CMPU 366 · Computational Linguistics

Probability, $n$-grams, and Smoothing

8 February 2022
Assignment 2

Due 10 p.m. on Wednesday.

Worksheet 3

First part due now

Fill in the rest during class today, due tonight
Shared reading
The idea of a statistical language model is to compute the probability of a sequence of words. Why should we care about these probabilities?
Speech recognition

\[ P(\text{i saw a van}) > P(\text{eyes awe of an}) \]

Spelling correction

The office is about fifteen minuets from my house.

\[ P(\text{about fifteen minutes from}) > P(\text{about fifteen minuets from}) \]

Machine translation

Translating *The doctor recommended a cat scan*,

\[ P(\text{La doctora recomendó una tomografía}) > P(\text{La doctora recomendó una exploración del gato}) \]

And more!
These are examples of computing

\[ P(w_1, w_2, w_3, w_4, w_5), \]
the probability of a sequence,

but language models also let us compute

\[ P(w_5 \mid w_1, w_2, w_3, w_4), \]
the probability of a word given some previous words.

Why would that be useful?
why
why is the sky blue
why women kill
why is my poop green
why were chainsaws invented
why do cats purr
Hi! I'm heading to the store.
P(The other day I was walking along and saw a lizard)?

= P(The, other, day, I, was, walking, along, and, saw, a, lizard)

= P(The) P(other | the) P(day | the other) P(I | The other day) ...

= 🤔 How do we compute this?
To estimate conditional probabilities, we use a text corpus that we’ve tokenized and we do some counting!

\[
P(lizard \mid \text{The other day I was walking along and saw a})
\]

\[
= \frac{C(\text{The other day I was walking along and saw a lizard})}{C(\text{The other day I was walking along and saw a})}
\]

\(C(x)\) is the count of how many times \(x\) occurred in the corpus
In practice, we make a simplifying *Markov assumption* that we can predict the probability of a future event without looking too far into the past, e.g.,

\[
P(lizard \mid \text{The, other, day, I, was, walking, along, and, saw, a}) \\
\approx P(lizard \mid \text{saw, a})
\]
We can estimate the true probabilities using *n-grams* – sequences of text that are always $n$ units (e.g., words) long.
Colorless green ideas sleep furiously.
Colorless green ideas sleep furiously.

unigrams

<s>
  Colorless
  green
  ideas
  sleep
  furiously
.
</s>
Colorless green ideas sleep furiously.

bigrams

<s> Colorless
Colorless green
green ideas
ideas sleep
sleep furiously
furiously .
</s>
Colorless green ideas sleep furiously.

trigrams

<s> <s> Colorless
<s> Colorless green
Colorless green ideas
green ideas sleep
ideas sleep furiously
sleep furiously .
furiously . </s>
Colorless green ideas sleep furiously.

4-grams

<s> <s> <s> Colorless
<s> <s> Colorless green
<s> Colorless green ideas
Colorless green ideas sleep
green ideas sleep furiously
ideas sleep furiously .
sleep furiously . </s>
What's the best value of $n$?

That is, how many previous words do we need?
Given any choice of $n$, are $n$-grams a sufficient model of language?
Language has *long-distance dependencies*:

*The computer / computers which I had just put into the machine room on the fifth floor is / are crashing.***

But we can often get away with *n*-gram models.
We estimate the probabilities of $n$-grams using the maximum likelihood estimate (MLE) – the estimate that maximizes the likelihood of the training data given the model.
For unigram probabilities,

that’s the fraction of times the word occurs in the corpus:

$$P(w_i) = \frac{C(w_i)}{|V|}$$

For bigram probabilities,

that’s the number of times that word follows the other word divided by the number of times the other word occurs in the corpus:

$$P(w_i \mid w_{i-1}) = \frac{C(w_{i-1}, w_i)}{C(w_{i-1})}$$
Shakespeare as a corpus

884 647 tokens

29 066 types

Shakespeare produced 300 000 bigram types out of 844 million possible bigrams.

Using MLE, 99.96% of the possible bigrams will have a probability of 0.
If we assume every sequence of words we’ll ever see occurs in our training data, that’s a kind of overfitting.

We want to learn from the training data but also generalize.
How do you handle unseen $n$-grams?

*Smoothing:* Redistribute probability mass from seen to unseen $n$-grams.

*Backoff:* Use lower-order $n$-grams when higher-order ones aren’t available.

*Interpolation:* Use lower-order and higher-order ones together, with weights.
You’ll get a better feel for $n$-gram counts and smoothing when you do Assignment 3!
What are your questions or observations from the shared reading?
Further reading
Groups
Acknowledgments

This class incorporates material from:

Jurafsky & Martin, *Speech and Language Processing*, 3rd ed. draft