Applied Deep Learning

Word Representations

September 21st, 2023 http://adl.miulab.tw

National Taiwan University
國立臺灣大學

Updated Schedule

Virtual | Physical | No Course

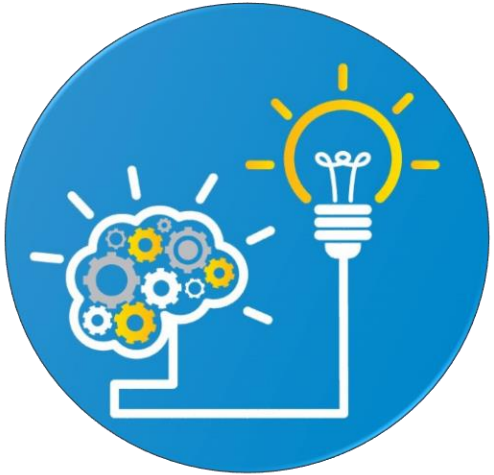
Week	Topic	TA Recitation	Assignment
1 2023/09/07	Course Logistics, Introduction	Colab, GPU, PyTorch	
2 2023/09/14	NN Basics, Backpropagation		
3 2023/09/21	Word Representations, Sequence Modeling	DL Workflow	
4 2023/09/28	Teachers' Day Break		A1 – BERT
5 2023/10/05	Attention, Transformer	HuggingFace Tutorial	
6 2023/10/12	Tokenization, Model Pre-Training (BERT, GPT)		
7 2023/10/19	Sequence Generation for Diverse Tasks	LLM Architecture	A2 – NLG
8 2023/10/26	Midterm Break		
9 2023/11/02	Natural Language Generation, Perplexity		
10 2023/11/09	Prompt-Based Learning	LLM Eval	A3 – LLM Tuning
11 2023/11/16	Adaptation	LLM Training	
12 2023/11/23	Conversational AI	LLM Inference	
13 2023/11/30	Beyond Supervised Learning		
14 2023/12/07	Break		
15 2023/12/14	Invited Talk		
16 2023/12/21	Final Project Presentation		

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.

Meaning Representations in Computers

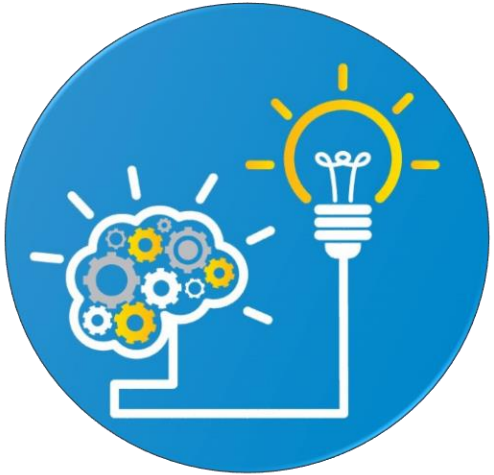
Knowledge-Based Representation



Corpus-Based Representation

Meaning Representations in Computers

Knowledge-Based Representation



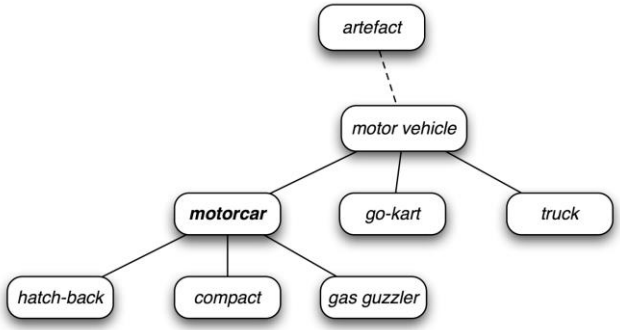
Corpus-Based Representation

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

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”)

$[0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ \ldots\ 0]$ AND $[0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ \ldots\ 0] = 0$

car motorcycle

Idea: words with similar meanings often have similar neighbors

Corpus-Based Representation

- Neighbor-based representation
  - Co-occurrence matrix constructed via neighbors
  - Neighbor definition: full document vs. 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 semantic information

Window-Based Co-occurrence Matrix

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

I love AI.
I love deep learning.
I enjoy learning.

