X_t

$$P(X_t \mid \boldsymbol{X}_{1:t-1}) \approx P(X_t \mid X_{t-1}) \tag{11}$$

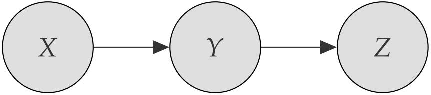
- For example, given the text "`the dog barks`", the prediction of the last word `barks` in a bi-gram model is formulated as

$$P(\texttt{barks} \mid \texttt{the dog}) \approx P(\texttt{barks} \mid \texttt{dog}) \tag{12}$$

- In probabilistic modeling, a bi-gram model is an application of the first-order Markov model

Markov Property

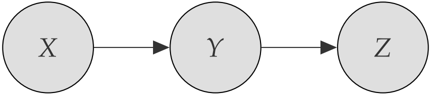
First-order Markov property: given



$$P(Z \mid X, Y) = P(Z \mid Y) \tag{13}$$

Markov Property

First-order Markov property: given



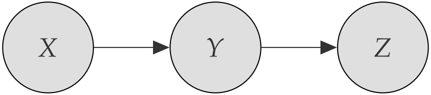
$$P(Z \mid X, Y) = P(Z \mid Y) \tag{13}$$

It simplifies the conditional probability

$$P(X_t \mid X_1, \ldots, X_{t-1}) \approx P(X_t \mid X_{t-1}) \tag{14}$$

Markov Property

First-order Markov property: given



$$P(Z \mid X, Y) = P(Z \mid Y) \tag{13}$$

It simplifies the conditional probability

$$P(X_t \mid X_1, \ldots, X_{t-1}) \approx P(X_t \mid X_{t-1}) \tag{14}$$

and also the joint probability

$$\begin{aligned} P(X_1, \ldots, X_t) \quad &\approx \quad P(X_t \mid X_{t-1}) \cdot P(X_{t-1} \mid X_{t-2}) \cdots \\ & \quad\; P(X_2 \mid X_1) \cdot P(X_1) \end{aligned} \tag{15}$$

Special Tokens

Consider the application of using a bi-gram model

$$
\begin{aligned}
P(\text{the dog barks}) \quad = \quad & P(\text{the}) \cdot P(\text{dog} \mid \text{the}) \\
& P(\text{barks} \mid \text{dog})
\end{aligned}
$$

Special Tokens

Consider the application of using a bi-gram model

$$
\begin{aligned}
P(\text{the dog barks}) \quad &= \quad P(\text{the}) \cdot P(\text{dog} \mid \text{the}) \\
&\qquad P(\text{barks} \mid \text{dog})
\end{aligned}
$$

The model needs

- a special token ($\square$) to distinguish $P(\text{the})$ from the marginal distribution of word "the"

Special Tokens

Consider the application of using a bi-gram model

$$P(\text{the dog barks}) = P(\text{the}) \cdot P(\text{dog} \mid \text{the}) \\ P(\text{barks} \mid \text{dog})$$

The model needs

- a special token ($\square$) to distinguish $P(\text{the})$ from the marginal distribution of word "`the`"
- another special token ($\blacksquare$) to indicate the end of a sentence

Special Tokens

Consider the application of using a bi-gram model

$$\begin{aligned} P(\text{the dog barks}) &= P(\text{the}) \cdot P(\text{dog} \mid \text{the}) \\ & \quad P(\text{barks} \mid \text{dog}) \end{aligned}$$

The model needs

- a special token ($\square$) to distinguish $P(\text{the})$ from the marginal distribution of word "`the`"
- another special token ($\blacksquare$) to indicate the end of a sentence

Factorization with special tokens:

$$\begin{aligned} P(\square \text{ the dog barks } \blacksquare) &= P(\text{the} \mid \square) \cdot P(\text{dog} \mid \text{the}) \\ & \quad P(\text{barks} \mid \text{dog}) \cdot P(\blacksquare \mid \text{barks}) \end{aligned}$$

Example: Parameter Estimation

Example sentences

- □ I am Sam ■
- □ Sam I am ■
- □ I do not like green eggs and ham ■

[Jurafsky and Martin, 2019]

Example: Parameter Estimation

Example sentences

- `□ I am Sam ■`
- `□ Sam I am ■`
- `□ I do not like green eggs and ham ■`

Some of the probabilities:

$$\hat{P}(\mathrm{I} \mid \square) = \frac{2}{3} \quad \hat{P}(\blacksquare \mid \mathrm{Sam}) = \frac{1}{2} \quad \hat{P}(\mathrm{do} \mid \mathrm{I}) = \frac{1}{3} \tag{16}$$

[Jurafsky and Martin, 2019]

Issues with a Fixed Vocabulary

- $P(X_t \mid X_{t-1})$ is defined a fixed vocabulary, for normalization purpose

$$P(X_t \mid X_{t-1}) = \frac{c(X_{t-1}, X_t)}{\sum_{X' \in \mathcal{V}} c(X_{t-1}, X')} \tag{17}$$

Issues with a Fixed Vocabulary

- $P(X_t \mid X_{t-1})$ is defined a fixed vocabulary, for normalization purpose

$$P(X_t \mid X_{t-1}) = \frac{c(X_{t-1}, X_t)}{\sum_{X' \in \mathcal{V}} c(X_{t-1}, X')} \tag{17}$$

- Issues with a fixed vocabulary
  - Unknown words: word x is not in the vocabulary
  - Zero probability: word combination (x, x') never appears in the training set

Unknown Words

Replace all words that are not in the vocab with a special token UNK.

