Lecture Note - 08: NLP, Word Representation, Language Model, Bigram, MLE

Dihui Lai

March 17, 2020

Contents

1 Word Semantics and Vector Representations

- Homonymous: a word can have multiple definitions e.g. mouse could mean small rodents or it could mean computer devices.
- Synonyms/antonym (words' relations): couch/sofa, vomit/throw up, filbert/hazelnut; long/short, big/little
- Word sentiments
- Can we represent a word using vectors and quantify those measures?

Term-term matrix or word-word matrix: count the number of times a word occurs in a context window around the target word (e.g. ± 7)

sugar, a sliced lemon, a tablespoonful of, **apricot** jam, a pinch each of,

	aardvark	...	computer	data	pinch	result	sugar	...
apricot	0	...	0	0	1	0	1	...
pineapple	0	...	0	0	1	0	1	...
digital	0	...	2	1	0	1	0	...
information	0	...	1	6	0	4	0	...

It can be inferred from the word-word matrxi that apricot and pineapple are more simliar to each other.

2 Cosine Similarity

The similarity of two words could be measured by dot-products of their vector representation

$$\vec{v} \cdot \vec{w} = \sum_{i=1}^{N} v_i w_i$$

The dot-product favors vectors of higher frequency to normalize the similarity without considering word frequency, we use cosine similarity meature

$$cosine(\vec{v}, \vec{w}) = \frac{\vec{v} \cdot \vec{w}}{|\vec{v}||\vec{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_1^N v_i^2}\sqrt{\sum_1^N w_i^2}}$$

3 Language Model

3.1 N-gram Language Models

- Models that assign probabilities to sequences of words are called language models or LM.
- An n-gram is a sequence of N words e.g. 2-gram (or bigram) "Good Morning", 3-gram "Turn it on"
- N-gram lanuage models estimate the probability of the last word of an n-gram given the previous words

LM: What is the probability of having a sentence that consists a sequence of words: w_1, w_2, w_3 ... w_N, i.e. $P(w_1, w_2, w_3 ... w_N)$.

Recall the chain rule:

$$
\begin{aligned}
&P(w_1, w_2, w_3 ... w_N) \\
&= P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)P(w_4|w_1, w_2, w_3)...P(w_N|w_1, w_2, ...w_{N-1})
\end{aligned}
$$

In the case of bigram, we assume $P(w_N|w_1, ..., w_{N-1}) = P(w_N|w_{N-1})$, since the word is only dependent on the previous word, it is also called Markov assumption. In general case of an n-gram, we assume $P(w_N|w_1, w_2, ...w_{N-1}) = P(w_N|w_{N-1}, w_{N-2}, ...w_{N-n+1})$

3.2 MLE Estimation for bigram

In the case of bigram, the MLE estimation can be formulated as

$$
P(w_N|w_{N-1}) = \frac{C(w_{N-1}w_N)}{\sum_w C(w_{N-1}w)} = \frac{C(w_{N-1}w_N)}{C(w_{N-1})}
$$

Here, C is the count of the words' occurence

3.3 Example: MLE Estimation for bigram

Estimate the bigram for the following corpus, here $\langle s \rangle$ and $\langle /s \rangle$ are introduced as the symbols that represents the begining and end of a setence.

$\langle s \rangle$ I am Sam $\langle /s \rangle$
$\langle s \rangle$ Sam I am $\langle /s \rangle$
$\langle s \rangle$ I do not like green eggs and ham $\langle /s \rangle$

We begin buy counting the words occurence and have $C(I) = 3$, $C(Sam) = 2$, $C(\langle /s \rangle) = 3$, $C(\langle s \rangle) = 3$ ...$C(\langle s \rangle I) = 2$, $C(\langle s \rangle Sam) = 1$

So we have $P(I|\langle s \rangle) = \frac{2}{3}$, $P(Sam|\langle s \rangle) = \frac{1}{3}$, $P(do|I) = \frac{1}{3}$, $P(am|I) = \frac{2}{3}$, $P(Sam|am) = \frac{1}{2}$, $P(\langle /s \rangle|Sam) = \frac{1}{2}$

The in-sample probability of $P(\langle s \rangle I\ am\ Sam \langle /s \rangle) = P(I|\langle s \rangle)P(am|I)P(Sam|am)P(\langle /s \rangle|Sam) = 2/3x2/3x1/2x1/2$

How do we compare two LM?

- A test data/hold out data set can be used to evaluate a LM. Apply the estiamated conditional probability to the test data set and compare the resulting probability.

- Perplexity is used instead of the raw probability.

$$
\begin{aligned}
PP(W) &= P(w_1, w_2, ...w_N)^{-\frac{1}{N}} \\
&= \sqrt[N]{\frac{1}{P(w_1, w_2, ...w_N)}}
\end{aligned}
$$

- Maximize probability is equivalent to minimize perplexity