similarity > 0

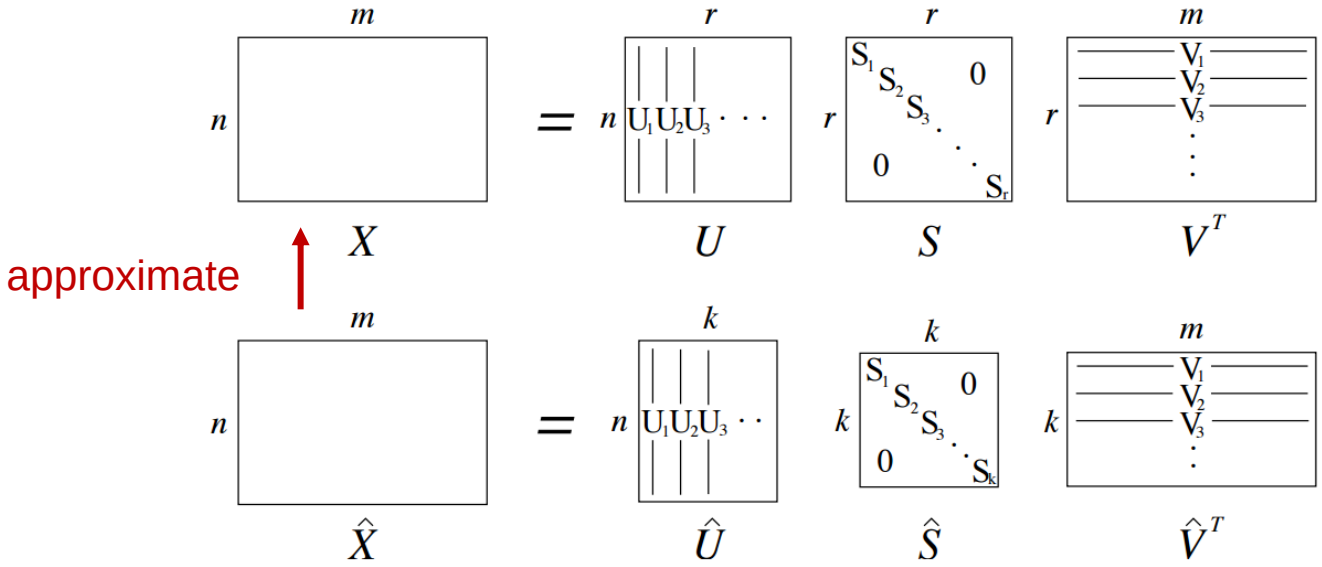
Counts	I	love	enjoy	AI	deep	learning
I	0	2	1	0	0	0
love	2	0	0	1	1	0
enjoy	1	0	0	0	0	1
AI	0	1	0	0	0	0
deep	0	1	0	0	0	1
learning	0	0	1	0	1	0

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

Idea: low dimensional word vector

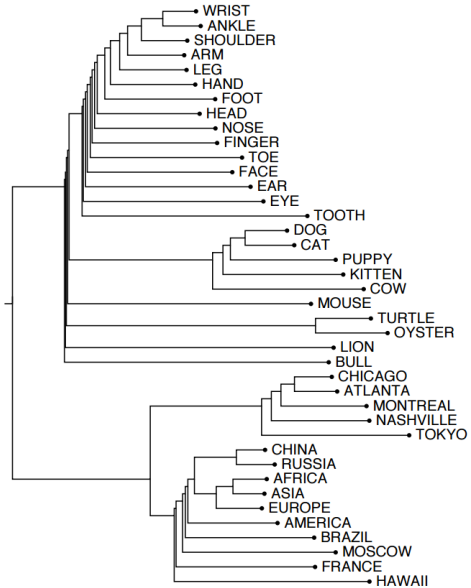
Low-Dimensional Dense Word Vector

- Method 1: dimension reduction on the matrix
- Singular Value Decomposition (SVD) of co-occurrence matrix X

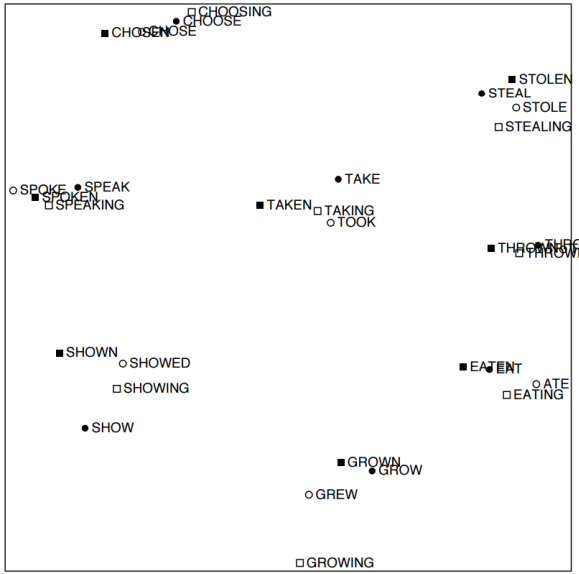


Low-Dimensional Dense Word Vector

- 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 nxm 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.

