Natural Language Processing

N-Gram Models

Joakim Nivre

Uppsala University
Department of Linguistics and Philology
joakim.nivre@lingfil.uu.se

Introduction

"Once upon a time, there was a ..."

- How can we guess the next word?
- Estimate $P(w|\text{"Once upon a time, there was a"})$ for any w
- This is a probabilistic language model

Language Models

- Language models are crucial in many NLP applications
- Example from speech recognition:

  she drank two beers
  she drank too beers
  she drank too deers

- Other NLP applications that make use of language models:
  1. Statistical machine translation
  2. Part-of-speech tagging
  3. Spell checking
  4. Optical character recognition

Probability Theory

- Let $w_1, \ldots, w_n$ be an arbitrary sequence of words
- We can compute $P(w_1, \ldots, w_n)$ using the chain rule:

$$
\begin{aligned}
P(w_1, \ldots, w_n) &= P(w_1)P(w_2|w_1)P(w_3|w_1, w_2) \cdots P(w_n|w_1, \ldots, w_{n-1}) \\
&= \prod_{i=1}^{n} P(w_i|w_1, \ldots w_{i-1})
\end{aligned}
$$

- But how do we find $P(w_i|w_1, \ldots, w_{i-1})$?

Estimation

- In theory, we can estimate $P(w_i|w_1, \ldots, w_{i-1})$ from data:

$$\hat{P}(w_i|w_1, \ldots, w_{i-1}) = \frac{f(w_1, \ldots, w_i)}{f(w_1, \ldots, w_{i-1})}$$

- In practice, this becomes infeasible as i grows larger
- With a vocabulary of 100,000 words, there are:

  10^5 possible unigrams ($i = 1$)
  10^{10} possible bigrams ($i = 2$)
  10^{15} trigrams ($i = 3$)
  10^{20} 4-grams ($i = 4$)
  ...

N-Gram Models

- We have to make independence assumptions
- In an *n*-gram model, we assume:

$$P(w_i|w_1, \ldots w_{i-1}) = P(w_i|w_{i-n+1}, \ldots, w_{i-1})$$

- Words are dependent only on $n-1$ preceding words
  1. Unigram ($n=1$): $P(w_i|w_1, \ldots, w_{i-1}) = P(w_i)$
  2. Bigram ($n=2$): $P(w_i|w_1, \ldots, w_{i-1}) = P(w_i|w_{i-1})$
  3. Trigram ($n=3$): $P(w_i|w_1, \ldots, w_{i-1}) = P(w_i|w_{i-2}, w_{i-1})$

Text Generation

1-gram model

1. your something
2. you she to offices the possible his of of his said sight , was laughing had .
3. white was not full meet old be to made , you no I . described that power he the , man , And ,
4. was Captain That she point labyrinth now must be far from . door had the from again what almost result fill , for coming as . a with made
5. his then a country-town by you ’ ago Men ?

Text Generation

2-gram model

1. Then here is the mud-bank what you , and instantly , and two officers waiting at once more valuable as I asked .
2. I may place is her husband and illegal constraint and outstanding , not recognised shape of finding that your heart , for communication between this man .
3. Mrs. Toller knows I mean that I have done very heartily at the 11 : That is a lad , his neighbour .
4. Then there has offered to its centre one left this case , upon me to violin-land , though the corner and hurried across the very large staples .
5. Holmes ran up by old-fashioned shutters of treachery to attend to which I thought I have a foreigner , too late Ezekiah Hopkins , with this rather cumbrous .

Text Generation

3-gram model

1. However , when last seen , but now I will leave no survivor from a solution by the Underground and hurried me into a bedroom , which boomed out every quarter of a brickish red .
2. ' I beg that you have ever done yet , among the trees and wayside hedges were just being lighted as we stepped from her imprudence in allowing this brute to trace some geese which were new to me .
3. Holmes had sat up in my uncle's life , and that a woman .
4. James and his hand and at the open , and has seen , but there are a thousand details which seem to have been hanged on far slighter evidence , I thought of !
5. Mr. Windibank draws my interest every quarter and pays it over to him .

