CZ/CE4045 Natural Language Processing

Tutorial 3: N-gram and Language Model

Question 1

- 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

Review

Bigram Counts

- Out of 9222 sentences
  - Eg. “I want” occurred 827 times

	i	want	to	eat	chinese	food	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

Review

Laplace-Smoothed Bigram Probabilities

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

	i	want	to	eat	chinese	food	lunch	spend
i	0.0015	0.21	0.00025	0.0025	0.00025	0.00025	0.00025	0.00075
want	0.0013	0.00042	0.26	0.00084	0.0029	0.0029	0.0025	0.00084
to	0.00078	0.00026	0.0013	0.18	0.00078	0.00026	0.0018	0.055
eat	0.00046	0.00046	0.0014	0.00046	0.0078	0.0014	0.02	0.00046
chinese	0.0012	0.00062	0.00062	0.00062	0.00062	0.052	0.0012	0.00062
food	0.0063	0.00039	0.0063	0.00039	0.00079	0.002	0.00039	0.00039
lunch	0.0017	0.00056	0.00056	0.00056	0.00056	0.0011	0.00056	0.00056
spend	0.0012	0.00058	0.0012	0.00058	0.00058	0.00058	0.00058	0.00058

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

	very	good	tennis	player	in	us	open	qualify	play
very	0	1	0	0	0	0	0	0	0
good	0	0	1	0	0	0	0	0	0
tennis	0	0	0	3	0	0	0	0	0
player	0	0	0	0	1	1	0	1	0
in	0	0	0	0	0	1	0	0	0
us	0	0	0	0	0	0	3	0	0
open	0	0	0	0	0	0	0	0	0
qualify	0	0	0	0	0	0	0	0	1
play	0	0	0	0	0	1	0	0	0

Answer Q1. (ii): Compute the bigram probabilities using Laplace smoothing

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

- Unigram counting
  - *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
very	0.1	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.1
good	0.1	0.1	0.2	0.1	0.1	0.1	0.1	0.1	0.1
tennis	1/12	1/12	1/12	4/12	1/12	1/12	1/12	1/12	1/12
player	1/12	1/12	1/12	1/12	2/12	2/12	1/12	2/12	1/12
in	0.1	0.1	0.1	0.1	0.1	0.2	0.1	0.1	0.1
US	1/12	1/12	1/12	1/12	1/12	1/12	4/12	1/12	1/12
open	1/12	1/12	1/12	1/12	1/12	1/12	1/12	1/12	1/12
qualify	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.2
play	0.1	0.1	0.1	0.1	0.1	0.2	0.1	0.1	0.1

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

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

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

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

Question 3

- Given the bigram probability in the following table, compute the probability of “I eat Chinese food” by using the table.

	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

- $P(I\ eat\ Chinese\ food)$

$$= \ P(eat|I) * \ P(Chinese|I\ eat) * \ \ P(food|I\ eat\ Chinese)$$

- Chain rules
  - Independence Assumption – bigram

- $P(I\ eat\ Chinese\ food)$

$$= \ P(eat|I) * \ P(Chinese|eat) * \ \ P(food|Chinese)$$
$$= \ 0.0036 \ * \ 0.021 \ * \ 0.52$$

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

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/