What's the Word?

Word Representations

Sep 25th & 28th, 2017

ADL x MLDS

YUN-NUNG (VIVIAN) CHEN

HTTP://ADL.MIULAB.TW

HTTP://MLDS.MIULAB.TW

國立臺灣大學

National Taiwan University

Slides credited from Dr. Richard Socher

Learning Target Function

Classification Task

$$f(x) = y \quad \Rightarrow \quad f : R^N \rightarrow R^M$$

- x: input object to be classified → a N-dim vector
- y: class/label → a M-dim vector

Assume both x and y can be represented as fixed-size vectors

"這規格有誠意!" ⟶ +

"太爛了吧~" ⟶ -

How do we represent the meaning of the word?

Meaning Representations

Definition of “Meaning”

- the idea that is represented by a word, phrase, etc.
- the idea that a person wants to express by using words, signs, etc.
- the idea that is expressed in a work of writing, art, etc.

Goal: word representations that capture the relationships between words

Meaning Representations in Computers

Knowledge-based representation

Corpus-based representation

✓ Atomic symbol

✓ Neighbors

- High-dimensional sparse word vector
- Low-dimensional dense word vector
  - Method 1 – dimension reduction
  - Method 2 – direct learning

Meaning Representations in Computers

Knowledge-based representation

Corpus-based representation

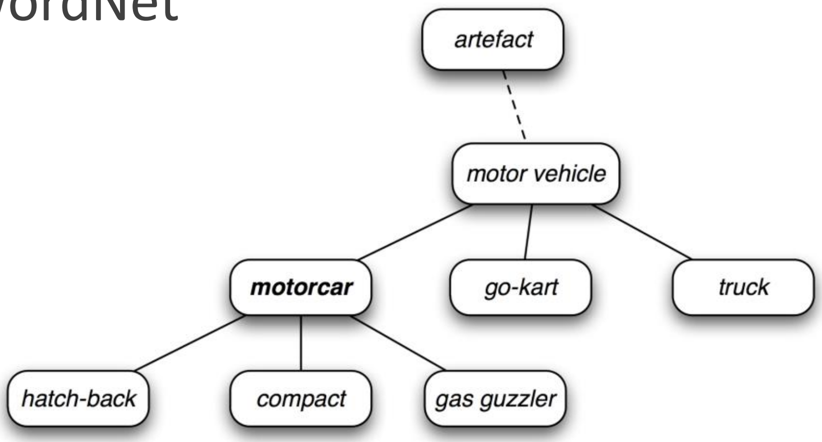
- ✓ Atomic symbol
- ✓ Neighbors
  - High-dimensional sparse word vector
  - Low-dimensional dense word vector
    - Method 1 – dimension reduction
    - Method 2 – direct learning

Knowledge-based representation

Hypernyms (is-a) relationships of WordNet

```
from nltk.corpus import wordnet as wn
panda = wn.synset('panda.n.01')
hyper = lambda s: s.hypernyms()
list(panda.closure(hyper))
```

```
[Synset('procyonid.n.01'),
Synset('carnivore.n.01'),
Synset('placental.n.01'),
Synset('mammal.n.01'),
Synset('vertebrate.n.01'),
Synset('chordate.n.01'),
Synset('animal.n.01'),
Synset('organism.n.01'),
Synset('living_thing.n.01'),
Synset('whole.n.02'),
Synset('object.n.01'),
Synset('physical_entity.n.01'),
Synset('entity.n.01')]
```



Issues:

- newly-invented words
- subjective
- annotation effort
- difficult to compute word similarity

Meaning Representations in Computers

Knowledge-based representation

Corpus-based representation

- ✓ Atomic symbol
- ✓ Neighbors
  - High-dimensional sparse word vector
  - Low-dimensional dense word vector
    - Method 1 – dimension reduction
    - Method 2 – direct learning

Corpus-based representation

Atomic symbols: ***one-hot*** representation

car $[0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ \ldots\ 0]$

car

Issues: difficult to compute the similarity
(i.e. comparing "car" and "motorcycle")

$\underbrace{[0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ \ldots\ 0]}_{\text{car}}$ AND $\underbrace{[0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ \ldots\ 0]}_{\text{motorcycle}} = 0$

