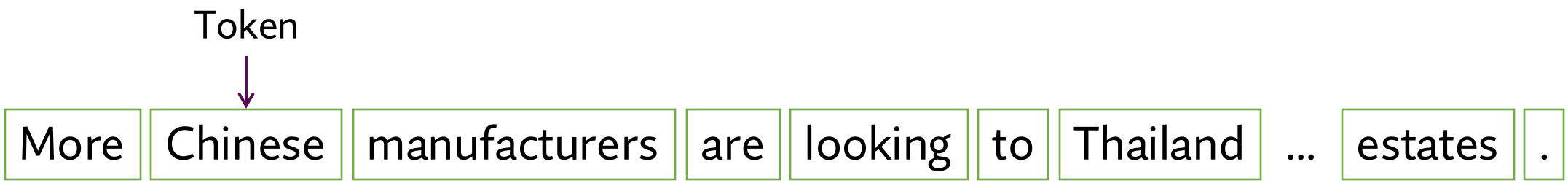
Feature Engineering

Papangkorn Inkeaw, Ph.D.

Feature Extraction

Chapter 5 (Part II) - Feature Extraction for Text Data

Text – Basic Knowledge



Text is a human-readable sequence of word(s).

Text – Basic Knowledge

n-gram A contiguous sequence of *n* tokens from a given sample of text.

More | Chinese | manufacturers | are | looking | to | Thailand | ... | estates | .

Unigram More | Chinese | manufacturers | are | looking | to | Thailand | ...

Bigram More Chinese | Chinese manufacturers | manufacturers are | are looking | looking to | to Thailand | ...

Trigram More Chinese manufacturers | Chinese manufacturers are | manufacturers are looking | are looking to | looking to Thailand | ...

Text – Basic Knowledge

n-gram model predicts the occurrence of a word based on the occurrence of its N – 1 previous words.

Unigram

Current

A word is independent from its previous words.

bigram

Current

The occurrence of a word depends on the occurrence of it's a previous words.

trigram

Current

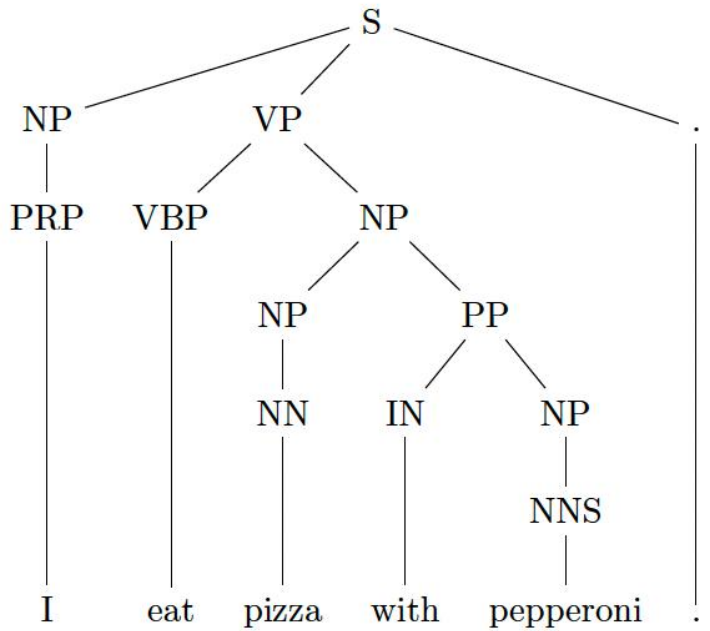
The occurrence of a word depends on the occurrence of it's two previous words.

Text – Basic Knowledge

Part-of-speech (POS) The grammatical function of each word in a sentence

We can use NLP tool to to categorizing words in a text in correspondence with a particular part of speech, depending on the definition of the word and its context.

I	eat	pizza	with	pepperoni	.
PRP	VBP	NNS	IN	NNS	.



