

Natural Language Processing

Tutorial 3: N-gram and Language Model

Dr. Sun Aixin



Question I

- Given the following three word sequences (the corpus)
 - very good tennis player in US Open
 - tennis player US Open
 - tennis player qualify play US Open

- (i) Build a table of bigram counts from the word sequences
- (ii) Compute the bigram probabilities using Laplace smoothing



Bigram model

Review

➤ With bigram model $P(w_n|w_{1:n-1}) \approx P(w_n|w_{n-1})$

- Our example

$$P(w|h) = P(it|I \text{ will make}) \approx P(it|make)$$

- $P(w_{1:n}) = \prod_{k=1}^n P(w_k|w_{1:k-1}) \approx \prod_{k=1}^n P(w_k|w_{k-1})$

➤ Now, how to compute $P(w_n|w_{n-1})$, like $P(it|make)$?

- Estimate bigram probabilities by **maximum likelihood estimation** or MLE
- We estimate $P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$ where $C(\cdot)$ is the count, or frequency



make decisions.
make sure ...
make it right
make it happen
make toys.

$$C(make) = 5$$

$$C(make \text{ it}) = 2$$

$$P(it|make) = 0.4$$



Answer Q1. (i)

➤ Given the corpus, build a table of bigram counts from the word sequences

- very good tennis player in US Open
- tennis player US Open
- tennis player qualify play US Open

➤ We should consider the **sentence boundaries** as tokens.

- <s> very good tennis player in US Open </s>
- <s> tennis player US Open </s>
- <s> tennis player qualify play US Open </s>

➤ Both <s> and </s> are counted as tokens.

make decisions
make sure
make it right
make it happen
make toys



$$C(\textit{make}) = 5$$

$$C(\textit{make it}) = 2$$

$$P(\textit{it}|\textit{make}) = 0.4$$



Answer Q1. (i)

<s> very good tennis player in US Open </s>

<s> tennis player US Open </s>

<s> tennis player qualify play US Open </s>

w_n

w_{n-1}

	very	good	tennis	player	in	us	open	qualify	play	</s>
<s>	1	0	2	0	0	0	0	0	0	0
very	0	1	0	0	0	0	0	0	0	0
good	0	0	1	0	0	0	0	0	0	0
tennis	0	0	0	3	0	0	0	0	0	0
player	0	0	0	0	1	1	0	1	0	0
in	0	0	0	0	0	1	0	0	0	0
us	0	0	0	0	0	0	3	0	0	0
open	0	0	0	0	0	0	0	0	0	3
qualify	0	0	0	0	0	0	0	0	1	0
play	0	0	0	0	0	1	0	0	0	0

Answer Q1. (i)

<s> very good tennis player in US Open </s>

<s> tennis player US Open </s>

<s> tennis player qualify play US Open </s>

w_n

	very	good	tennis	player	in	us	open	qualify	play	</s>	w_{n-1}	count
<s>	1	0	2	0	0	0	0	0	0	0	<s>	3
very	0	1	0	0	0	0	0	0	0	0	very	1
good	0	0	1	0	0	0	0	0	0	0	good	1
tennis	0	0	0	3	0	0	0	0	0	0	tennis	3
player	0	0	0	0	1	1	0	1	0	0	player	3
in	0	0	0	0	0	1	0	0	0	0	in	1
us	0	0	0	0	0	0	3	0	0	0	us	3
open	0	0	0	0	0	0	0	0	0	3	open	3
qualify	0	0	0	0	0	0	0	0	1	0	qualify	1
play	0	0	0	0	0	1	0	0	0	0	play	1

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$



	very	good	tennis	player	in	us	open	qualify	play	</s>
<s>	1	0	2	0	0	0	0	0	0	0
very	0	1	0	0	0	0	0	0	0	0
good	0	0	1	0	0	0	0	0	0	0
tennis	0	0	0	3	0	0	0	0	0	0
player	0	0	0	0	1	1	0	1	0	0
in	0	0	0	0	0	1	0	0	0	0
us	0	0	0	0	0	0	3	0	0	0
open	0	0	0	0	0	0	0	0	0	3
qualify	0	0	0	0	0	0	0	0	1	0
play	0	0	0	0	0	1	0	0	0	0

w_{n-1}	count
<s>	3
very	1
good	1
tennis	3
player	3
in	1
us	3
open	3
qualify	1
play	1

	very	good	tennis	player	in	us	open	qualify	play	</s>
<s>	2	1	3	1	1	1	1	1	1	1
very	1	2	1	1	1	1	1	1	1	1
good	1	1	2	1	1	1	1	1	1	1
tennis	1	1	1	4	1	1	1	1	1	1
player	1	1	1	1	2	2	1	2	1	1
in	1	1	1	1	1	2	1	1	1	1
us	1	1	1	1	1	1	4	1	1	1
open	1	1	1	1	1	1	1	1	1	4
qualify	1	1	1	1	1	1	1	1	2	1
play	1	1	1	1	1	2	1	1	1	1