Low-Dimensional Dense Word Vector

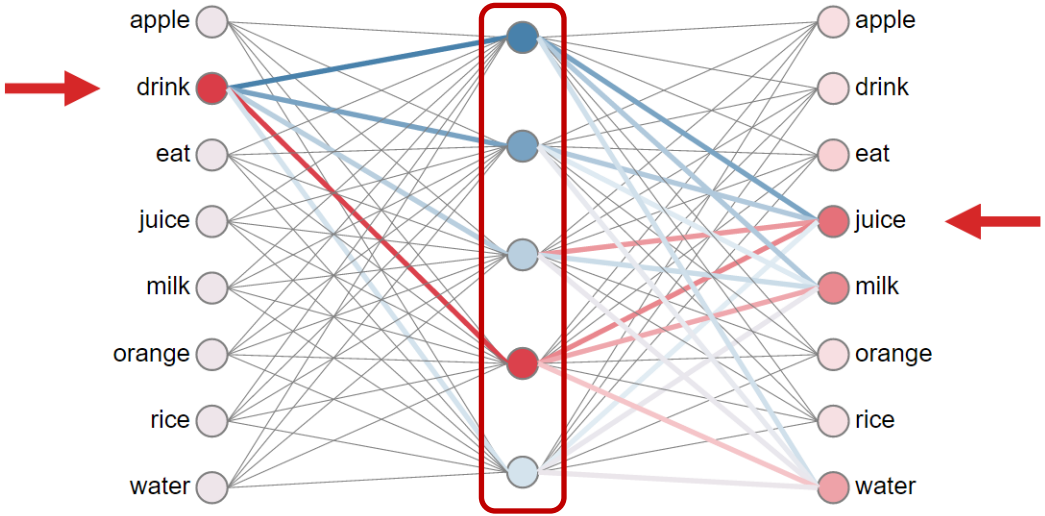
- Method 2: directly learn low-dimensional word vectors
  - Learning representations by back-propagation. (Rumelhart et al., 1986)
  - A neural probabilistic language model (Bengio et al., 2003)
  - NLP (almost) from Scratch (Collobert & Weston, 2008)
  - Recent and most popular models: word2vec (Mikolov et al. 2013) and Glove (Pennington et al., 2014)
    - As known as “Word Embeddings”

Word Embeddings
Word2Vec

Word2Vec Idea from Language Modeling

- LM objective: predicting the next words given the proceeding contexts

$$p(w_{t+1} \mid w_t)$$



Finding: similar words have similar hidden representations

T. Mikolov et al., “Linguistic Regularities in Continuous Space Word Representations,” in *Proceedings of NAACL-HLT*, 2013.

Word2Vec Variants

- **Skip-gram**: predicting surrounding words given the target word (Mikolov+, 2013)

$$p(w_{t-m}, \cdots w_{t-1}, w_{t+1}, \cdots, w_{t+m} \mid w_t)$$

better

- **CBOW (continuous bag-of-words)**: predicting the target word given the surrounding words (Mikolov+, 2013)

$$p(w_t \mid w_{t-m}, \cdots w_{t-1}, w_{t+1}, \cdots, w_{t+m})$$

- **LM (Language modeling)**: predicting the next words given the proceeding contexts (Mikolov+, 2013)

