Unsupervised Learning: Word Embedding
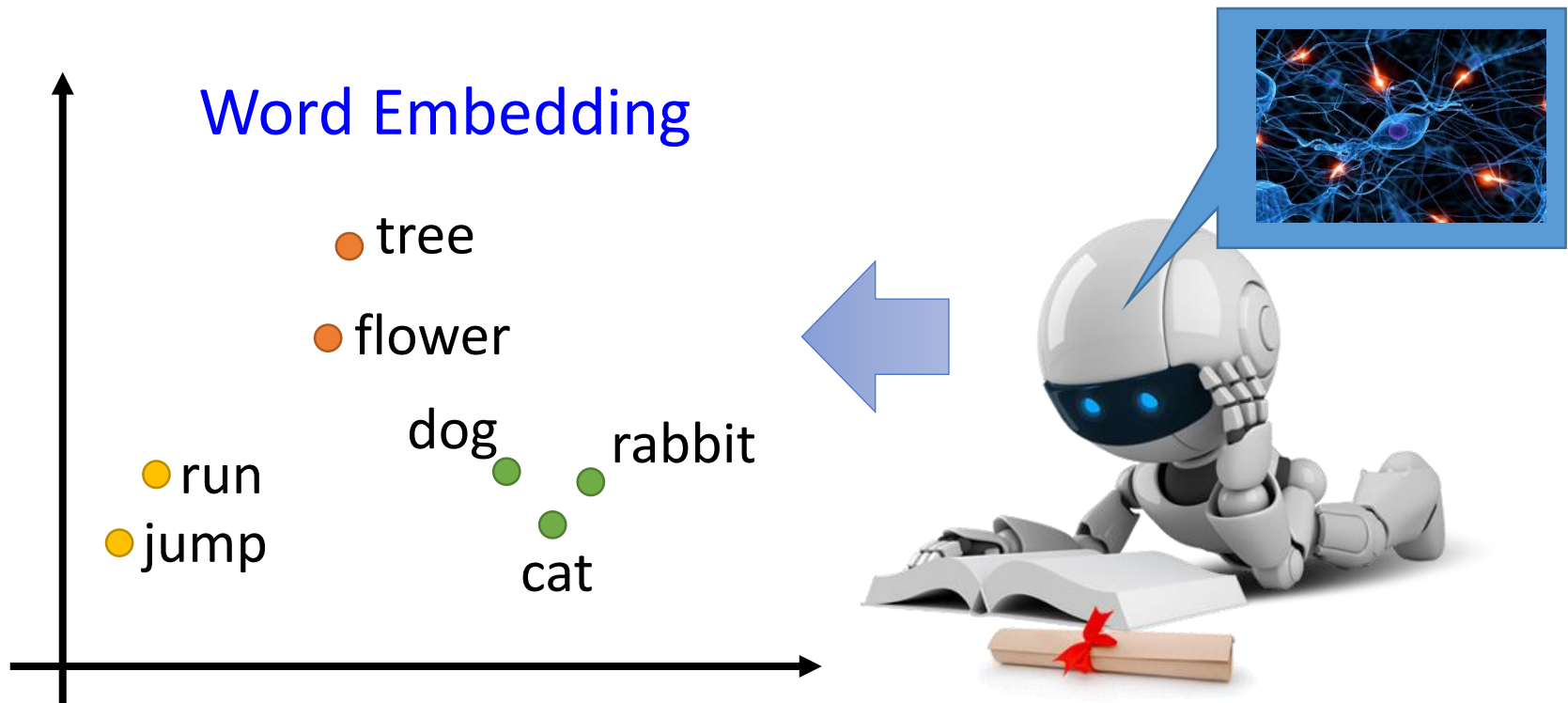
Word Embedding

- Machine learns the meaning of words from reading a lot of documents without supervision