Idea: words with similar meanings often have similar neighbors

Corpus-based representation

Co-occurrence matrix

- Neighbor definition: full document v.s. windows

full document
word-document co-occurrence matrix gives general topics
→ "Latent Semantic Analysis"

windows
context window for each word
→ capture syntactic (e.g. POS) and sematic information

Meaning Representations in Computers

Knowledge-based representation

Corpus-based representation

- ✓ Atomic symbol
- ✓ Neighbors
  - High-dimensional sparse word vector
  - Low-dimensional dense word vector
    - Method 1 – dimension reduction
    - Method 2 – direct learning

Window-based Co-occurrence Matrix

Example

- Window length=1
- Left or right context
- Corpus:

I love NTU.
I love deep learning.
I enjoy learning.

similarity > 0

Counts	I	love	enjoy	NTU	deep	learning
I	0	2	1	0	0	0
love	2	0	0	1	1	0
enjoy	1	0	0	0	0	1
NTU	0	1	0	0	0	0
deep	0	1	0	0	0	1
learning	0	0	1	0	1	0

Issues:

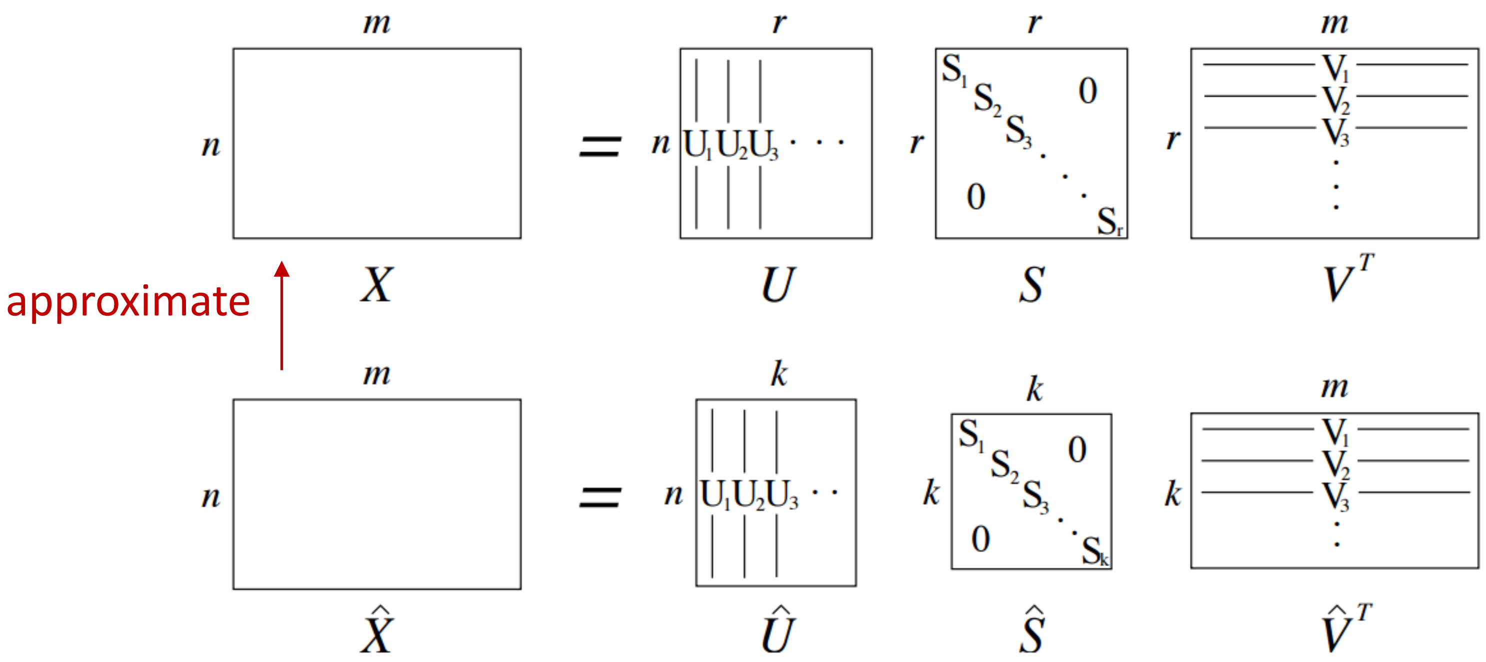
- matrix size increases with vocabulary
- high dimensional
- sparsity → poor robustness

