Language models and n-grams

Today's lecture is based on Chapter 4 of the textbook
Wednesday's quiz will cover chapter 4+8.

Language models assign a probability to a sentence or a sequence of words.

Applications

1. Autocomplete for texting
2. Spelling correction
3. Speech recognition
4. Machine Translation
5. Other NLG
   - summarization
   - question answering
   - dialog system

p(it's hard to wreck a nice beach)

f_1 → high / large

p(high winds... > p(large winds

p(high gusts < p(large gusts

Goal: compute the probability of a sentence or sequence of words

$$P(W) = P(w_1, w_2, w_3, w_4, w_5 \ldots w_N)$$

Also want to calculate prob of an upcoming word

$$P(w_5 \mid w_1, w_2, w_3, w_4)$$

"Language model" LM

Compute the joint prob:

p(the underdog Philadelphia Eagles won)

Manipulate probabilities

Manipulate probabilities

conditional prob

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

given; joint prob; prior prob

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

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

Chain Rule

$$p(x_1, x_2, x_3 \cdots x_N) = p(x_1) \odot p(x_2|x_1) \odot p(x_3|x_1,x_2) \odot \cdots p(x_N|x_1, \cdots x_{N-1})$$

$$\Rightarrow p(w_1\ w_2\ w_3\ w_4 \cdots w_N) = \prod_i p(w_i | w_1\ w_2\ w_3 \cdots w_{i-1})$$

How do we estimate these probabilites for a sentence?

$p(\text{won} | \text{the underdog team})$

Maximum liklihood estimation (count and divide)

MLE

$$\frac{\text{count}(\text{the underdog team won})}{\text{count}(\text{the underdog team})}$$

Problematic
- Sentence may not appear
- need a large data set
- many way combining words into sentences
- Never going to have enough data to estimate the prob. of whole sentences

Simplifying assumption = The Markov assumption

$p(\text{won} \mid \text{the underdog team}) \approx p(\text{won} \mid \text{team})$

$p(w \mid \text{context})$ only depends on prev k words

NOT the whole context

N-gram
- N=2 bigram
- tri gram

$p(\text{won} \mid \text{underdog team}) \approx$

$w_i \quad w_{i-2} \quad w_{i-1}$

$$p(w_1 w_2 w_3 w_4 \dots w_n) \approx \prod_i p(w_i \mid w_{i-k} \dots w_{i-1})$$

num context words that we take into account

$$p(w_i \mid w_1, w_2, w_3) \approx p(w_i \mid w_{i-k} \dots w_{i-1})$$

How much history should we use in our Markov assumption?

n-gram models
- sequence size (= history plus current word)
- "word"

unigram = 1 word, no history

$$p(w_1, w_2, w_3, \dots w_n)$$

$$\prod p(w_i) = \frac{\text{count}(w_i)}{\text{total words in our corpus}}$$

bigram = 1 word, 1 as history

unigram = 1 word, no history

bigram = 1 word, 1 as history

trigram = 1 word, 2 " "

4 gram = 1 word, 3 " "

$\prod p(w_i) = \frac{count(w_i)}{\text{total words in our corpus}}$

$\prod p(w_i | w_{i-1})$

$\prod p(w_i | w_{i-2}, w_{i-1})$

$\prod p(w_i | w_{i-3}, w_{i-2}, w_{i-1})$

MLE

unigram $p(w_i) = \frac{count(w_i)}{len(corpus)} = \frac{100}{1 \text{ billion}}$

bigram $p(w_i | w_{i-1}) \quad \frac{count(w_{i-1}\ w_i)}{count(w_{i-1}\ *)}$

trigram $p(w_i | w_{i-2}\ w_{i-1}) \quad \frac{count(w_{i-2}\ w_{i-1}\ w_i)}{count(w_{i-2}\ w_{i-1}\ *)}$

Is MLE effective here?

count = 0 ←

Historical Notes

Markov chains/assumptions

1910s = Andrei Markov 1913

20k letters Eugene Onegin

1948 Claude Shanon uses n-gram to approximate English

1950s/60s Chompsky decries finite-state Markov models

1980s Fred Jelinek at IBM Watson uses n-grams for ASR

① Machine Translation

1993

$\arg\max_e p(e|f) = \arg\max_e p(e)\ p(f|e)$

1993

$$\arg\max_e p(e|f) = \arg\max_e p(e)\, p(f|e)$$

$p(e)$: LM ↗ $p(f|e)$: translation model ↑

② Stock market prediction

> Fred left IBM to JHU
- Renaissance Tech
  - Peter Brown
  - Robert Mercer / Trump's back / Brexit
    Cambridge analytica
    Breitbart News
    ACL Lifetime achievement award

Problems for MLE

zeros

Train
denied the allegations
.. .. reports
.. .. claims
.. - requests

Test
denied the memo

$p(\text{memo} \mid \text{denied the}) = 0$ and we assign 0 probability to sentences containing it

Out of Vocab (OOV)
<UNK> to deal with OOVs
fixed lexicon L of size V
normalize training data by replacing any word not in L w/ <UNK>

Avoid zeros with smoothing

Simplest smoothing ← Kneser-Ney smoothing, Stupid

Simplest smoothing ← Kneser-Ney smoothing, Stupid backoff

add one aka Laplace smoothing

$$P_{add-1}(w_i \mid w_{i-1}) = \frac{c(w_{i-1}\ w_i) + 1}{c(w_{i-1}) + V}$$

(0)

Backoff

Interpolation

$$p(w_i \mid w_{i-2}\ w_{i-1}) = \lambda_1\, p(w_i \mid w_{i-2}\ w_{i-1}) + \lambda_2\, p(w_i \mid w_{i-1}) + \lambda_3\, p(w_i)$$

$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$