Word Embedding

1-of-N Encoding
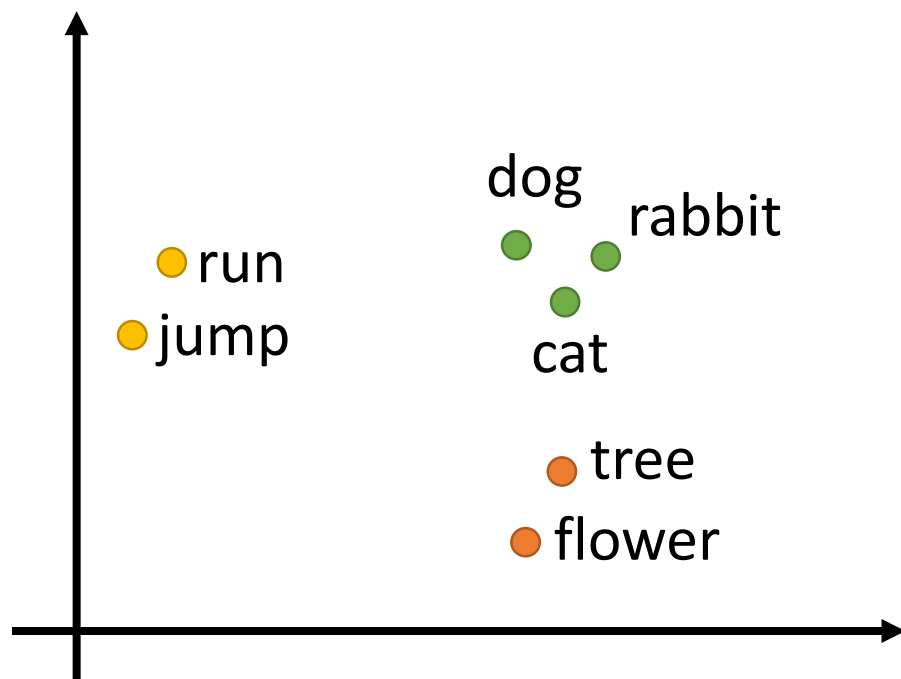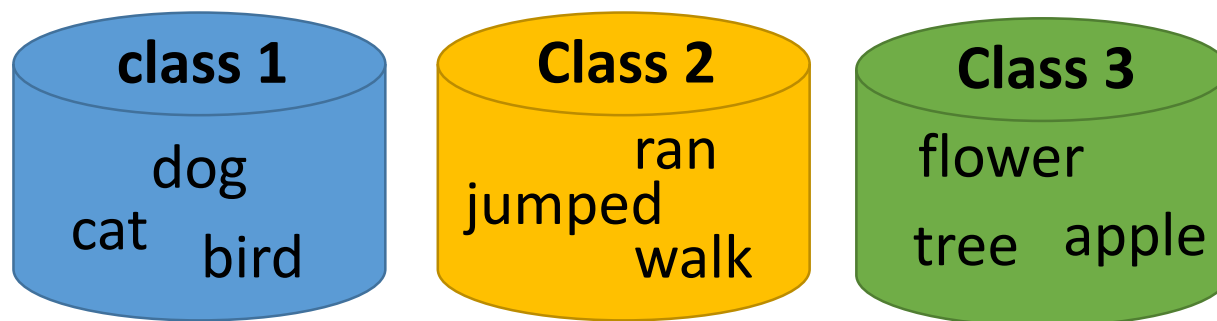
apple = [1 0 0 0 0]

bag   = [0 1 0 0 0]

cat   = [0 0 1 0 0]

dog   = [0 0 0 1 0]

elephant = [0 0 0 0 1]

Word Embedding

run
jump

dog
rabbit
cat

tree
flower

Word Class

class 1
dog
cat
bird

Class 2
ran
jumped
walk

Class 3
flower
tree apple

Word Embedding

- Machine learns the meaning of words from reading a lot of documents without supervision

- A word can be understood by its context

蔡英文、馬英九 are something very similar

You shall know a word by the company it keeps

馬英九 520宣誓就職

蔡英文 520宣誓就職

How to exploit the context?

- **Count based**
  - If two words w_i and w_j frequently co-occur, $V(w_i)$ and $V(w_j)$ would be close to each other
  - E.g. Glove Vector: http://nlp.stanford.edu/projects/glove/

$$V(w_i) \cdot V(w_j) \longleftrightarrow N_{i,j}$$

Inner product
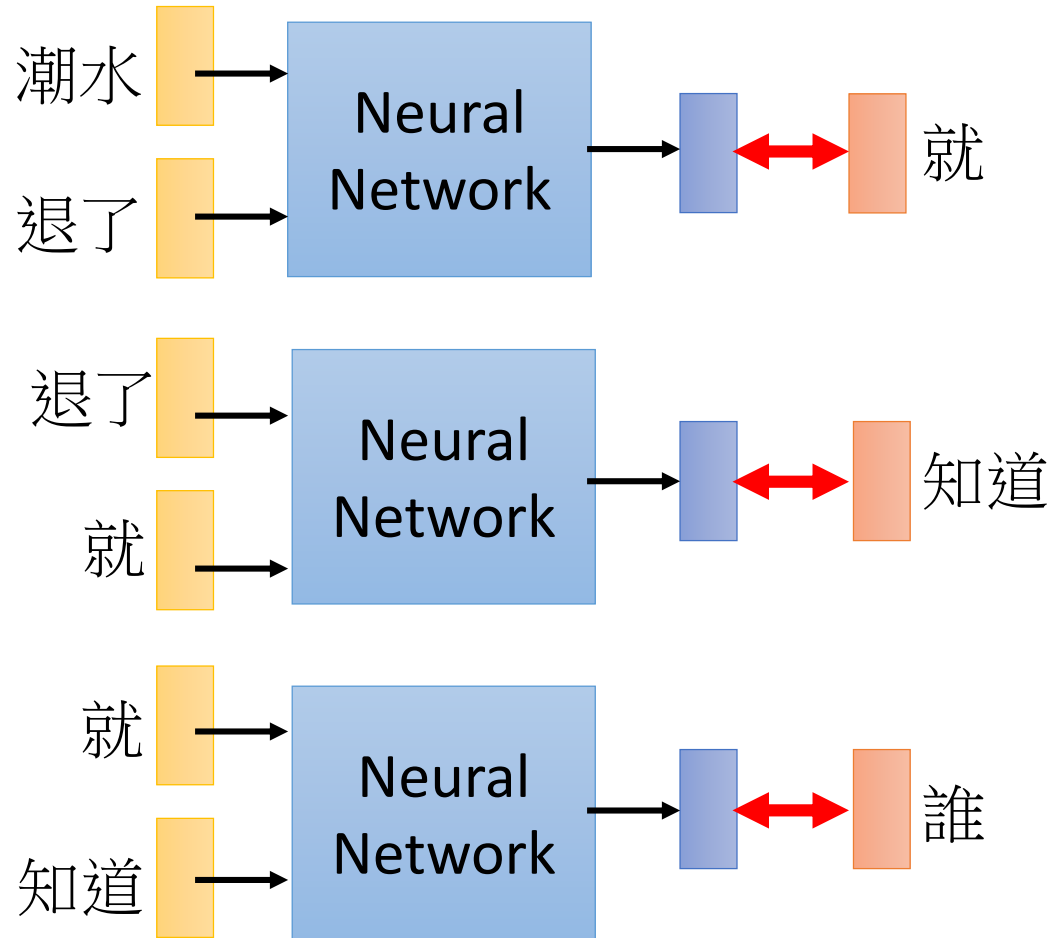
Number of times w_i and w_j in the same document