From Text to Feature Vector

General Feature Extraction Process

- POS tagging
- Stemming
- Lemmatization
- Name-entity recognition

- Remove stop words
- Remove irrelevant words
- Create dictionary

One-hot Encoding

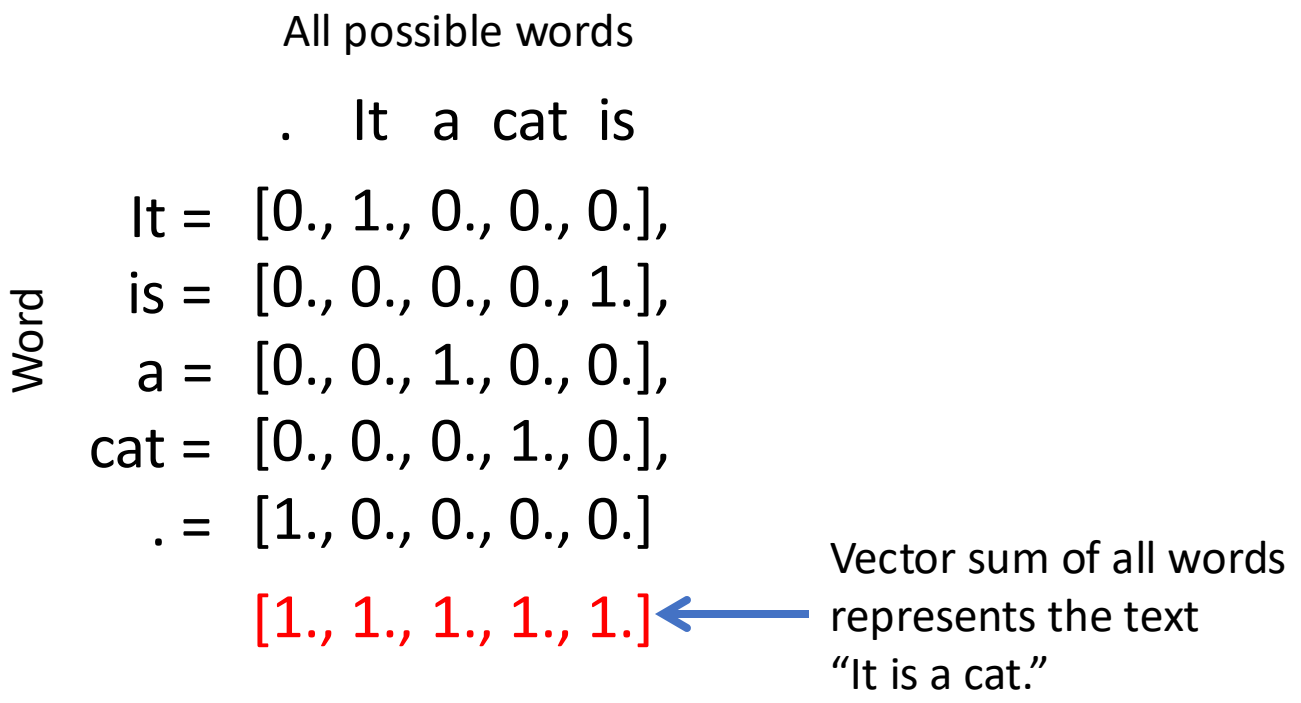
- A representation of categorical variables as binary vectors.
- Each word is represented as a binary vector that is:
  - All zero values
  - Except the index of the word, which is marked with a 1.

All possible words

Word	.	It	a	cat	is
It =	[0.,	1.,	0.,	0.,	0.],
is =	[0.,	0.,	0.,	0.,	1.],
a =	[0.,	0.,	1.,	0.,	0.],
cat =	[0.,	0.,	0.,	1.,	0.],
. =	[1.,	0.,	0.,	0.,	0.]