Idea: low dimensional word vector

Low-Dimensional Dense Word Vector

Method 1: dimension reduction on the matrix

Singular Value Decomposition (SVD) of co-occurrence matrix X



Meaning Representations in Computers

Knowledge-based representation

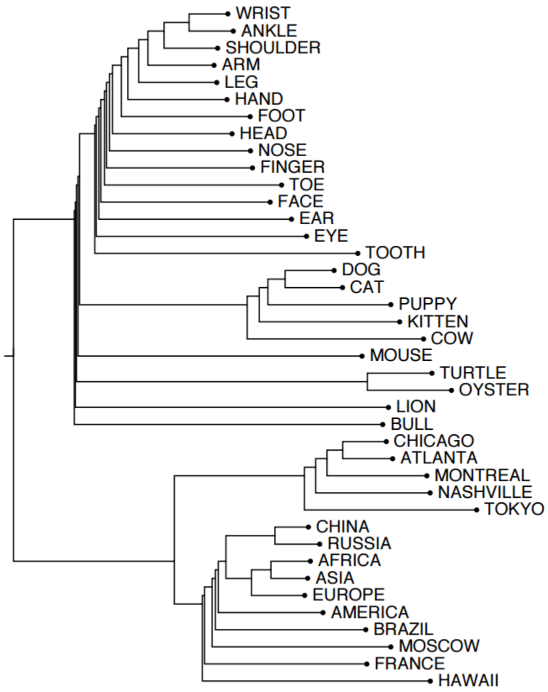
Corpus-based representation

- ✓ Atomic symbol
- ✓ Neighbors
  - High-dimensional sparse word vector
  - Low-dimensional dense word vector
    - Method 1 – dimension reduction
    - Method 2 – direct learning

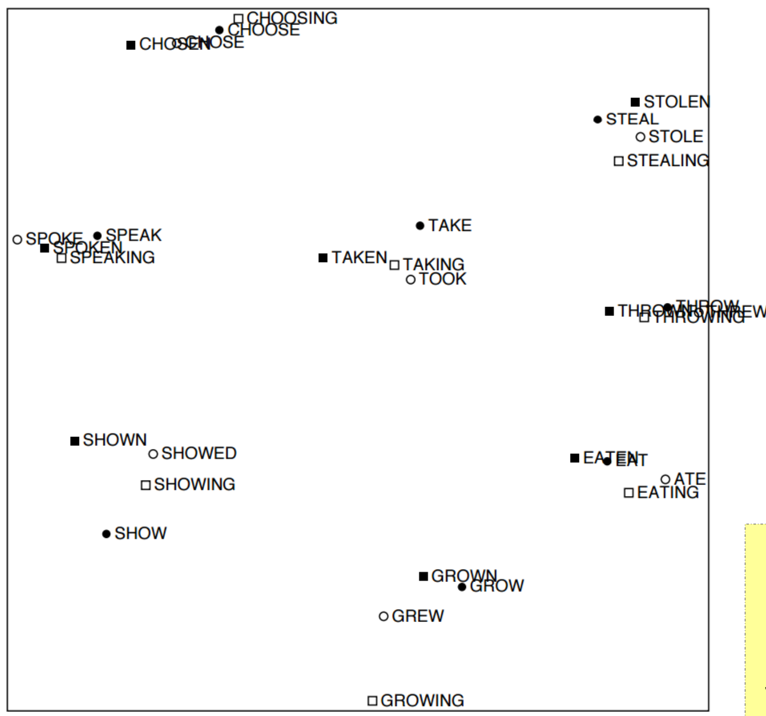
Low-Dimensional Dense Word Vector

Method 1: dimension reduction on the matrix

Singular Value Decomposition (SVD) of co-occurrence matrix X



semantic relations



syntactic relations

Issues:

- computationally expensive: $O(mn^2)$ when $n<m$ for $n \times m$ matrix
- difficult to add new words

Idea: directly learn low-dimensional word vectors

Rohde et al., "An Improved Model of Semantic Similarity Based on Lexical Co-Occurrence," 2005.