- **Perdition based**

Prediction-based – Training

Collect data:

潮水 退了 就 知道 誰 …
不爽 不要 買 …
公道價 八萬 一 …
………

**Minimizing
cross entropy**

Prediction-based - 推文接話

推 louisee :話說十幾年前我念公立國中時,老師也曾做過這種事,但

https://www.ptt.cc/bbs/Teacher/M.1317226791.A.558.html

推 AO56789: 我同學才扯好不好，他有一次要交家政料理報告
→ AO56789:其中一個是要寫一樣水煮料理的食譜，他居然給我寫

著名簽名檔 (出處不詳)

Prediction-based – Language Modeling

P("wreck a nice beach")

=P(wreck|START)P(a|wreck)P(nice|a)P(beach|nice)

P(b|a): the probability of NN predicting the next word.

P(next word is "wreck")

P(next word is "a")

P(next word is "nice")

P(next word is "beach")

| Neural Network | Neural Network | Neural Network | Neural Network |

1-of-N encoding of "START"

1-of-N encoding of "wreck"

1-of-N encoding of "a"

1-of-N encoding of "nice"

i-th output $= P(w_t = i \mid context)$

softmax

most computation here

tanh

$C(w_{t-n+1})$

$C(w_{t-2})$

$C(w_{t-1})$

Table look−up in C

Matrix C

shared parameters across words

index for w_{t-n+1}

index for w_{t-2}
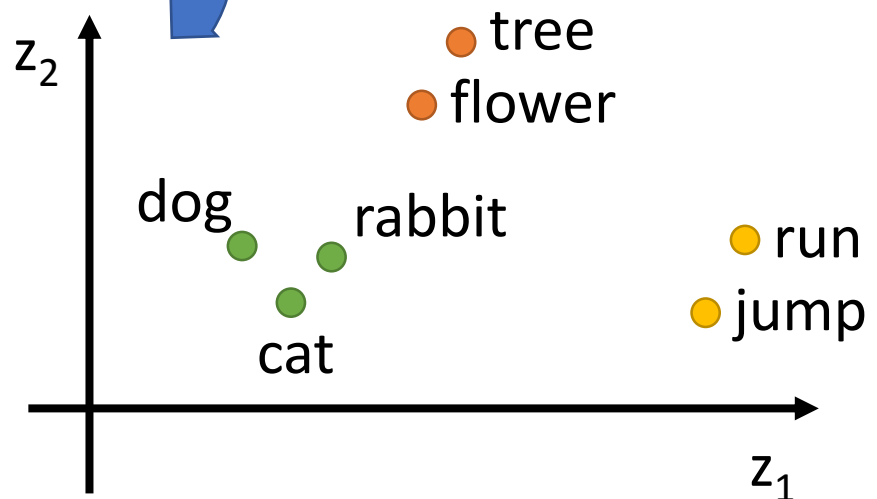
index for w_{t-1}

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, *3*(Feb), 1137-1155.
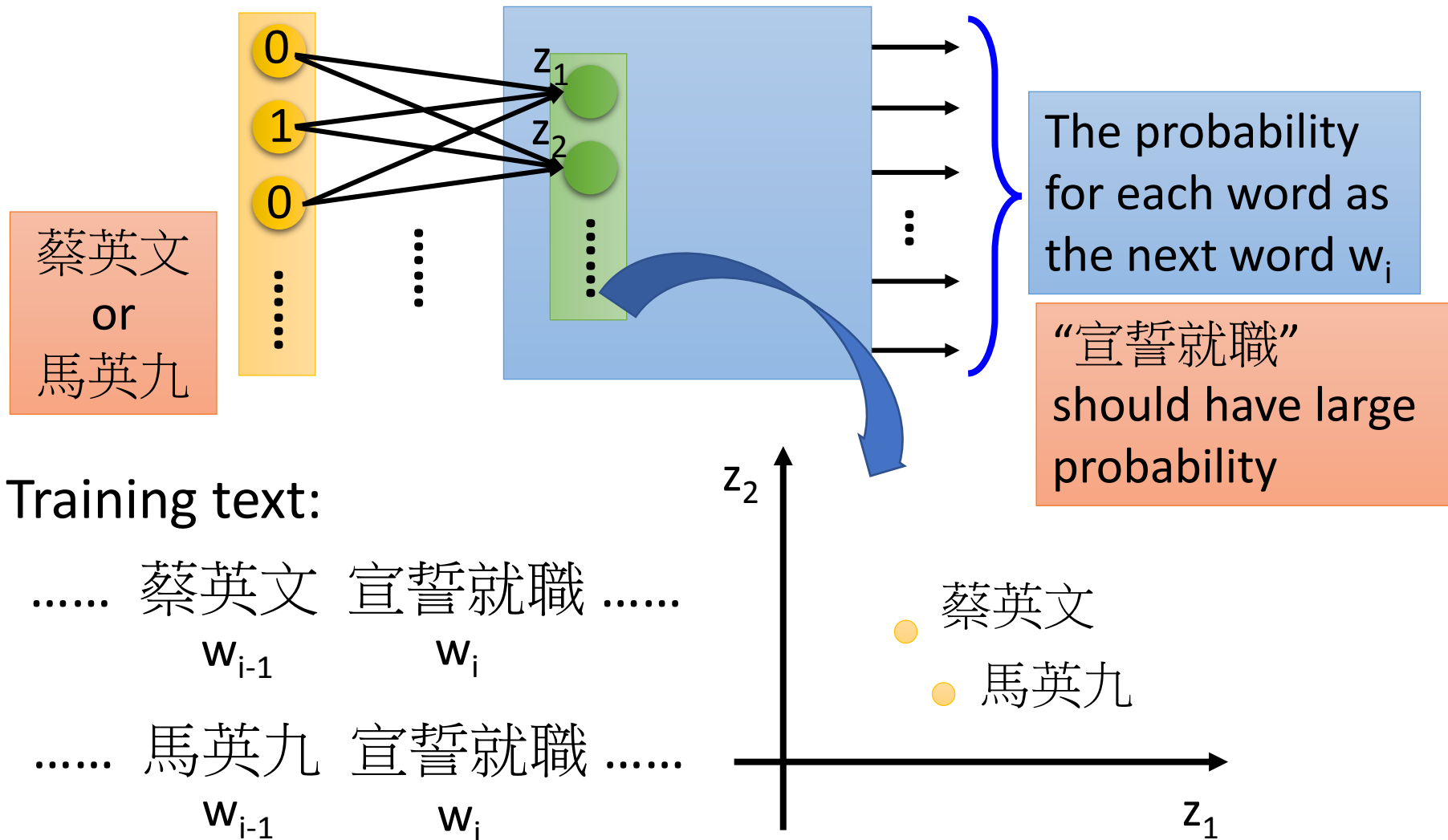
Prediction-based

...... w_{i-2} w_{i-1} w_i

1-of-N encoding of the word w_{i-1}

0
1
0

z_1
z_2

The probability for each word as the next word w_i
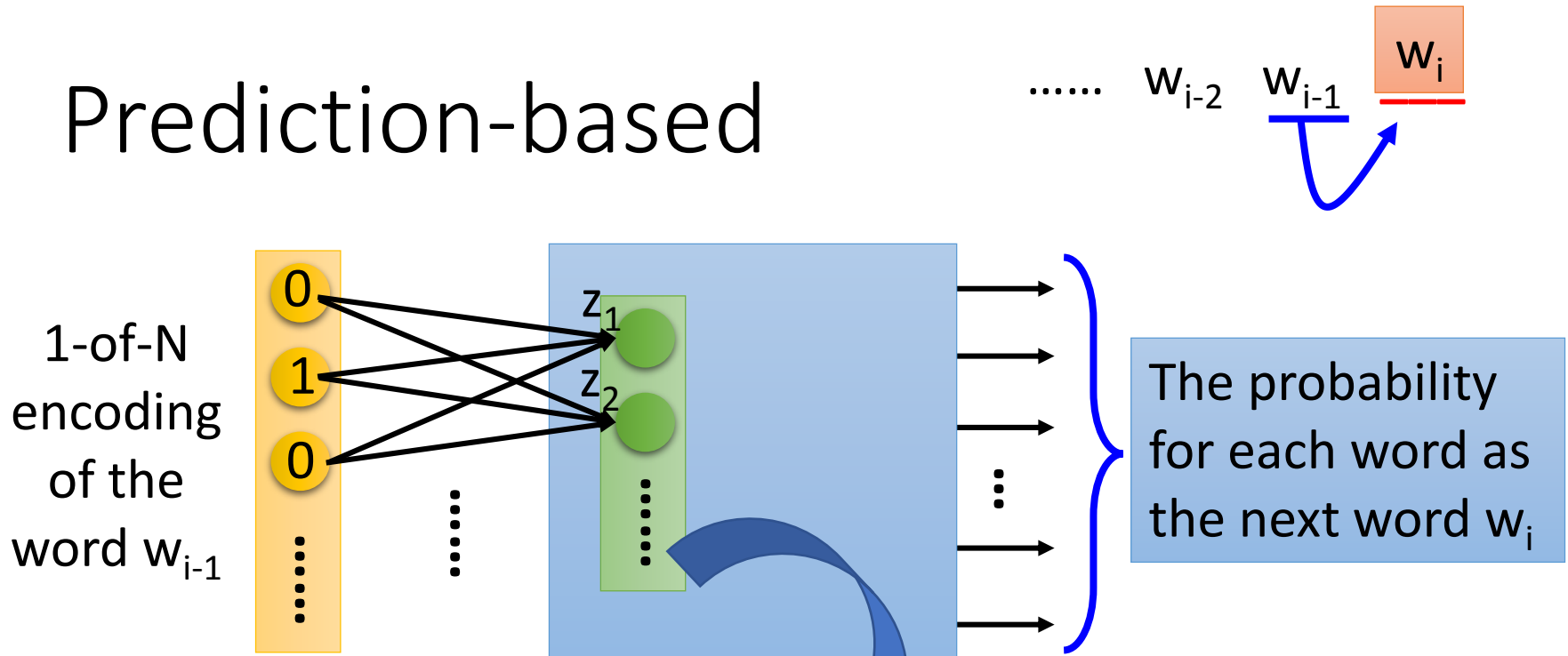
➢ Take out the input of the neurons in the first layer

