LING439/539 - Statistical NLP
Chapter 6. Statistical inferences: n-gram model over sparse data

Tuesday, September 6 2016

Word prediction

Please turn your homework ....

What word is likely to follow the above sentence is *in*, or possibly *over*... etc.

N-gram models

N-token sequence of words:

- 2-gram (bigram): a two-word sequence of words *please turn*, *turn your*, ...
- 3-gram (trigram): a three-word sequence of words *please turn your*, *turn your homework*, ...
- ...

$\Rightarrow$ language models or LMs

speech recognition, handwriting recognition, (statistical) machine translation, spelling correction, etc.

Word counting in corpus

Word type vs. word token

They picnicked by the pool then lay back on the grass and looked at the stars

16 tokens vs. 14 types

- `cat corpus | tr " " '\012' | wc -l`
- `cat corpus | tr " " '\012' | sort | uniq -c | wc -l`

$P(w|h)$

$P(w|h)$ is a probability of a word w given some history h.

Suppose the history h is "*its water is so transparent that*" and we want to know the probability that the next word is *the*:

$P(\textbf{the}|\textbf{its water is so transparent that})$
How can we compute this probability ?

$$P(\text{the}|\text{its water is so transparent that}) =$$

$$\frac{C(\text{its water is so transparent that the})}{C(\text{its water is so transparent that})}$$

Try using the Web:

- *"its water is so transparent that the"*
- *"its water is so transparent that"*

$P(\text{the}|\text{its water is so transparent that}) =$

$$\frac{C(\text{its water is so transparent that the})}{C(\text{its water is so transparent that})}$$

Try using the Web:

- *"its water is so transparent that the"*: About 5,130 results (0.47 seconds)
- *"its water is so transparent that"*: About 6,710 results (0.25 seconds)

Accessed on September 5 2016

$$\frac{C(\text{its water is so transparent that the})}{C(\text{its water is so transparent that})} = \frac{5130}{6710} = 0.7645305514158$$

However, we may have counts of **zeros** ....

$$\frac{C(\text{its water is so transparent that the})}{C(\text{its water is so transparent that})} = \frac{0}{6710} = \mathbf{0}$$

Chain rule of probability

We represent a sequence of N words either as $w_1 ... w_n$ or w_1^n

For the joint probability of each word in a sequences, we use $P(w_1, w_2, ..., w_n)$

$$
\begin{aligned}
P(w_1^n) &= P(w_1)P(w_2|w_1)P(w_3|w_1^2)...P(w_n|w_1^{n-1}) \\
&= \prod_{k=1}^{n} P(w_k|w_1^{k-1})
\end{aligned}
$$

Actually, using the chain rule doesn't really seem to help us. We still don't know any way to compute the exact probability of a word given a long sequence of preceding words ($P(w_n|w_1^{n-1})$).

Bigram

The bigram model approximates the probability of a word by using **only** the conditional probability of the preceding word.

$P(\text{the}|\text{its water is so transparent that})$
$\approx P(\text{the}|\text{that})$

Markov assumption

$P(w_n|w_1^{n-1}) \approx P(w_n|w_{n-1})$

Generalization of the Markov assumption for N-gram

Markov assumption for bigram

$$P(w_n|w_1^{n-1}) \approx P(w_n|w_{n-1})$$

Markov assumption for trigram

$$P(w_n|w_1^{n-1}) \approx P(w_n|w_{n-2}w_{n-1}) = P(w_n|w_{n-2}^{n-1})$$

Markov assumption for N-gram

$$P(w_n|w_1^{n-1}) \approx \mathbf{?}$$

Generalization of the Markov assumption for N-gram

Markov assumption for bigram

$$P(w_n|w_1^{n-1}) \approx P(w_n|w_{n-1})$$

Markov assumption for trigram

$$P(w_n|w_1^{n-1}) \approx P(w_n|w_{n-2}w_{n-1}) = P(w_n|w_{n-2}^{n-1})$$

Markov assumption for N-gram

$$P(w_n|w_1^{n-1}) \approx P(w_n|w_{n-N+1}^{n-1})$$

Maximum likelihood estimation for the bigram probability

How do we estimate bigram probabilities ?
$\Rightarrow$ Maximum likelihood estimation (MLE).

$$
\begin{aligned}
P(w_n|w_{n-1}) &= \frac{C(w_{n-1}w_n)}{\sum_w C(w_{n-1}w)} \\
&= \frac{C(w_{n-1}w_n)}{C(w_{n-1})}
\end{aligned}
$$

The sum of all bigram counts that starts with a given word w_{n-1} **must** be equal to the unigram count for the word w_{n-1}.