For example

- Original text: "`the dog barks at the dumbwaiter where the thief is hiding`"
- After preprocessing: "`the dog barks at the` UNK `where the thief is hiding`"

Unknown Words

Replace all words that are not in the vocab with a special token UNK.

For example

- Original text: "`the dog barks at the dumbwaiter where the thief is hiding`"
- After preprocessing: "`the dog barks at the` UNK `where the thief is hiding`"

Quiz

Can we simply ignore the unknown words? For example, what if the preprocessed text is

"`the dog barks at the  where the thief is hiding`"

Tri-gram Models

We can extend the conditional probability to depend on previous two tokens

$$P(X_t \mid X_1, \ldots, X_{t-1}) \approx P(X_t \mid X_{t-2}, X_{t-1}) \tag{18}$$

Comments

- More dependency leads to more accurate predictions
- Parameter estimation

$$\hat{P}(X_t \mid X_{t-2}, X_{t-1}) = \frac{c(\boldsymbol{X}_{t-2:t})}{c(\boldsymbol{X}_{t-2:t-1})} \tag{19}$$

Number of Parameters

- Uni-gram model
  - Ignore context words completely $P(X_t \mid \boldsymbol{X}_{1:t-1}) \approx P(X_t)$
  - Number of parameters $\mathcal{O}(|\mathcal{V}|)$

	X_1	$\cdots$	$X_{|\mathcal{V}|}$
$P(X_t)$			

Number of Parameters

- Uni-gram model
  - Ignore context words completely $P(X_t \mid \boldsymbol{X}_{1:t-1}) \approx P(X_t)$
  - Number of parameters $\mathcal{O}(|\mathcal{V}|)$

	X_1	$\cdots$	$X_{\lvert\mathcal{V}\rvert}$
$P(X_t)$			

- Bi-gram model
  - Use only the adjacent word $P(X_t \mid \boldsymbol{X}_{1:t-1}) \approx P(X_t \mid X_{t-1})$
  - Number of parameters $\mathcal{O}(|\mathcal{V}|^2)$

$P(X_t \mid X_{t-1})$	X_1	$\cdots$	$X_{\lvert\mathcal{V}\rvert}$
X_1			
$\vdots$			
$X_{\lvert\mathcal{V}\rvert}$			

Number of Parameters

- Uni-gram model
  - Ignore context words completely $P(X_t \mid \boldsymbol{X}_{1:t-1}) \approx P(X_t)$
  - Number of parameters $\mathcal{O}(|\mathcal{V}|)$

	X_1	$\cdots$	$X_{|\mathcal{V}|}$
$P(X_t)$			

- Bi-gram model
  - Use only the adjacent word $P(X_t \mid \boldsymbol{X}_{1:t-1}) \approx P(X_t \mid X_{t-1})$
  - Number of parameters $\mathcal{O}(|\mathcal{V}|^2)$

$P(X_t \mid X_{t-1})$	X_1	$\cdots$	$X_{|\mathcal{V}|}$
X_1			
$\vdots$			
$X_{|\mathcal{V}|}$			

- Tri-gram model
  - Use two preceding words $P(X_t \mid \boldsymbol{X}_{1:t-1}) \approx P(X_t \mid X_{t-2}, X_{t-1})$
  - Number of parameters $\mathcal{O}(|\mathcal{V}|^3)$

Generation with Bi-gram Models

Demo Code

- A bi-gram model with no smoothing
- Training with the dataset from the arXiv paper abstracts
- Generating by *randomly* sampling from this bi-gram model

Demo Code

- A bi-gram model with no smoothing
- Training with the dataset from the arXiv paper abstracts
- Generating by *randomly* sampling from this bi-gram model

Checkout the demo code for some examples

Smoothing Techniques

High-order Markov Models

A motivating example:

`The printer on the 5th floor of Rice hall crashed`

High-order Markov Models

A motivating example:

`The printer on the 5th floor of Rice hall crashed`

n-gram Language Models

- Uni-gram: $P(X_t)$
- Bi-gram: $P(X_t \mid X_{t-1})$
- Tri-gram: $P(X_t \mid X_{t-2}, X_{t-1})$
- 4-gram: $P(X_t \mid X_{t-3}, X_{t-2}, X_{t-1})$
- 5-gram: $P(X_t \mid X_{t-4}, X_{t-3}, X_{t-2}, X_{t-1})$

Discounting

It is the same method used in parameter estimation of naive Bayes classifiers

$$P(X_t \mid X_{t-1}) = \frac{c(X_{t-1}, X_t) + \alpha}{c(X_{t-1}) + \alpha V} \tag{20}$$

where $\alpha > 0$ is a hyper-parameter.