➢ Use it to represent a word w

➢ Word vector, word embedding feature: V(w)

z_2

tree
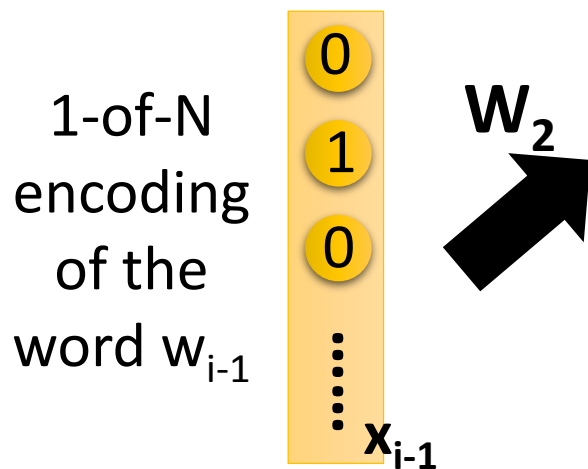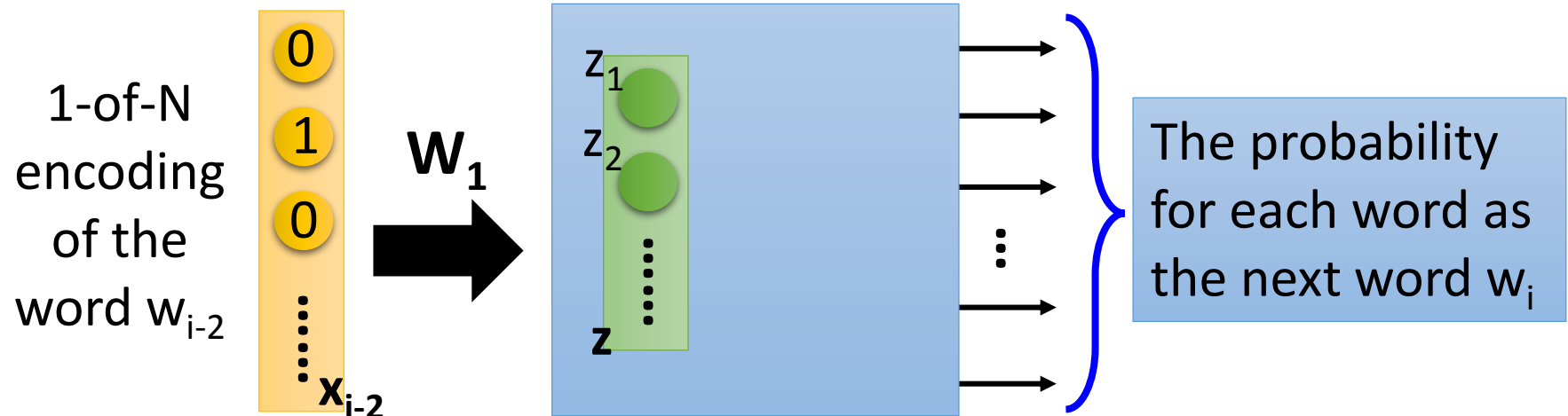flower
dog rabbit
cat
run
jump

z_1

Prediction-based

You shall know a word by the company it keeps

蔡英文
or
馬英九

0
1
0
⋮

z_1
z_2
⋮

The probability for each word as the next word w_i

"宣誓就職" should have large probability

Training text:

...... 蔡英文 宣誓就職 ......
　　　w_{i-1}　　w_i

...... 馬英九 宣誓就職 ......
　　　w_{i-1}　　w_i

z_2

蔡英文
馬英九

z_1

Prediction-based — Sharing Parameters

1-of-N encoding of the word w_{i-2}

W_1

z_1
z_2
⋮
z

x_{i-2}

The probability for each word as the next word w_i

1-of-N encoding of the word w_{i-1}

W_2

x_{i-1}

The length of x_{i-1} and x_{i-2} are both $|V|$.

The length of z is $|Z|$.