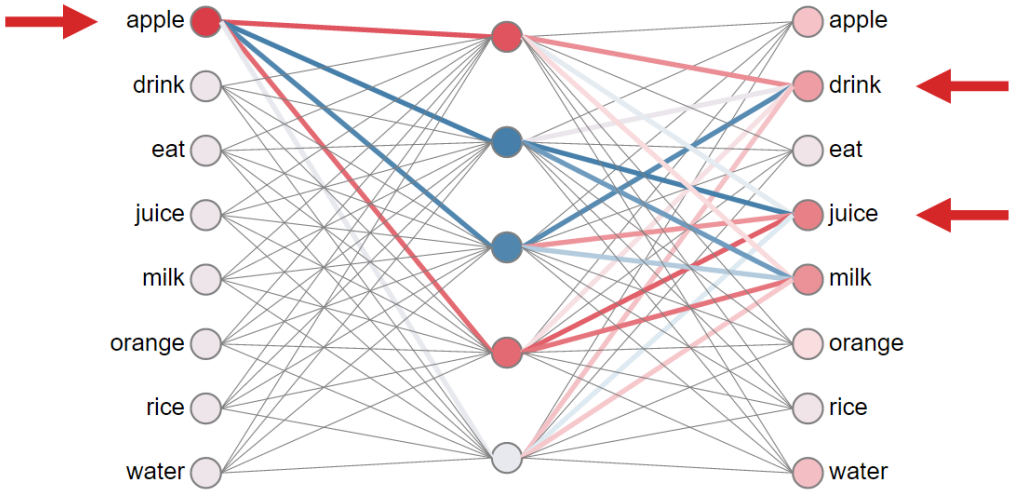
$$p(w_{t+1} \mid w_t)$$

first

Mikolov et al., “Efficient estimation of word representations in vector space,” in *ICLR Workshop*, 2013.
Mikolov et al., “Linguistic regularities in continuous space word representations,” in *NAACL HLT*, 2013.

Word2Vec Skip-Gram Visualization https://ronxin.github.io/wevi/

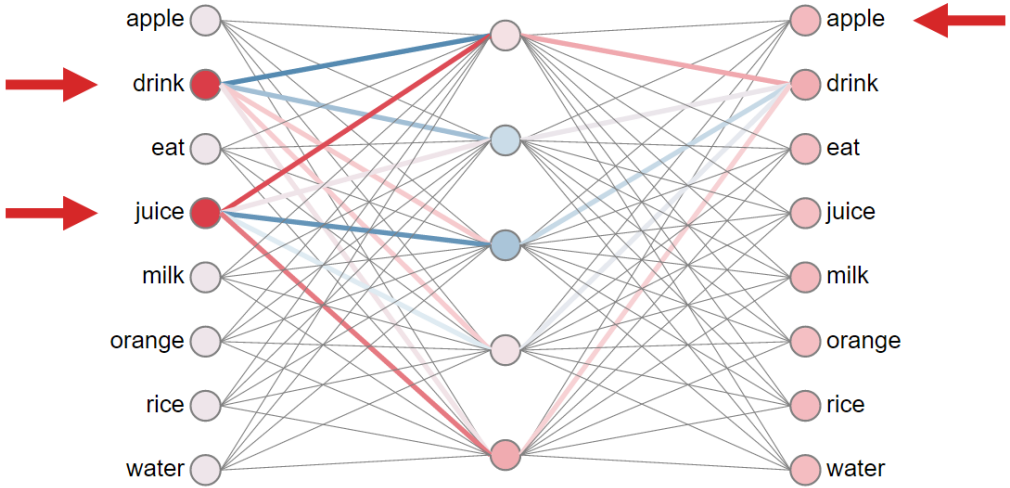
- Skip-gram training data:
  apple|drink^juice,orange|eat^apple,rice|drink^juice,juice|drink^milk,milk|drink^rice,water|drink^milk,juice|orange^apple,juice|apple^drink,milk|rice^drink,drink|milk^water,drink|water^juice,drink|juice^water



Word2Vec CBOW

- Goal: predicting the target word given the surrounding words

$$p(w_t \mid w_{t-m}, \cdots w_{t-1}, w_{t+1}, \cdots, w_{t+m})$$



Word Embeddings
GloVe

Comparison

- Count-based
  - LSA, HAL (Lund & Burgess), COALS (Rohde et al), Hellinger-PCA (Lebret & Collobert)
  - Pros
    - ✓ Fast training
    - ✓ Efficient usage of statistics
  - Cons
    - ✓ Primarily used to capture word similarity
    - ✓ Disproportionate importance given to large counts
- Direct prediction
  - NNLM, HLBL, RNN, Skipgram/CBOW (Bengio et al; Collobert & Weston; Huang et al; Mnih & Hinton; Mikolov et al; Mnih & Kavukcuoglu)
  - Pros
    - ✓ Generate improved performance on other tasks
    - ✓ Capture complex patterns beyond word similarity
  - Cons
    - ✓ Benefits mainly from large corpus
    - ✓ Inefficient usage of statistics

Combining the benefits from both worlds → GloVe

GloVe

- Idea: ratio of co-occurrence probability can encode meaning
- P_{ij} is the probability that word w_j appears in the context of word w_i

$$P_{ij} = P(w_j \mid w_i) = X_{ij}/X_i$$

- Relationship between the words w_i and w_j

$$\begin{aligned} v_{w_i} \cdot v'_{\tilde{w}_j} &= v_{w_i}^T v'_{\tilde{w}_j} = \log P(w_j \mid w_i) \\ &= \log P_{ij} = \log(X_{ij}) - \log(X_i) \end{aligned} \qquad \boxed{P_{ij} = X_{ij}/X_i}$$

$$C(\theta) = \sum_{i,j=1}^{V} f(X_{ij})(v_{w_i}^T v'_{\tilde{w}_j} + b_i + \tilde{b}_j - \log X_{ij})^2$$

Pennington et al., ”GloVe: Global Vectors for Word Representation,” in EMNLP, 2014.

