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

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

\[ p(w_n | w_1, w_2, w_3, w_4) \]

"Language model" LM

Compute the joint prob:

\[ p(\text{the underdog Philadelphia Eagles won}) \]

Manipulate probabilities.
Manipulate probabilities

\[ p(B \mid A) = \frac{p(A, B)}{p(A)} \]

\[ p(A, B) = p(B \mid A) \cdot p(A) \]

\[ p(A, B, C, D) = p(A) \cdot p(B \mid A) \cdot p(C \mid A, B) \cdot p(D \mid A, B, C) \]

Chain Rule

\[ p(x_1, x_2, x_3, \ldots, x_n) = p(x_1) \cdot p(x_2 \mid x_1) \cdot p(x_3 \mid x_1, x_2) \cdot \ldots \cdot p(x_n \mid x_1, \ldots, x_{n-1}) \]

\[ p(w_1, w_2, w_3, w_4, \ldots, w_n) = \prod_{i} p(w_i \mid w_1, w_2, w_3, \ldots, w_{i-1}) \]

How do we estimate these probabilities for a sentence?

\[ p(\text{won} \mid \text{the underdog team}) \]

Maximum likelihood estimation (count and divide)

\[ \frac{\text{count (the underdog team won)}}{\text{count (the underdog team)}} \]
Problematic
- Sentence may not appear
- Need a large data set
- Many ways combining words into sentences
- Never going to have enough data to estimate the prob. of whole sentence

Simplifying assumption = The Markov assumption

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

\[ \approx \]

\[ p(\text{won} | \text{team context}) \]

\[ \approx \]

\[ p(\text{won} | \text{underdog team}) \]

\[ p(w, w_2, w_3, w_4 \ldots w_n) \approx \prod_i p(w_i | w_{i-k} \ldots w_{i-1}) \]

\[ p(w_i | w_1, w_2, w_3) \approx \]

\[ p(w_i | w_{i-k}, w_{i-1}) \]

How much history should we use in our Markov assumption?

n-gram models

↑ sequence size

(= history plus current word)

unigram = 1 word, no history

bigram = 1 word, 1 as history

n-gram = n words, n-1 as history

\[ p(w_1, w_2, w_3, \ldots w_n) \]

\[ \approx \frac{\text{count}(w_i)}{\text{total words in our corpus}} \]
unigram = 1 word, no history

bigram = 1 word, 1 as history

trigram = 1 word, 2 as history

4-gram = 1 word, 3 as history

MLE

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

\[ p(w_i | w_{i-1}) = \frac{\text{count}(w_i, w_{i-1})}{\text{count}(w_{i-1})} \]

\[ p(w_i | w_{i-2}, w_{i-1}) = \frac{\text{count}(w_i, w_{i-2}, w_{i-1})}{\text{count}(w_{i-2}, w_{i-1})} \]

Is MLE effective here?

Historical Notes

Markov chains/assumptions

1910s = Andrei Markov

1913 = letters Eugene Onegina

1948 = Claude Shannon uses n-gram to approximate English

1950s/60s = Chomsky decreses finite-state Markov models

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

1993 = Machine Translation

\[ \arg\max_{e} p(e | f) = \arg\max_{e} p(e) \cdot p(f|e) \]
1993
arg \max \ e \ p(e|f) = \ arg \ max \ e \ p(e) p(f|e)

2) Stock market prediction LM translation model

Fred left IBM to JHU
- Renaissance Tech
- Peter Brown
- Robert Marcus

Problems for MLE
zeros

Train denied the allegations
- report
- claims
- requests

Test denied the memo

\[ p(\text{memo} | \text{denied the}) = 0 \]
and we assign 0 probability to sentences containing it.

Out of Vocab (OOV)
\(<\text{UNK}>\) to deal with OOVs

Fixed lexicon L of size V
normalize training data by replacing any word not in L with \(<\text{UNK}>\)

Avoid zeros with smoothing

Simplest smoothing \(\times\) Kneser-Ney smoothing, stupid
Simplest smoothing is Kneser-Ney smoothing, stupid backoff

\[ P_{add-1}(w_i | w_{i-1}) = \frac{c(w_{i-1} \cdot w_i) + 1}{c(w_{i-1}) + \sqrt{v}} \]

Back off

Interpolation

\[ P(w_i | w_{i-2} \cdot w_{i-1}) = \lambda_1 P(w_i | w_{i-2} \cdot w_{i-1}) \]
\[ + \lambda_2 P(w_i | w_{i-1}) \]
\[ + \lambda_3 P(w_i) \]
\[ \lambda_1 + \lambda_2 + \lambda_3 = 1 \]