$z = W_1 x_{i-2} + W_2 x_{i-1}$

The weight matrix W_1 and W_2 are both $|Z| \times |V|$ matrices.

$W_1 = W_2 = W$ ➡ $z = W (x_{i-2} + x_{i-1})$

Prediction-based
– Sharing Parameters

1-of-N encoding of the word w_{i-2}

0
1
0
⋮

z_1
z_2

1-of-N encoding of the word w_{i-1}

0
0
1
⋮

The probability for each word as the next word w_i

The weights with the same color should be the same.
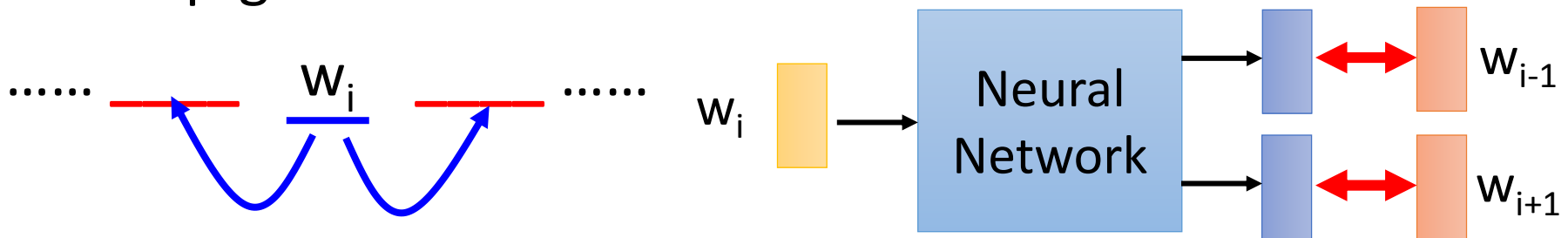
Or, one word would have two word vectors.
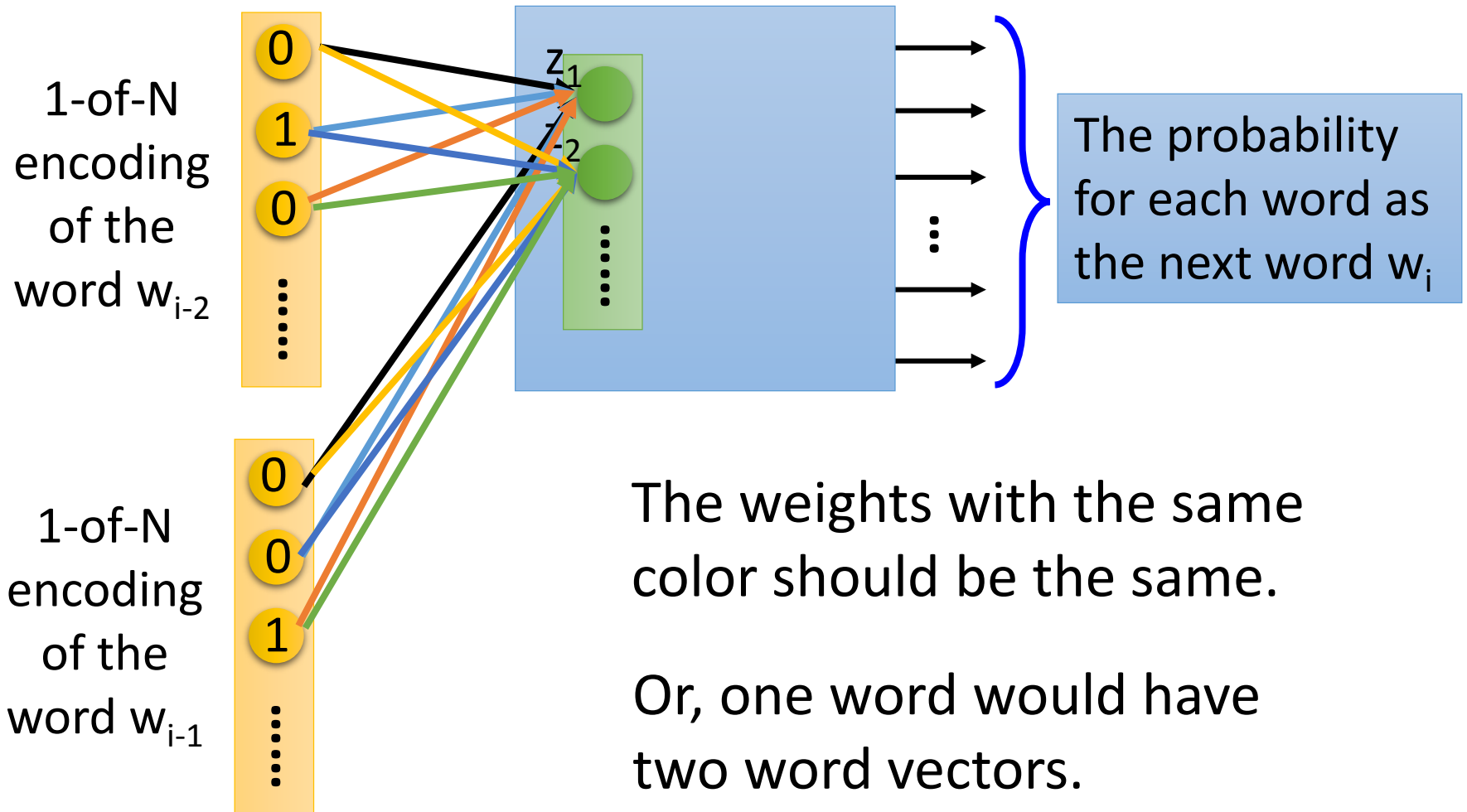
Prediction-based – Various Architectures