Text Generation

4-gram model

1. Seeing that his passion was becoming ungovernable , I left him and returned towards Hatherley Farm .
2. You will excuse me , said my wife , and in order to see whether the objections are fatal , or if he had been to the side from which I could see that two of them were of the war he fought in Jackson's army , and afterwards from your gesture , that Miss Rucastle was perfectly happy , and that I can .
3. I rang the bell and called for the weekly county paper , which contained a verbatim account of the matter , but you do not see the point .
4. It hadn't pulled up before she shot out of the window ?
5. Why does fate play such tricks with poor , helpless worms ?

Text Generation

5-gram model

1. But what is it you wish ?
2. He was too good and kind to leave me so .
3. You may remember the old Persian saying , ' There is danger for him who taketh the tiger cub , and danger also for whoso snatches a delusion from a woman . '
4. The paper was made in Bohemia , I said .
5. You will observe , said Holmes , are you sure about this whistle and metallic sound ?

Evaluation

- Higher *n*-grams capture more linguistic structure
- But higher *n*-grams also require more training data
- What is the optimal trade-off?
- And how can we evaluate models more exactly?

Quiz

- How do we compute $P(\text{I love you})$ in a bigram model?
  1. $P(\text{I})P(\text{love})P(\text{you})$
  2. $P(\text{I})P(\text{love}|\text{I})P(\text{you}|\text{love})$
  3. $P(\text{I})P(\text{love}|\text{I})P(\text{you}|\text{I love})$

Natural Language Processing

Evaluating Language Models

Joakim Nivre

Uppsala University
Department of Linguistics and Philology
joakim.nivre@lingfil.uu.se

Entropy

- Remember:

$$H[X] = -\sum_{x \in \Omega_X} P(X = x) \log_2 P(X = x)$$

- Entropy can be seen as the expected amount of information (in bits), or as the difficulty of predicting the variable

Cross-Entropy

- The cross-entropy of distributions P and $\hat{P}$:

$$H[P, \hat{P}] = -\sum_{x \in \Omega_X} P(X = x) \log_2 \hat{P}(X = x)$$

- Cross-entropy can be seen as a measure of how closely $\hat{P}$ approximates the (true) distribution P
  - $H[X]$ is a lower bound for $H[P, \hat{P}]$
  - $H[X] = H[P, \hat{P}]$ iff $P(X = x) = \hat{P}(X = x)$ for all $x \in \Omega_X$

Estimating Cross-Entropy

- In theory, we could use cross-entropy to evaluate language models, preferring the model with lowest cross-entropy
- In practice, we don't know the true distribution, but we can estimate it using a test sample:

$$\hat{H}[P, \hat{P}] = -\frac{1}{N}\sum_{i=1}^{N} \log_2 \hat{P}(X = x_i)$$

  - Here $P(X = x_i)$ is estimated by $\frac{1}{N}f(X = x_i)$
  - But $\hat{P}(X = x_i)$ is also estimated by $\frac{1}{N}f(X = x_i)$
  - Therefore, we must test the model on a new data set

Perplexity

- Language models are often evaluated in terms of perplexity
- Perplexity is directly related to entropy:

$$PP[X] = 2^{H[X]}$$

- Both perplexity and entropy are inversely related to probability
  - We prefer the model with lowest entropy/perplexity
  - We prefer the model with highest probability

Natural Language Processing

Smoothing

Joakim Nivre

Uppsala University
Department of Linguistics and Philology
joakim.nivre@lingfil.uu.se

Introduction

- We can use an *n*-gram model to predict word probabilities
- We can use cross-entropy to evaluate the quality of the model
- But how do we estimate *n*-gram probabilities from data?