Meaning Representations in Computers

Knowledge-based representation

Corpus-based representation

- ✓ Atomic symbol
- ✓ Neighbors
  - High-dimensional sparse word vector
  - Low-dimensional dense word vector
    - Method 1 – dimension reduction
    - Method 2 – direct learning

Low-Dimensional Dense Word Vector

Method 2: directly learn low-dimensional word vectors

- Learning representations by back-propagation. (Rumelhart et al., 1986)
- A neural probabilistic language model (Bengio et al., 2003)
- NLP (almost) from Scratch (Collobert & Weston, 2008)
- Recent and most popular models: word2vec (Mikolov et al. 2013) and Glove (Pennington et al., 2014)
  - To be introduced in detail by the lecture “Word Embeddings”

Word2Vec

Idea: predict surrounding words of each word

Benefit: faster, easily incorporate a new sentence/document or add a word to vocab

Goal: predict surrounding words within a window of each word

Objective function: maximize the log probability of any context word given the current center word

$$w_1, w_2, \cdots, \underbrace{w_{t-m}, \cdots, w_{t-1}, w_t, w_{t+1}, \cdots, w_{t+m}}_{\text{context window (size=m)}}, \cdots, w_{T-1}, w_T$$

$$C(\theta) = \frac{1}{T} \sum_{t=1}^{T} \sum_{-m \leq j \leq m, j \neq 0} \log p(w_{t+j} \mid w_t)$$

Word2Vec

Goal: predict surrounding words within a window of each word

Objective function: maximize the log probability of any context word given the current center word

$$w_1, w_2, \cdots, \underbrace{w_{t-m}, \cdots, w_{t-1}, w_t, w_{t+1}, \cdots, w_{t+m}}_{\text{context window (size=m)}}, \cdots, w_{T-1}, w_T$$

$$O(\theta) = \frac{1}{T} \sum_{t=1}^{T} \sum_{-m \leq j \leq m, j \neq 0} \log p(w_{t+j} \mid w_t)$$

target word vector

$$p(o \mid c) = \frac{\exp(u_o^T v_c)}{\sum_w \exp(u_w^T v_c)}$$

outside center

u: outside word vector

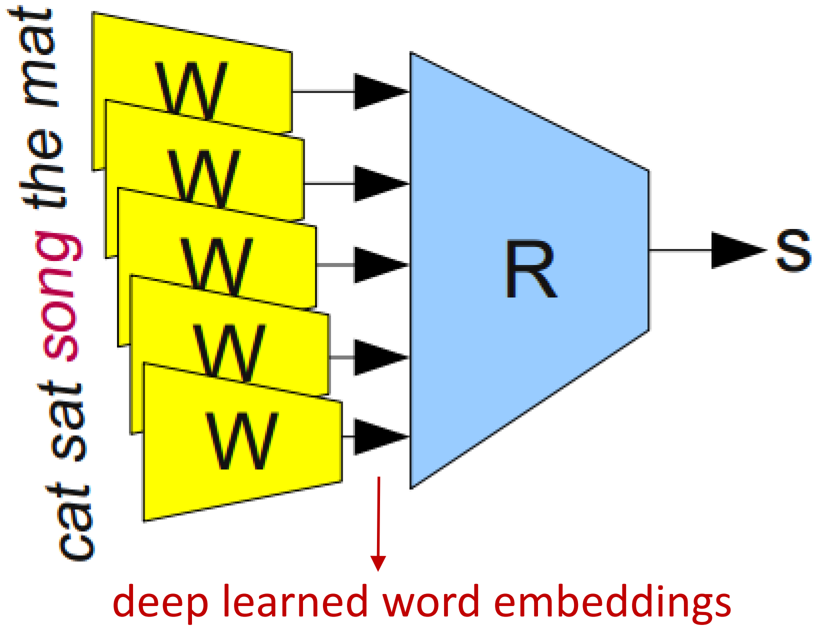
v: center word vector

representation learning via deep learning → called "word embeddings"

Major Advantages of Word Embeddings

Propagate ***any*** information into them via neural networks

- form the basis for all language-related tasks



The networks, R and Ws, can be updated during model training

Concluding Remarks

Knowledge-based representation

Corpus-based representation

✓Atomic symbol

✓Neighbors

- High-dimensional sparse word vector
- Low-dimensional dense word vector
  - Method 1 – dimension reduction
  - Method 2 – direct learning