- Continuous bag of word (CBOW) model
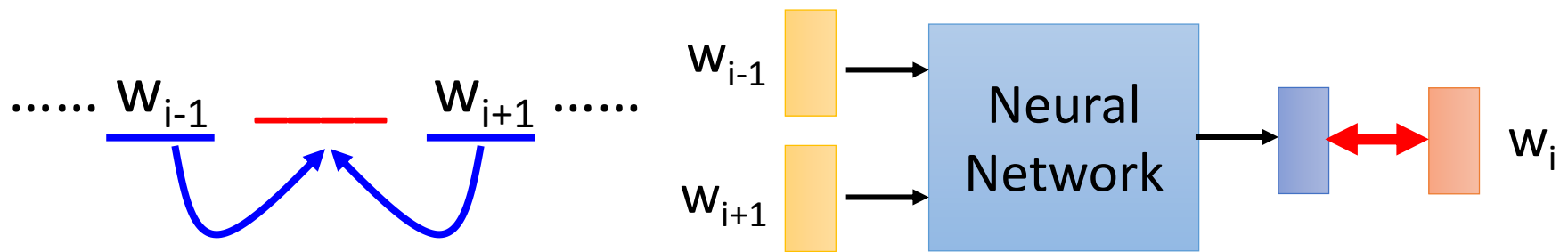
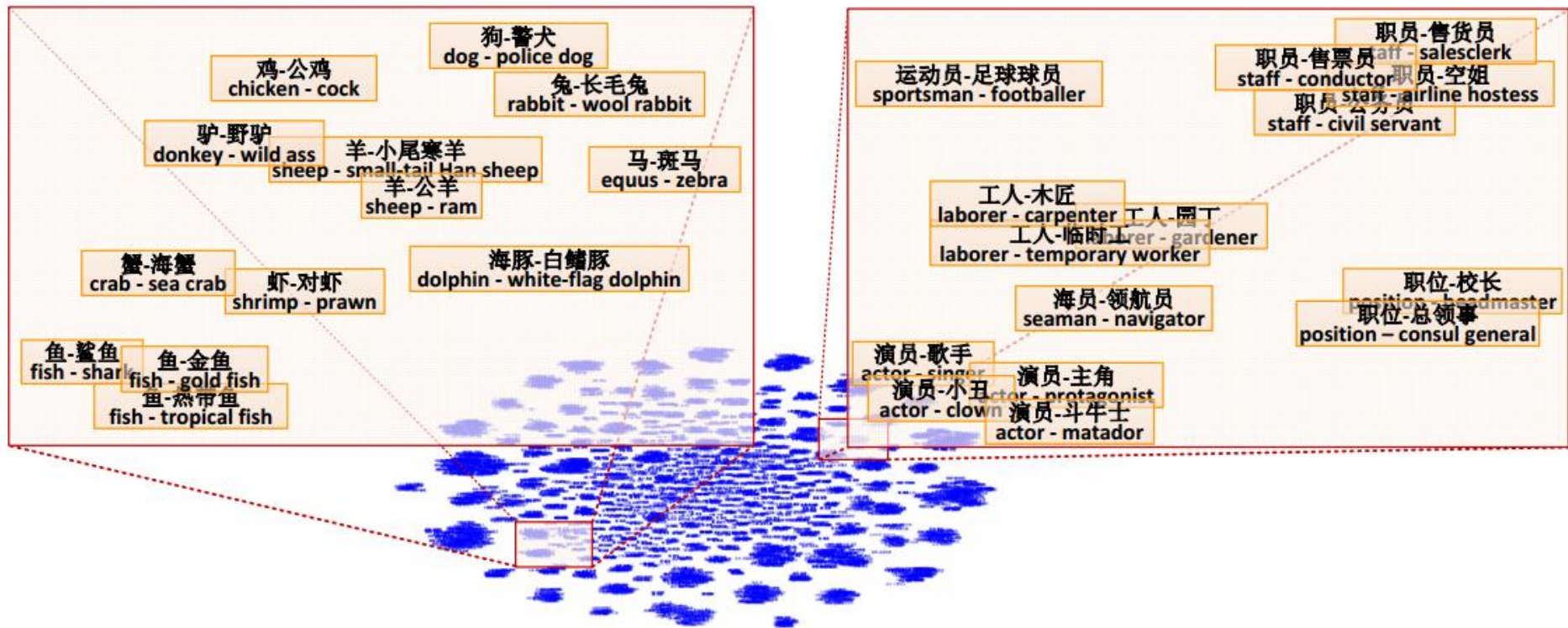*predicting the word given its context*