Maximum Likelihood Estimation

- Assume a bigram model:

$$\hat{P}(w_1, w_2) = \frac{f(w_1, w_2)}{N}$$

$$\hat{P}(w_1) = \frac{f(w_1)}{N}$$

$$\hat{P}(w_2|w_1) = \frac{\hat{P}(w_1, w_2)}{\hat{P}(w_1)} = \frac{f(w_1, w_2)}{f(w_1)}$$

- What can go wrong?

Unseen Events

- MLE takes the probability of all unseen events to be zero (0)
  - We use multiplication to combine *n*-gram probabilities
  - For any value of x, $0x = 0$
  - Thus, a single zero probability destroys all information
  - How do we know if an unseen event is impossible or just rare?
- In *n*-gram modeling there are two types of unseen events:
  1. Unseen words
  2. Unseen *n*-grams (involving known words)
- These are typically handled using different techniques

Unseen Words

- Create an unknown word token <UNK>
- At training time:
  - Create a fixed vocabulary V
  - Replace any training word not in V by <UNK>
  - Count <UNK> like any other word
- At test time:
  - Use <UNK> probabilities for any word not in training

Unseen N-Grams

- Assume that no *n*-gram of known words has 0 probability
- Redistribute probability mass from seen to unseen events
- This is known as smoothing or regularization
  - Finding a good smoothing method may be crucial
  - Not just about avoiding zero probabilities
  - Also improve estimates for low-frequency events

Additive Smoothing

- Just add one to all the counts

$$\text{MLE:} \quad \hat{P}(w_1, w_2) = \frac{f(w_1, w_2)}{N}$$

$$\text{Add } k\text{:} \quad \hat{P}(w_1, w_2) = \frac{f(w_1, w_2) + k}{N + k|V^2|}$$

  - Note the need to increase the denominator $(+k|V^2|)$
  - Automatic with marginalization over modified counts
- The special case of adding 1 is known as Laplace smoothing

More Advanced Methods

- Backoff – back off to a simpler model for rare events

$$\hat{P}(w_1, w_2) = \begin{cases} (1-\delta)\frac{f(w_1,w_2)}{N} & \text{if } f(w_1, w_2) > t \\ \alpha(w_1)\frac{f(w_2)}{N} & \text{otherwise} \end{cases}$$

- Interpolation – combine simple and complex models

$$\hat{P}(w_1, w_2) = \lambda\frac{f(w_1, w_2)}{N} + (1-\lambda)\frac{f(w_2)}{N}$$

- How much probability mass to reserve for unseen events?

Quiz 1

- Assume our vocabulary is $V = \{\text{one}, \text{for}, \text{all}\}$ and our training sample contains the following bigrams:

$$\{(\text{one}, \text{for}), (\text{for}, \text{all}), (\text{all}, \text{for}), (\text{for}, \text{one})\}$$

- Which of the following statements are correct?
  1. The MLE of $P(\text{one}, \text{for}) = 1$
  2. The MLE of $P(\text{one}, \text{for}) = \frac{1}{4}$
  3. The MLE of $P(\text{one}, \text{all}) = 0$
  4. The MLE of $P(\text{one}, \text{all}) = \frac{1}{4}$

Quiz 2

- Assume our vocabulary is $V = \{\text{one}, \text{for}, \text{all}\}$ and our training sample contains the following bigrams:

$$\{(\text{one}, \text{for}), (\text{for}, \text{all}), (\text{all}, \text{for}), (\text{for}, \text{one})\}$$

- Which of the following statements are correct?
  1. The Add-1 estimate of $P(\text{one}, \text{for}) = \frac{2}{4}$
  2. The Add-1 estimate $P(\text{one}, \text{for}) = \frac{2}{13}$
  3. The Add-1 estimate $P(\text{one}, \text{all}) = 0$
  4. The Add-1 estimate $P(\text{one}, \text{all}) = \frac{1}{13}$