CSCI 381/ CSCI 780: Natural Language Processing

Lecture 2: Language modeling

Today's lecture:
Language modeling

- Word prediction
- Probability review
- Joint probability distributions
- Markov processes/Markov assumptions
- Chain rule
- Parameter estimation for LMs
- Sparse data problems

Slides are based on material in J&M, Michael Collins' NLP course at Columbia and Julia Hockenmaier's NLP course at UIUC.

Word prediction

- I'd like to make a collect…
  - call

Word prediction and language modeling

- Computing the probability of a sequence of words
  - I would like to make a collect call
  - Make I like call collect would to a

- Which sequence has a higher probability of being encountered in an English text?

Language modeling

- Given a **training corpus** of texts, learn a <u>probability distribution ***p***</u> over sentences in the corpus
- We would like higher probability assigned to sentences that are likely
- Low probability assigned to sentences that are not likely

Language modeling

- Distribution ***p*** should have the following properties:

$$\sum_{\mathbf{s} \in S} p(s)=1$$

$$p(s) \geq 0 \text{ for all } s \in S$$

S – set of all training sentences, $s \in S$.

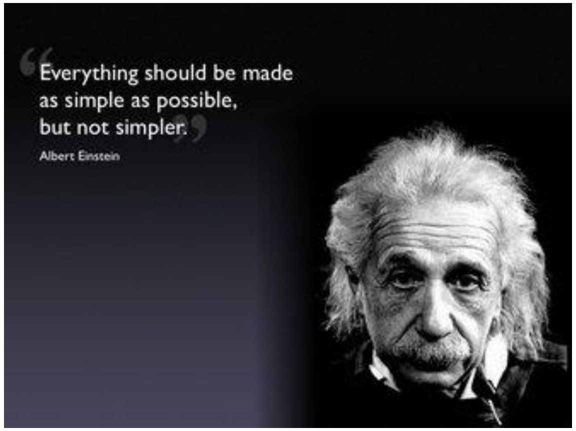
Language modeling: why do we need it?

- Speech recognition
- Spelling correction
  - He is trying to fine out.
  - They are leaving in about fifteen minuets.
- Character recognition
  - "I have a gub" (Take the Money and Run)
- Machine translation, etc.

Naïve language model

- Given a training corpus of *S* sentences, compute the number of occurrences of each sentence in the corpus. Let c(*s*) denote the number of times sentence *s* occurred in training. Then:
    - $p(s)=c(s)/S$

- Problem?



Joint probability

$p(s)=p(\text{I, would, like, to, make, a, collect, call})$

Chain rule

- **Chain rule – joint probability can be expressed via conditional probabilities as follows:**

$$p(A,B)=p(A)\cdot p(B|A)$$

$$p(A,B,C)=p(A)\cdot p(B|A)\cdot p(C|A,B)$$

$$\begin{aligned}&p(\mathrm{I, would, like, to, make, a, collect, call})\\&= p(\mathrm{I})\cdot p(\mathrm{would}|\mathrm{I})\cdot p(\mathrm{like}|\mathrm{I,would})\cdot p(\mathrm{to}|\mathrm{I,would,like})\cdot p(\mathrm{make}|\mathrm{I,would,like,to})\cdot\\&\quad p(\mathrm{a}|\mathrm{I,would,like,to,make})\cdot p(\mathrm{collect}|\mathrm{I,would,like,to,make,a})\cdot\\&\quad p(\mathrm{call}|\mathrm{I,would,like,to,make,a,collect})\end{aligned}$$

Markov processes

- **First-order Markov process** (independence assumption)

p(I, would, like, to, make, a, collect, call)

$= p(\text{I}) \cdot p(\text{would}|\text{I}) \cdot p(\text{like}|\text{would}) \cdot p(\text{to}|\text{like}) \cdot p(\text{make}|\text{to}) \cdot p(\text{a}|\text{make}) \cdot p(\text{collect}|\text{a}) \cdot p(\text{call}|\text{collect})$

Second-order Markov process

$p(\text{I, would, like, to, make, a, collect, call})$

$= p(\text{I}) \cdot p(\text{would}|\text{I}) \cdot p(\text{like}|\text{I,would}) \cdot p(\text{to}|\text{would,like}) \cdot p(\text{make}|\text{like,to}) \cdot p(\text{a}|\text{to,make}) \cdot p(\text{collect}|\text{make,a}) \cdot p(\text{call}|\text{a,collect})$

Markov assumptions and LMs

Unigram language model (every word is independent of the previous words)

Bigram language model – first-order Markov process

Trigram language model – second-order Markov process

N-grams LMs

- **N-gram** language models – predicting the next word using the previous N-1 words in the sentence.
  - I would like to make a collect…
    - Unigram model – N-1=0 P(call)
    - Bigram model: N-1=1 P(call|collect)
    - Trigram model: N-1=2 P(call|a, collect)
    - …

Training corpus and parameter estimation

❑Karlsson-on-the-Roof

On **a** perfectly **ordinary** street in Stockholm, in **a** perfectly **ordinary** house, lives **a** perfectly **ordinary** family called Ericson. There is **a** perfectly **ordinary** Daddy and **a** perfectly **ordinary** Mommy and three perfectly **ordinary** children—Bobby, Betty, and Eric….