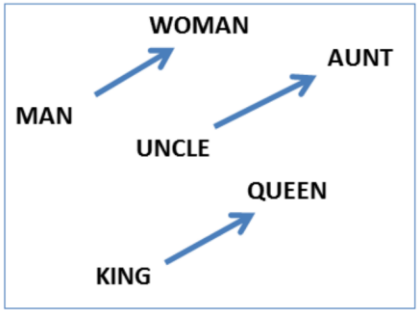
Word Vector Evaluation

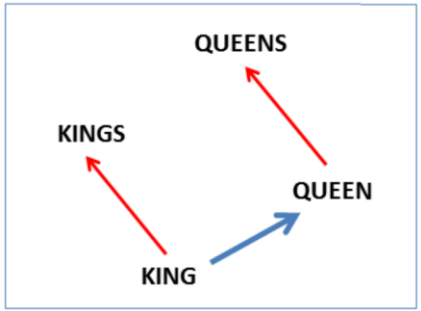
Intrinsic Evaluation – Word Analogies

- Word linear relationship $w_A : w_B = w_C : w_x$

$$x = \arg\max_x \frac{(v_{w_B} - v_{w_A} + v_{w_C})^T v_{w_x}}{\left\| v_{w_B} - v_{w_A} + v_{w_C} \right\|}$$

- **Syntactic** and **Semantic** example questions [link]





Issue: what if the information is there but not linear

Intrinsic Evaluation – Word Analogies

- Word linear relationship $w_A : w_B = w_C : w_x$
- Syntactic and **Semantic** example questions [link]

city---in---state
Chicago : Illinois = Houston : Texas
Chicago : Illinois = Philadelphia : Pennsylvania
Chicago : Illinois = Phoenix : Arizona
Chicago : Illinois = Dallas : Texas
Chicago : Illinois = Jacksonville : Florida
Chicago : Illinois = Indianapolis : Indiana
Chicago : Illinois = Aus8n : Texas
Chicago : Illinois = Detroit : Michigan
Chicago : Illinois = Memphis : Tennessee
Chicago : Illinois = Boston : Massachusetts

Issue: different cities may have same name

capital---country
Abuja : Nigeria = Accra : Ghana
Abuja : Nigeria = Algiers : Algeria
Abuja : Nigeria = Amman : Jordan
Abuja : Nigeria = Ankara : Turkey
Abuja : Nigeria = Antananarivo : Madagascar
Abuja : Nigeria = Apia : Samoa
Abuja : Nigeria = Ashgabat : Turkmenistan
Abuja : Nigeria = Asmara : Eritrea
Abuja : Nigeria = Astana : Kazakhstan

Issue: can change with time

Intrinsic Evaluation – Word Analogies

- Word linear relationship $w_A : w_B = w_C : w_x$
- **Syntactic** and Semantic example questions [link]

superlative
bad : worst = big : biggest
bad : worst = bright : brightest
bad : worst = cold : coldest
bad : worst = cool : coolest
bad : worst = dark : darkest
bad : worst = easy : easiest
bad : worst = fast : fastest
bad : worst = good : best
bad : worst = great : greatest

past tense
dancing : danced = decreasing : decreased
dancing : danced = describing : described
dancing : danced = enhancing : enhanced
dancing : danced = falling : fell
dancing : danced = feeding : fed
dancing : danced = flying : flew
dancing : danced = generating : generated
dancing : danced = going : went
dancing : danced = hiding : hid
dancing : danced = hiding : hit

Intrinsic Evaluation – Word Correlation

- Comparing word correlation with human-judged scores
- Human-judged word correlation [link]

Word 1	Word 2	Human-Judged Score
tiger	cat	7.35
tiger	tiger	10.00
book	paper	7.46
computer	internet	7.58
plane	car	5.77
professor	doctor	6.62
stock	phone	1.62

Ambiguity: synonym or same word with different POSs

Extrinsic Evaluation – Subsequent Task

- Goal: use word vectors in neural net models built for subsequent tasks
- Benefit
  - Ability to also classify words accurately
    - Ex. countries cluster together a classifying location words should be possible with word vectors
  - Incorporate any information into them other tasks
    - Ex. project sentiment into words to find most positive/negative words in corpus

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
- Low dimensional word vector
  - word2vec
  - GloVe: combining count-based and direct learning
- Word vector evaluation
  - Intrinsic: word analogy, word correlation
  - Extrinsic: subsequent task