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

Dihui Lai

March 20, 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?

## 1.1 Term-term matrix/Word-word matrix

Count the number of times ($n$) a word occurs in a context window ($w$) around the target word ($t$). Let's consider the following setence as an example:

*Data scientists are big data wranglers, gathering and analyzing large sets of structured and unstructured data*

In the setence, the words 'are', 'and', 'of' are stop words and serve as building blocks to form a setence. While constructing a word representation, let us ignore them for the moment and consider the words in their base format. Thus we end up with a sentence as of the following

*data scientist big data wrangler gather analyze large set structure unstructure data*

The unique words appeared in the sentence form a dictionary: {*data, scientist big wrangler, gather, analyze, large, set, structure, unstructure*}.

As a first step to construct a term-term matrix, we use the words from the dictionary as columns and the each word in the setence as rows. For simplicity, we consider the terms appear in a context - window of size 2, i.e. $w = \pm 1$. Check the first word in the setence *data*, the words appear within the context-window are *scientist* and *big*. We then fill the corresponding cells $M_{12} = 1$, $M_{13} = 1$ in the term-term matrix. Similarly, the second word *scientist*, has non-zero cell in the matrix $M_{21} = 1$ and $M_{23} = 1$. We repeat this practice and get the term-term matrix below

| | data | scientist | big | wrangler | gather | analyze | large | set | structure | unstructure |
|---|---|---|---|---|---|---|---|---|---|---|
| data | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| scientist | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| big | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| data | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| wrangler | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| gather | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| analyze | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| large | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| set | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| structured | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| unstructured | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| data | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

To construct the term-term matrix we aggregate the rows of the above matrix according to the row keys

Table 1: Term-term matrix: the cells in the matrix indicate the word count, appearing in a context window of size 2

| | data | scientist | big | wrangler | gather | analyze | large | set | structure | unstructure |
|---|---|---|---|---|---|---|---|---|---|---|
| data | 0 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| scientist | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| big | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| wrangler | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| gather | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| analyze | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| large | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| set | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| structured | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| unstructured | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

### 1.2 Word Representation using Neural Network

## 2 Cosine Similarity

By looking at the Term-term matrix in Table. 1, we can see that data and scientist seems to have a common context word *big* and could be close in their meanings. How can we quantify this? One possibility is using the dot-product of their vector representation.

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

However, the dot-product favors vectors of higher frequency. Words that appears often are likely to have higher dot-product value than word of low occurence. To normalize the frequency, we can use cosine similarity meature, which is defined as below

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

## 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