Bag of Words

- What about full texts instead of single words?
- The vector representation of a text is simply the vector sum of all the words it contains:



Bag of Words

- In practice, it's much more convenient to use a dictionary instead of an actual vector.
- This is known as a **bag-of-words**, and word order is discarded.

Dictionary

Word	cat	dog	bird	panda
They =	[0.,	0.,	0.,	0.],
are =	[0.,	0.,	0.,	0.],
cat =	[1.,	0.,	0.,	0.],
and =	[0.,	0.,	0.,	0.],
dog =	[0.,	1.,	0.,	0.]
They are cat and dog =	[1.,	1.,	0.,	0.]

Bag of words represents the text "They are cat and dog"

TF-IDF

- The **term frequency** *tf(t,d)* of term *t* in document *d* is defined as the relative frequency of times that *t* occurs in *d*.

$$\mathrm{tf}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

- Logarithmically scaled frequency is defined as $\mathrm{tf}(t,d) = \log\left(1 + f_{t,d}\right)$
- Document frequency: Rare terms are more informative than frequent terms
- The **inverse document frequency** is a measure of how much information the word provides.
- The idf(t,D) is the ratio of the total number of documents D to the number of documents containing the term t.

$$\mathrm{idf}(t,D) = \frac{N}{n_t}$$

TF-IDF

- Then, the Term frequency–inverse document frequency tfidf(t,d,D) is calculated as

$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \times \mathrm{idf}(t, D)$$

- A high weight in tf–idf is reached by a high term frequency (in the given document).
- A low document frequency of the term in the whole collection of documents; the weights hence tend to filter out common terms.

Word Embeddings

“ Words with similar meanings should occur in similar contexts. ”

- The distributional hypothesis in linguistics

- From a word we can get some idea about the context where it might appear.

__ bird __ __

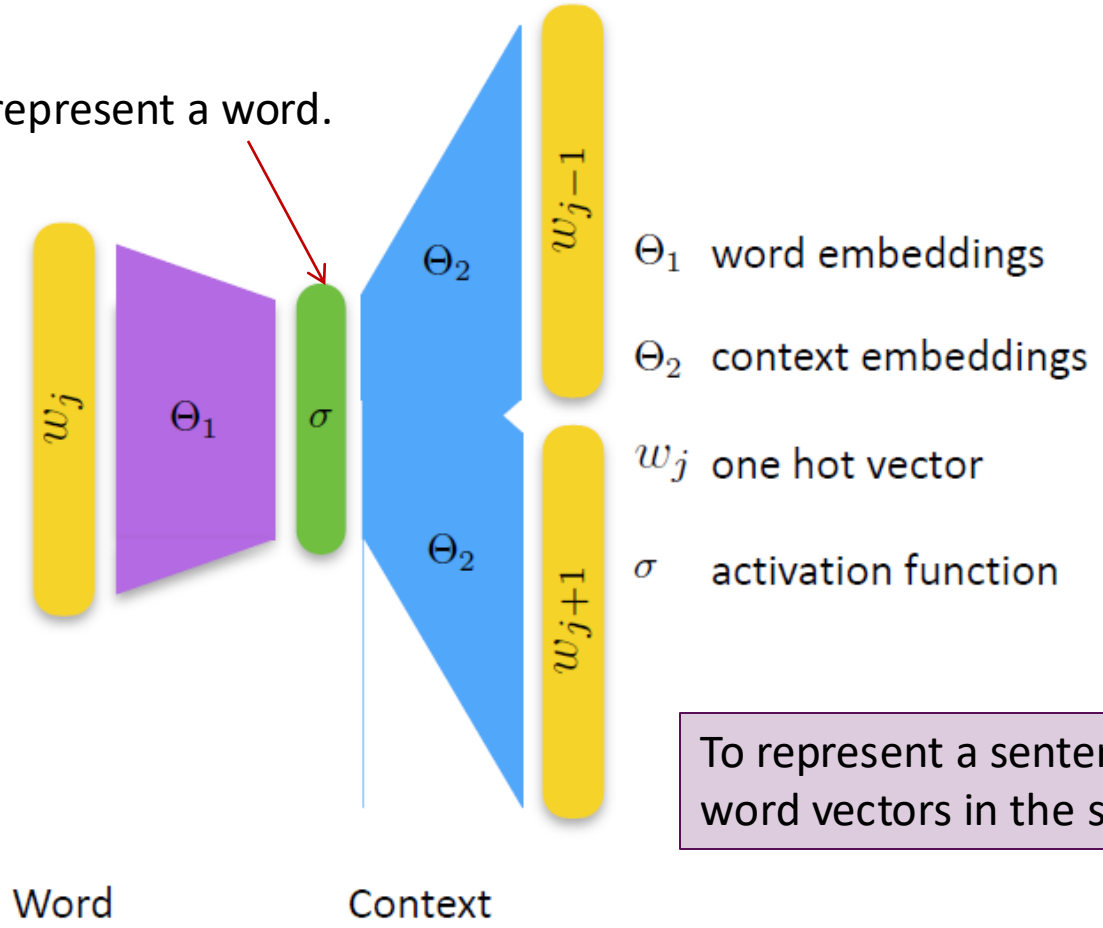
- From the context, we have some idea about possible words.

