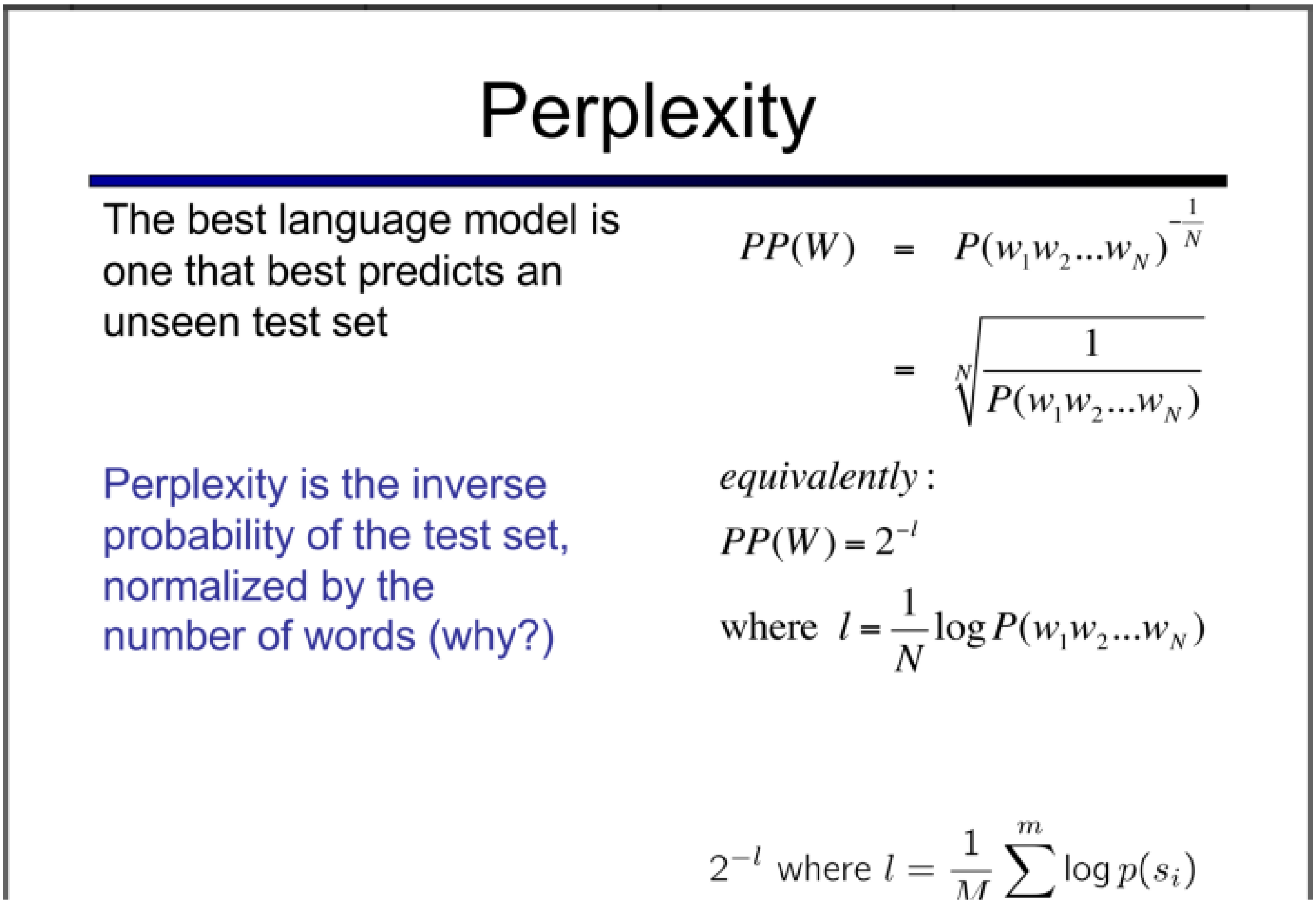
Perplexity Intuition (and its derivation)

Never be perplexed again by perplexity.

