N-Gram Language Models

Pretend I have a corpus of sentences:

- I eat lunch today.
- They like to eat.
- I like lunch.
- Lunch was great today.

it is not BOW, consider sequence of words.

Introduces sequence

We want to build some sort of model to predict, given some test words, the most likely words to appear. For instance

I like ____ } our test sentence

Based only on our corpus, we see that the word like is followed by the word lunch 50% of the time, and to 50% of the time. In fact, we can construct a transition matrix:

	I	eat	lunch	today	to	they	was	great	like
I	0	0.5	0	0	0	0	0	0	0.5
eat	0	0	1	0	0	0	0	0	0
lunch	0	0	0	0.5	0	0	0.5	0	0
today	0	0	0	0	0	0	0	0	0
to	0	1	0	0	0	0	0	0	0
they	0	0	0	0	0	0	0	0	1
was	0	0	0	0	0	0	0	1	0
great	0	0	0	1	0	0	0	0	0
like	0	0	0.5	0	0.5	0	0	0	0

100% chance the next word after to is eat

50% chance next word after like is lunch

NXN in shape

N: no. of unique words.

This transition matrix will help us determine the likelihood of certain bigrams. For instance, like to:

$$p(x_i = \text{to} \mid x_{i-1} = \text{like}) = 0.5$$

However, our transition matrix isn't yet correct. Certain rows, like today, do not add up to 1. This is because today is always the end of a sentence, so no words appear after it. We need to account for this here:

we add these as placeholders so we can model the start/end of a sentence

START I eat lunch today END
START They like to eat. END
START I like lunch. END
START Lunch was great today END

We add a "START" and "END" placeholder and treat it like any other word:

Our new transition matrix will look like this:

	START	I	eat	lunch	today	to	they	was	great	like	END
START	0	0.5	0	0.25	0	0	0.25	0	0	0	0
I	0	0	0.5	0	0	0	0	0	0	0.5	0
eat	0	0	0	0.5	0	0	0	0	0	0	0.5
lunch	0	0	0	0	0.3	0	0	0.3	0	0	0.3
today	0	0	0	0	0	0	0	0	0	0	1
to	0	0	1	0	0	0	0	0	0	0	0
they	0	0	0	0	0	0	0	0	0	1	0
was	0	0	0	0	0	0	0	0	1	0	0
great	0	0	0	0	1	0	0	0	0	0	0
like	0	0	0	0.5	0	0.5	0	0	0	0	0
END	0	0	0	0	0	0	0	0	0	0	0

Now let's use this to measure the probability of a sentence: I like to eat.

$$p(X_0=\text{START}, X_1=\text{I}, X_2=\text{like}, X_3=\text{to}, X_4=\text{eat}, X_5=\text{END})$$

$$= \prod_{i=1}^{N+1} p(X_i \mid X_{i-1})$$

the likelihood of a word X_i following a word X_{i-1}

$$= p(X_1 = I \mid X_0 = START) \times$$

$$p(X_2 = like \mid X_1 = I) \times$$

$$p(X_3 = to \mid X_2 = like) \times$$

$$p(X_4 = eat \mid X_3 = to) \times$$

$$p(X_5 = END \mid X_4 = eat)$$

{ like, eat, to }

	START	I	eat	lunch	today	to	they	was	great	like	END
START	0	0.5	0	0.25	0	0	0.25	0	0	0	0
I	0	0	0.5	0	0	0	0	0	0	0.5	0
eat	0	0	0	0.5	0	0	0	0	0	0	0.5
lunch	0	0	0	0	0.3	0	0	0.3	0	0	0.3
today	0	0	0	0	0	0	0	0	0	0	1
to	0	0	1	0	0	0	0	0	0	0	0
they	0	0	0	0	0	0	0	0	0	1	0
was	0	0	0	0	0	0	0	0	1	0	0
great	0	0	0	0	1	0	0	0	0	0	0
like	0	0	0	0.5	0	0.5	0	0	0	0	0
END	0	0	0	0	0	0	0	0	0	0	0

$= 0.5 \times 0.5 \times 0.5 \times 1 \times 0.5$

$= 0.0625$ (this is the probability of this sentence!)

Let's take a gibberish sentence, like

START Lunch they I today END

$P(X_1 = \text{lunch} \mid X_0 = \text{START}) \times \longrightarrow 0.25$

$P(X_2 = \text{they} \mid X_1 = \text{lunch}) \times \longrightarrow 0$

$P(X_3 = \text{I} \mid X_2 = \text{they}) \times \longrightarrow 0$

$P(X_4 = \text{today} \mid X_3 = \text{I}) \times \longrightarrow 0$

$P(X_5 = \text{END} \mid X_4 = \text{today}) \longrightarrow 1$

$= .25 \times 0 \times 0 \times 0 \times 1$

$= 0$

This sentence, according to our model, is essentially impossible.

Perplexity:

short sentences will have higher probability.

We have our sentence probability, but not all sentences will be the same length. We can't compare the probability of a sentence of length 4 words with one that is of length 12. Therefore, we'll use perplexity as our evaluation metric:

$$\text{Perplexity} = \frac{1}{\sqrt[N]{p(\text{sentence})}}$$

o/p of language models.

N = # of words in sentence

p(sentence) = probability of the sentence.

Going back to our test sentence:

I like to eat.

We calculated p(sentence) = 0.0625.

Therefore our perplexity is

$$\text{Perplexity} = \frac{1}{\sqrt[4]{0.0625}}$$

$$= 2.$$

Minimizing the perplexity is equivalent to maximizing the probability of a sentence. Thus, the lower the perplexity, the more likely a sentence is to be found in natural language.

'The' may or may not be removed as a stopword. Uncommon words may all be replaced w/ _OOV_ and this reduces overfitting.