There is only one person in the entire house who is not **ordinary**—and that is Karlsson-on-the-Roof. He lives on the roof, Karlsson does. This alone is out of the **ordinary**. Things may be different in other parts of the world, but in Stockholm people hardly ever live in **a** little house of their own on top of a roof. But Karlsson does. He is **a** very small, very round, and very self-possessed gentleman—and he can fly! Anybody can fly by airplane or helicopter, but only Karlsson can fly all by himself. He simply turns **a** button in the middle of his tummy and, presto, the cunning little engine on his back starts up. Karlsson waits for **a** moment or two to let the engine warm up; then he accelerates, takes off, and glides on his way with all the dignity and poise of **a** statesman; that is, if you can picture **a** statesman with **a** motor on his back.

Parameter estimation

- What is p(ordinary)?
- What is p(ordinary|perfectly)?
  - *Recall the chain rule: p*(A,B)=p(A)· p(B|A)

    = p(B)· p(A|B)=

  So, p(A|B)=p(A,B)/p(B)

  p(A,B)=c(A,B)/c(B)

  p(ordinary|perfectly)=c(perfectly,ordinary)/c(perfectly)
- What is p(ordinary|a,perfectly)?

Parameter estimation under 3-gram LM

$p(w_i | w_{i-2}, w_{i-1})$
$= c(w_{i-2}, w_{i-1}, w_i)/c(w_{i-2}, w_{i-1})$

$c(w_{i-2}, w_{i-1}, w_i) \rightarrow$ trigram count
$c(w_{i-2}, w_{i-1}) \rightarrow$ bigram count

p(ordinary|a,perfectly)
=c(a,perfectly,ordinary)/c(a,perfectly)

Model parameters

- How many parameters will a trigram model have?
  - Say, we have a vocabulary V of 20,000 unique words occurring in the training data
  - $|V|^3$

Example

- Find p("But Karlsson does .")
  - Under unigram model
  - Under bigram model
  - Under trigram model
- How many words (tokens) does the sentence have?

Maximum likelihood estimate

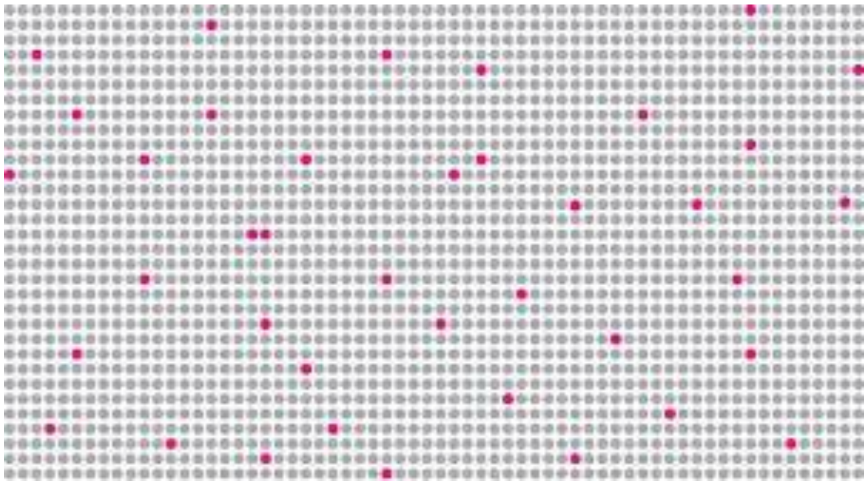
- The approach we used on the previous slide is called MLE:
  - It estimates the probability of an event using the observed frequencies in the training corpus
  - The probabilities correspond to **relative frequencies (MLE estimate)**
  - **No probability mass is assigned to events not seen in training! – Is this a problem?**

Bias and sensitivity to training data

- Any language model will be trained from some corpus , so some acceptable English n-grams will not occur

Sparse data problems

- Even with a large training corpus, most of the English n-grams will not occur in the training data and thus will have an MLE estimate of zero!



Smoothing

- Assigning non-zero probability to “unseen events”