Linear Interpolation

Estimate the following three models with MLE:

- Uni-gram: $P(X_t)$
- Bi-gram: $P(X_t \mid X_{t-1})$
- Tri-gram: $P(X_t \mid X_{t-2}, X_{t-1})$

Linear Interpolation

Estimate the following three models with MLE:

- Uni-gram: $P(X_t)$
- Bi-gram: $P(X_t \mid X_{t-1})$
- Tri-gram: $P(X_t \mid X_{t-2}, X_{t-1})$

Then, the new probability of X_t given X_{t-2} and X_{t-1} is

$$\begin{aligned} P_{LI}(X_t \mid X_{t-2}, X_{t-1}) &= \lambda_1 \cdot P(X_t) + \lambda_2 \cdot P(X_t \mid X_{t-1}) \\ &\quad + \lambda_3 \cdot P(X_t \mid X_{t-2}, X_{t-1}) \end{aligned} \tag{21}$$

$\{\lambda_i\}$ are learned with a held-out corpus (a development set).

Language Model Evaluation

Sentence Evaluation (I)

Evaluation with joint probabilities

$$P(\text{I love black coffee}) \ \text{vs.} \ P(\text{black coffee pleases me}) \tag{22}$$

Direct comparison between the probabilities will tell us which sentence is more *fluent*.

Sentence Evaluation (II)

Limitation of comparing joint probabilities directly

$$P(\text{I love black coffee}) \text{ vs. } P(\text{I like black coffee very much}) \tag{23}$$

Due to the *length difference*, the second probability may always be smaller than the first.

Likelihood

- Test data: including the special tokens

$$\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M$$

- Likelihood

$$\begin{aligned}
\text{Log-lik}(\{\boldsymbol{x}_{m=1}^{M}\}) &= \log_2 \prod_{m=1}^{M} \prod_{t=1} P(x_{m,t} \mid \boldsymbol{x}_{m,1:t-1}) && (24) \\
&= \sum_{m=1}^{M} \sum_{t=1} \log_2 P(x_{m,t} \mid \boldsymbol{x}_{m,1:t-1}) && (25)
\end{aligned}$$

- Factors
  - Number of the tokens
  - No intuitive explanation

Perplexity

The definition of perplexity is

$$\text{Perplexity} = 2^{-\frac{1}{T}\text{Log-lik}(\{\boldsymbol{x}_{m=1}^{M}\})} \tag{26}$$

where T is the total number of the log probabilities in Log-lik($\{\boldsymbol{x}_{m=1}^{M}\}$).

Special Case

- An impossible case

$$P(x_t \mid \boldsymbol{x}_{1:t-1}) = 1 \tag{27}$$

Special Case

- An impossible case

$$P(x_t \mid \boldsymbol{x}_{1:t-1}) = 1 \tag{27}$$

- Perplexity

$$\begin{aligned} \text{Perplexity} &= 2^{-\frac{1}{T}\sum_{k=1}^{M}\sum_{m=1}\log_2 1} \\ &= 2^0 \\ &= 1 \end{aligned} \tag{28}$$

Special Case (II)

- A trivial case

$$P(x_t \mid \boldsymbol{x}_{1:t-1}) = \frac{1}{|\mathcal{V}|} \tag{29}$$

Special Case (II)

- A trivial case

$$P(x_t \mid \boldsymbol{x}_{1:t-1}) = \frac{1}{|\mathcal{V}|} \tag{29}$$

- Perplexity

$$\begin{aligned}
\text{Perplexity} &= 2^{-\frac{1}{T}\sum_{k=1}^{M}\sum_{m=1}\log_2 \frac{1}{|\mathcal{V}|}} \\
&= 2^{-\frac{1}{T}(T \cdot \log_2 \frac{1}{|\mathcal{V}|})} \\
&= 2^{-\log_2 \frac{1}{|\mathcal{V}|}} \\
&= |\mathcal{V}|
\end{aligned} \tag{30}$$

Typical Values of Perplexity

- $|\mathcal{V}| = 50K$
- A uni-gram model: Perplexity = 955
- A bi-gram model: Perplexity = 137
- A tri-gram model: Perplexity = 74

Lower is better

[Collins, 2017]

A Few Comments on Perplexity

Perplexity

- is an intrinsic evaluation measurement

A Few Comments on Perplexity

Perplexity

- is an intrinsic evaluation measurement
- is not necessarily correlated with the performance of
  - e.g., lower perplexity does not mean better translation (wrt BLEU score)
- is not directly comparable even on the same test data
  - you need the exactly same input for comparison

Reference

Collins, M. (2017).
Natural language processing: Lecture notes.

Jurafsky, D. and Martin, J. (2019).
Speech and language processing.