The red __ is nice

Word Embeddings

Skipgram

Use the output of this layer to represent a word.



Θ_1 word embeddings

Θ_2 context embeddings

w_j one hot vector

σ activation function

To represent a sentence, we can average all word vectors in the sentence.

Source: https://data4sci.com/nlp

Word Embeddings

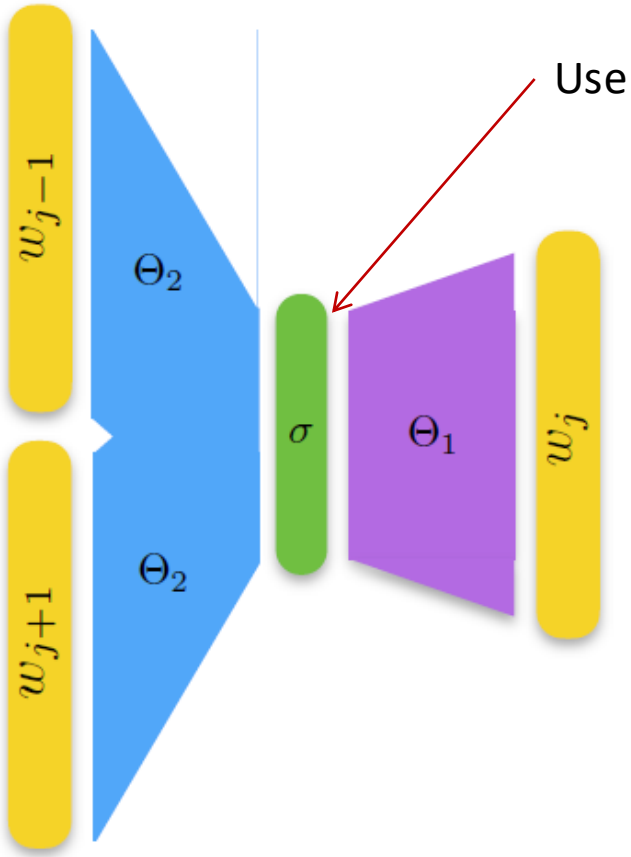
Continuous Bag of Words

Θ_1 word embeddings

Θ_2 context embeddings

w_j one hot vector

σ activation function



Use the output of this layer to represent a word.

To represent a sentence, we can average all word vectors in the sentence.

Source: https://data4sci.com/nlp

References & Study Resources

- Guozhu Dong and Huan Liu. (2020). *Feature Engineering for Machine Learning and Data Analytics*. CRC Press.
- https://data4sci.com/nlp