Very small corpus

```
<s> I am Sam </s>
<s> Sam I am </s>
<s> I do not like green eggs and ham </s>
```

```
3 I        2 <s> I
3 <s>      2 I am
3 </s>     ...
2 am
2 Sam
1 not
1 like
1 ham
1 green
1 eggs
1 do
1 and
```

$P(\texttt{I}|\texttt{<s>}) =$ $P(\texttt{Sam}|\texttt{<s>}) =$ $P(\texttt{am}|\texttt{I}) =$

$P(\texttt{</s>}|\texttt{Sam}) =$ $P(\texttt{Sam}|\texttt{am}) =$ $P(\texttt{do}|\texttt{I}) =$

$$P(\texttt{I}|\texttt{<s>}) = \frac{2}{3} \qquad P(\texttt{Sam}|\texttt{<s>}) = \frac{1}{3} \qquad P(\texttt{am}|\texttt{I}) = \frac{2}{3}$$

$$P(\texttt{</s>}|\texttt{Sam}) = \frac{1}{2} \qquad P(\texttt{Sam}|\texttt{am}) = \frac{1}{2} \qquad P(\texttt{do}|\texttt{I}) = \frac{1}{3}$$

Berkeley Restaurant Project

A dialogue system that answered questions about a database of restaurants in Berkeley, California. It contains 9,332 sentences.

```
33_1_0001   okay let's see i want to go to a thai restaurant .
            [uh] with less than ten dollars per person
33_1_0002   <i> <like> <to> <eat> [uh] i like to eat at lunch
            time .  so that would be eleven a_m to one p_m
33_1_0003   i don't want to walk for more than five minutes
33_1_0004   tell me more about the [uh] na- nakapan [uh]
            restaurant on martin luther king
33_1_0005   i like to go to a hamburger restaurant
33_1_0006   let's start again
33_1_0007   i like to get a hamburger at an american restaurant
33_1_0008   i'd like to eat dinner .  and i don't mind walking
            [uh] .  for half an hour
```

https://github.com/wooters/berp-trans

	i	want	to	eat	chinese	foot	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

Bigram counts for eight of the words (out of $V = 1446$) in the Berkeley Restaurant Project corpus of 9332 sentences.

(unigram)	i	want	to	eat	chinese	foot	lunch	spend
	2533	927	2417	746	158	1093	341	278

	i	**want**	**to**	**eat**	**chinese**	**foot**	**lunch**	**spend**
i	.002	.33	0	.0036	0	0	0	.00079
want	.0022	0	.66	0.0011	.0065	.0065	.0054	.0011
to	.00083	0	.0017	.28	.00083	0	.0025	.087
eat	0	0	.0027	0	.021	.0027	.0056	0
chinese	.0063	0	0	0	0	.52	.0063	0
food	.014	0	.014	0	.00092	.0037	0	0
lunch	.0059	0	0	0	0	.0029	0	0
spend	.0036	0	.0036	0	0	0	0	0

Bigram probabilities for eight of the words in the Berkeley Restaurant Project corpus of 9332 sentences.

The probability of the sentence *I want English food*:

$P($`<s> I want English food </s>`$)$

	i	**want**	**to**	**eat**	**chinese**	**foot**	**lunch**	**spend**
i	.002	.33	0	.0036	0	0	0	.00079
want	.0022	0	.66	0.0011	.0065	.0065	.0054	.0011
to	.00083	0	.0017	.28	.00083	0	.0025	.087
eat	0	0	.0027	0	.021	.0027	.0056	0
chinese	.0063	0	0	0	0	.52	.0063	0
food	.014	0	.014	0	.00092	.0037	0	0
lunch	.0059	0	0	0	0	.0029	0	0
spend	.0036	0	.0036	0	0	0	0	0

Bigram probabilities for eight of the words in the Berkeley Restaurant Project corpus of 9332 sentences.

The probability of the sentence *I want English food*:

$P(\texttt{<s> I want English food </s>})$

$= P(\texttt{i}|\texttt{<s>})\ P(\texttt{want}|\texttt{i})\ P(\texttt{english}|\texttt{want})\ P(\texttt{food}|\texttt{english})\ P(\texttt{</s>}|\texttt{food})$