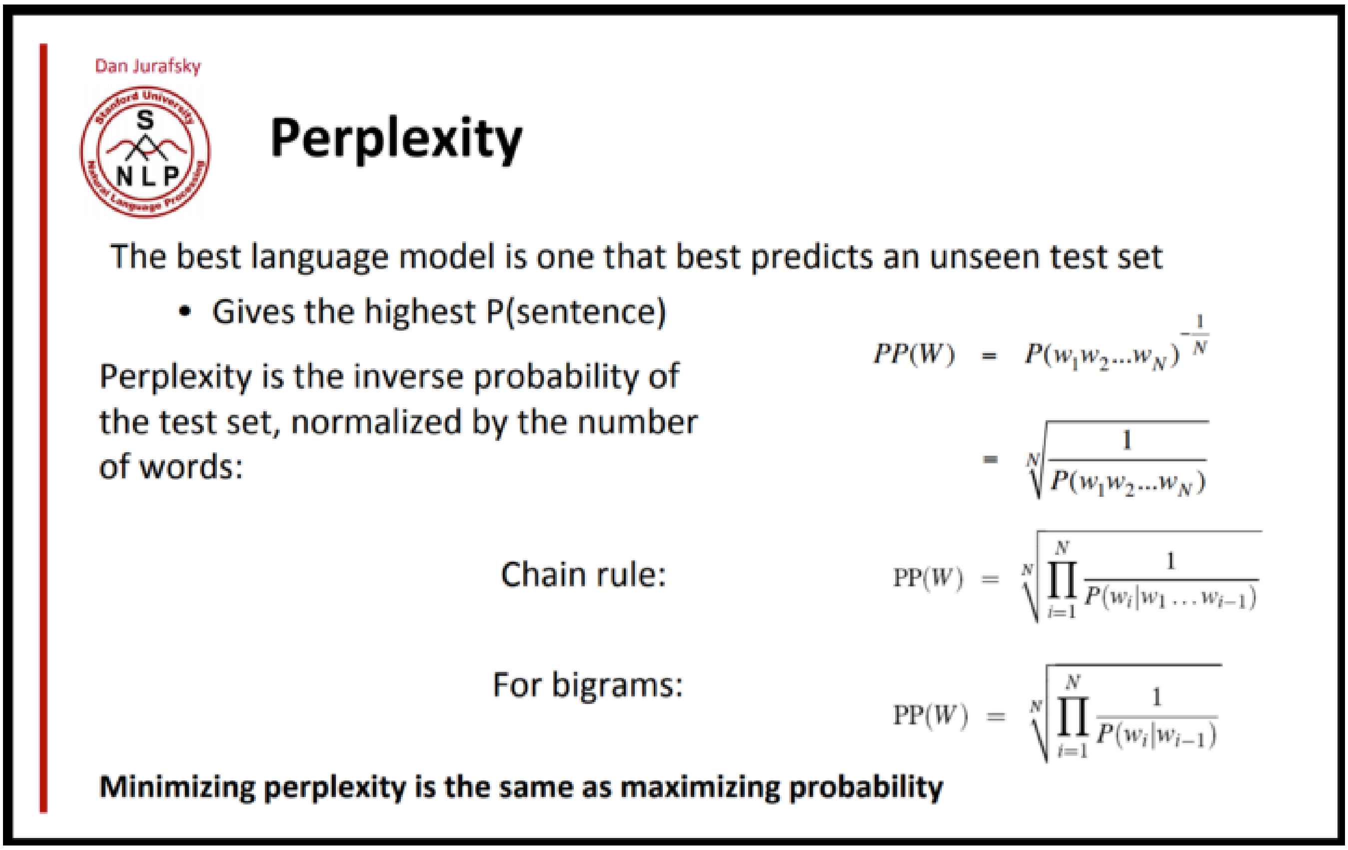
Aerin Kim Oct 12, 2018 · 4 min read ★

You might have seen something like this in an NLP class:



A slide from Dr. Luke Zettlemoyer's NLP class

Or



A slide of CS 124 at Stanford (Dr. Dan Jurafsky)

During the class, we don't really spend time to derive the perplexity. Maybe perplexity is a basic concept that you probably already know? This post is for those who don't.