w_{n-1}	count
<s>	13
very	11
good	11
tennis	13
player	13
in	11
us	13
open	13
qualify	11
play	11



	very	good	tennis	player	in	us	open	qualify	play	</s>
<s>	1	0	2	0	0	0	0	0	0	0
very	0	1	0	0	0	$P(w_n w_{n-1}) = \frac{C(w_{n-1}w_n) + 1}{C(w_{n-1}) + V}$				
good	0	0	1	0	0					
tennis	0	0	0	3	0					
player	0	0	0	0	1	1	0	1	0	0
in	0	0	0	0	0	1	0	0	0	0
us	0	0	0	0	0	0	3	0	0	0
open	0	0	0	0	0	0	0	0	0	3
qualify	0	0	0	0	0	0	0	0	1	0
play	0	0	0	0	0	1	0	0	0	0

w_{n-1}	count
<s>	3
very	1
good	1
tennis	3
player	3
in	1
us	3
open	3
qualify	1
play	1

	very	good	tennis	player	in	us	open	qualify	play	</s>
<s>	2/13	1/13	3/13	1/13	1/13	1/13	1/13	1/13	1/13	1/13
Very	1/11	2/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11
Good	1/11	1/11	2/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11
tennis	1/13	1/13	1/13	4/13	1/13	1/13	1/13	1/13	1/13	1/13
player	1/13	1/13	1/13	1/13	2/13	2/13	1/13	2/13	1/13	1/13
in	1/11	1/11	1/11	1/11	1/11	2/11	1/11	1/11	1/11	1/11
us	1/13	1/13	1/13	1/13	1/13	1/13	4/13	1/13	1/13	1/13
open	1/13	1/13	1/13	1/13	1/13	1/13	1/13	1/13	1/13	4/13
qualify	1/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11	2/11	1/11
play	1/11	1/11	1/11	1/11	1/11	2/11	1/11	1/11	1/11	1/11

w_{n-1}	count
<s>	13
very	11
good	11
tennis	13
player	13
in	11
us	13
open	13
qualify	11
play	11



	very	good	tennis	player	in	us	open	qualify	play	</s>
<s>	1	0	2	0	0	0	0	0	0	0
very	0	1	0	0	0	0	0	0	0	0
good	0	0	1	0	0	0	0	0	0	0
tennis	0	0	0	3	0	0	0	0	0	0
player	0	0	0	0	1	1	0	1	0	0
in	0	0	0	0	0	1	0	0	0	0
us	0	0	0	0	0	0	3	0	0	0
open	0	0	0	0	0	0	0	0	0	3
qualify	0	0	0	0	0	0	0	0	1	0
play	0	0	0	0	0	1	0	0	0	0

w_{n-1}	count
<s>	3
very	1
good	1
tennis	3
player	3
in	1
us	3
open	3
qualify	1
play	1

	very	good	tennis	player	in	us	open	qualify	play	</s>
<s>	2/13	1/13	3/13	1/13	1/13	1/13	1/13	1/13	1/13	1/13
Very	1/11	2/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11
Good	1/11	1/11	2/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11
tennis	1/13	1/13	1/13	4/13	1/13	1/13	1/13	1/13	1/13	1/13
player	1/13	1/13	1/13	1/13	2/13	2/13	1/13	2/13	1/13	1/13
in	1/11	1/11	1/11	1/11	1/11	2/11	1/11	1/11	1/11	1/11
us	1/13	1/13	1/13	1/13	1/13	1/13	4/13	1/13	1/13	1/13
open	1/13	1/13	1/13	1/13	1/13	1/13	1/13	1/13	1/13	4/13
qualify	1/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11	2/11	1/11
play	1/11	1/11	1/11	1/11	1/11	2/11	1/11	1/11	1/11	1/11

w_{n-1}	count
<s>	13
very	11
good	11
tennis	13
player	13
in	11
us	13
open	13
qualify	11
play	11



	very	good	tennis	player	in	us	open	qualify	play	</s>
<s>	1	0	2	0	0	0	0	0	0	0
very	0	1	0	0	0	0	0	0	0	0
good	0	0	1	0	0	0	0	0	0	0
tennis	0	0	0	3	0	0	0	0	0	0
player	0	0	0	0	1	1	0	1	0	0
in	0	0	0	0	0	1	0	0	0	0
us	0	0	0	0	0	0	3	0	0	0
open	0	0	0	0	0	0	0	0	0	3
qualify	0	0	0	0	0	0	0	0	1	0
play	0	0	0	0	0	1	0	0	0	0

(i) Build a table of bigram counts from the word sequences

(ii) Compute the bigram probabilities using Laplace smoothing

	very	good	tennis	player	in	us	open	qualify	play	</s>
<s>	2/13	1/13	3/13	1/13	1/13	1/13	1/13	1/13	1/13	1/13
Very	1/11	2/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11
Good	1/11	1/11	2/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11
tennis	1/13	1/13	1/13	4/13	1/13	1/13	1/13	1/13	1/13	1/13
player	1/13	1/13	1/13	1/13	2/13	2/13	1/13	2/13	1/13	1/13
in	1/11	1/11	1/11	1/11	1/11	2/11	1/11	1/11	1/11	1/11
us	1/13	1/13	1/13	1/13	1/13	1/13	4/13	1/13	1/13	1/13
open	1/13	1/13	1/13	1/13	1/13	1/13	1/13	1/13	1/13	4/13
qualify	1/11	1/11	1/11	1/11	1/11	1/11	1/11	1/11	2/11	1/11
play	1/11	1/11	1/11	1/11	1/11	2/11	1/11	1/11	1/11	1/11

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n) + 1}{C(w_{n-1}) + V}$$



Question 2

- Write out the equation for trigram probability estimation, and use the equation to compute the trigram probability for $P(US | tennis\ player)$ and $P(player | good\ tennis)$ according to the corpus given in Q1.

- $$P(w_n | w_{n-1}, w_{n-2}) = \frac{C(w_{n-2}w_{n-1}w_n)}{C(w_{n-2}w_{n-1})}$$



Answer 2

➤ Dataset

- very good tennis player in US open
- tennis player US Open
- tennis player qualify play US Open

Dataset with <s> and </s>, for trigram

- <s> <s> very good tennis player in US open </s>
- <s> <s> tennis player US Open</s>
- <s> <s>tennis player qualify play US Open </s>

$$P(w_n | w_{n-1}, w_{n-2}) = \frac{C(w_{n-2}w_{n-1}w_n)}{C(w_{n-2}w_{n-1})}$$

- $P(US | tennis\ player) = 1/3$
- $P(player | good\ tennis) = 1/1$

Think about smoothing



Question 3

- Given the bigram probability in the following table, compute the probability of “I eat Chinese food” by using the table. Explain how you compute the probability.
- **State your assumptions** and if more probability values are needed, you may use random values.

	i	want	to	eat	chinese	food	lunch	spend
i	0.002	0.33	0	0.0036	0	0	0	0.00079
want	0.0022	0	0.66	0.0011	0.0065	0.0065	0.0054	0.0011
to	0.00083	0	0.0017	0.28	0.00083	0	0.0025	0.087
eat	0	0	0.0027	0	0.021	0.0027	0.056	0
chinese	0.0063	0	0	0	0	0.52	0.0063	0
food	0.014	0	0.014	0	0.00092	0.0037	0	0
lunch	0.0059	0	0	0	0	0.0029	0	0
spend	0.0036	0	0.0036	0	0	0	0	0

Answer 3

If not considering $\langle s \rangle$ and $\langle /s \rangle$:

➤ $P(I \text{ eat Chinese food})$
= $P(\text{eat}|I) * P(\text{Chinese}|I \text{ eat}) * P(\text{food}|I \text{ eat Chinese})$

➤ Chain rules: Independence Assumption – bigram

➤ $P(I \text{ eat Chinese food})$
= $P(\text{eat}|I) * P(\text{Chinese}|\text{eat}) * P(\text{food}|\text{Chinese})$
= $0.0036 * 0.021 * 0.52$



Answer 3

In practice, we should consider $\langle s \rangle$ and $\langle /s \rangle$:

- $P(I \text{ eat Chinese food})$
 $= P(I | \langle s \rangle) * P(\text{eat} | I) * P(\text{Chinese} | I \text{ eat}) * P(\text{food} | I \text{ eat Chinese})$
 $* P(\langle /s \rangle | I \text{ eat Chinese food})$
- $P(\langle s \rangle I \text{ eat Chinese food} \langle /s \rangle)$
 $= P(I | \langle s \rangle) * P(\text{eat} | I) * P(\text{Chinese} | \text{eat}) * P(\text{food} | \text{Chinese}) * P(\langle /s \rangle | \text{food})$
 $= ??? * 0.0036 * 0.021 * 0.52 * ???$

??? → unknown probabilities from the question.

Question 4

➤ Why do we need to do smoothing for language model?

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n) + 1}{C(w_{n-1}) + V}$$

Answer 4

- Our maximum likelihood estimation is based on training data
- Text data are 'sparse' for the estimation
 - for n-grams that occur a sufficient number of times, it is fine
 - some perfectly acceptable English sequences will be missing from the training corpus
 - 0 probability problem
 - estimate is poor when the counts are small
- e.g., Laplace smoothing and other more advanced smoothing



Question 5

- Given some text, what are the general steps to collect all counts needed for building an n -gram language model?



Answer 5 (The Big Picture)

➤ Training phase.

- Reset all n-gram counts to 0.
- For each sentence in the training data:
 - Update n-gram counts (A).

➤ Evaluation phase.

- For each sentence to be evaluated:
 - For each n-gram in the sentence:
 - Call smoothing routine to evaluate probability of n-gram given training counts (B).
- Compute overall perplexity of evaluation data from n-gram probabilities.

Resources

- Lucene http://lucene.apache.org/core/7_4_0/index.html
- OpenNLP <https://opennlp.apache.org/>
- Stanford NLP <https://nlp.stanford.edu/>
- spaCy <https://spacy.io/>
- NLTK <https://www.nltk.org/>