In general, perplexity is a measurement of **how well a probability model predicts a sample**. In the context of Natural Language Processing, perplexity is one way to **evaluate language models**.

But why is perplexity in NLP defined the way it is?

$$PP(W) = P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}}$$

If you look up **the perplexity of a discrete probability distribution** in Wikipedia:

$$2^{H(p)} = 2^{-\sum_x p(x) \log_2 p(x)}$$

from https://en.wikipedia.org/wiki/Perplexity

where **H(p) is the entropy of the distribution p(x)** and *x* is a random variable over all possible events.

In the previous post, we derived **H(p)** from scratch and intuitively showed **why entropy is the average number of bits that we need to encode the information.** If you don't understand **H(p)**, please read this ⇩ before reading further.

The intuition behind Shannon's Entropy
[WARNING: TOO EASY!]

Now we agree that H(p) =-Σ p(x) log p(x).

Then, perplexity is just an exponentiation of the entropy!

Yes. Entropy is the average number of bits to encode the information contained in a random variable, so the exponentiation of the entropy should be **the total amount of all possible information,** or more precisely, the weighted average number of choices a random variable has**.**

For example, **if the average sentence in the test set could be coded in 100 bits, the model perplexity is 2^{100} per sentence.**

Let’s confirm that the definition in Wikipedia matches to the one in the slides.

$$\underset{\text{(of our model } q\text{)}}{\text{Perplexity}} = 2^{\text{entropy}} = 2^{\text{avg. \# of bits}}$$

tot # of words

x can be any R.V. In NLP, it's each word.

$$= 2^{-\sum_{i=1}^{N} \underbrace{p(x_i)}_{\text{real dist.}} \cdot \underbrace{\log_2 q(x_i)}_{\text{our prediction!}}}$$

$$= e^{-\sum_{i=1}^{N} p(x_i) \cdot \ln q(x_i)}$$

We assume all words have the same frequency.

$$= e^{-\sum_{i=1}^{N} \frac{1}{N} \cdot \ln q(x_i)}$$

$$= q(x_1)^{-\frac{1}{N}} \cdot q(x_2)^{-\frac{1}{N}} \cdots q(x_N)^{-\frac{1}{N}}$$

$$= \prod_{i=1}^{N} q(x_i)^{-\frac{1}{N}} = \sqrt[N]{\frac{1}{q(x_1)\, q(x_2) \cdots q(x_N)}} \quad \blacksquare$$

Where

p : A probability distribution that we want to model. A training sample is drawn from **p** and it’s unknown distribution.

q : A proposed probability model. Our prediction.

> *We can evaluate our prediction* ***q*** *by testing against samples drawn from* ***p****. Then* ***it’s basically calculating the cross-entropy****. In the derivation above, we assumed all words have the same probability (1 / # of words) in* ***p****.*

concept of smoothing in NLP was introduced.

- If we use a uniform probability model for **q** (simply 1/N for all words), the perplexity will be equal to the vocabulary size.
- The derivation above is for illustration purpose only in order to reach the formula in UW/Stanford slides. In both slides, it assumes that we are calculating the perplexity of the entire corpus using a unigram model and there is no duplicated word. (It assumes the # of total words (N) is the same as the number of unique words.) Also, it assumes all words have the same probability 1/N. These are not realistic assumptions.

Takeaway

- Less entropy (or less disordered system) is favorable over more entropy. Because predictable results are preferred over randomness. This is why people say **low perplexity is good and high perplexity is bad since the perplexity is the exponentiation of the entropy** (and you can safely think of the concept of perplexity as entropy).
- A language model is a probability distribution over sentences. And the best language model is one that best predicts an unseen test set.
- **Why do we use perplexity instead of entropy?**
  If we think of perplexity as a **branching factor** (the weighted average number of choices a random variable has), **then that number is easier to understand than the entropy.** I found this surprising because I thought there will be more profound reasons. I asked Dr. Zettlemoyer if there is any other reason other than easy interpretability. His answer was "I think that is it! **It is largely historical since lots of other metrics would be reasonable to use as well!**"

If you like my post, could you please clap? It gives me motivation to write more. :)