- Skip-gram

predicting the context given a word

Word Embedding

Source: http://www.slideshare.net/hustwj/cikm-keynotenov2014

15
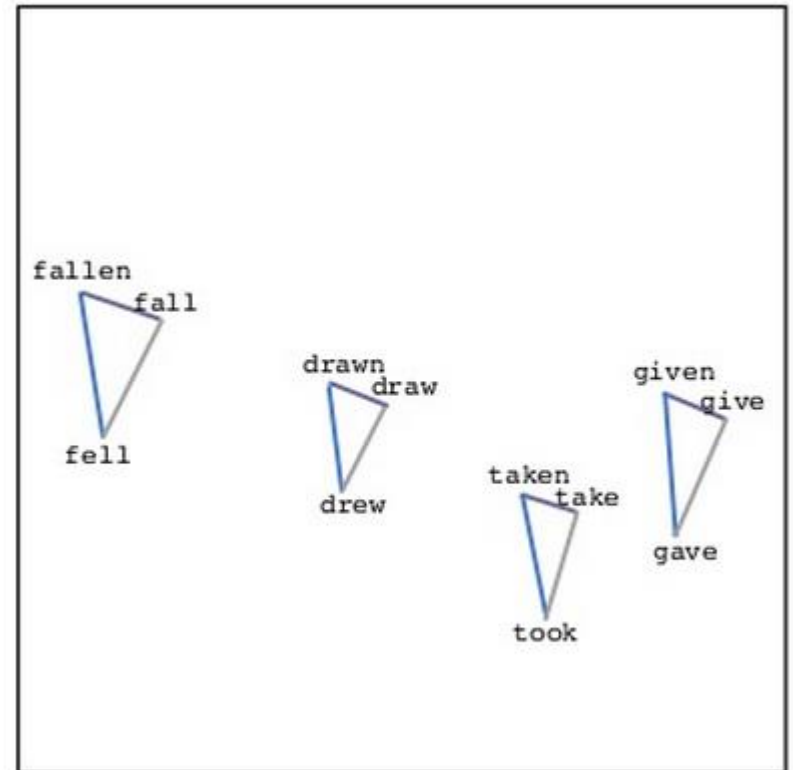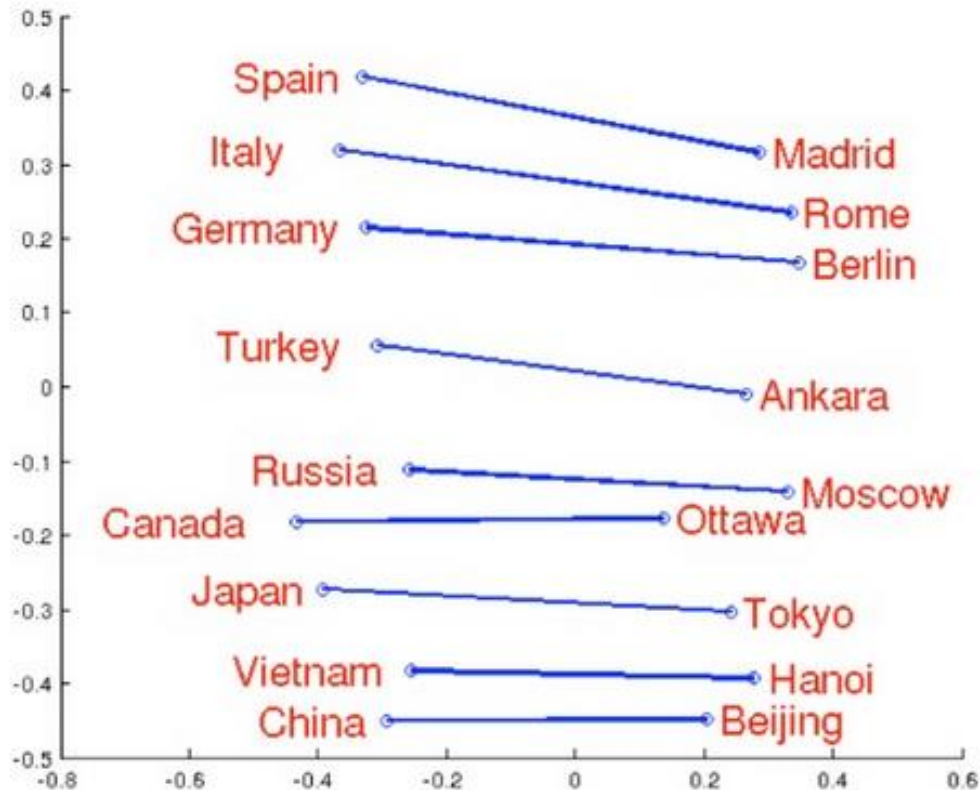
Word Embedding

Fu, Ruiji, et al. "Learning semantic hierarchies via word embeddings."*Proceedings of the 52th Annual Meeting of the Association for Computational Linguistics: Long Papers*. Vol. 1. 2014.

16

Word Embedding

- Characteristics

$$V(Germany)$$
$$\approx V(Berlin) - V(Rome) + V(Italy)$$

$$V(hotter) - V(hot) \approx V(bigger) - V(big)$$
$$V(Rome) - V(Italy) \approx V(Berlin) - V(Germany)$$
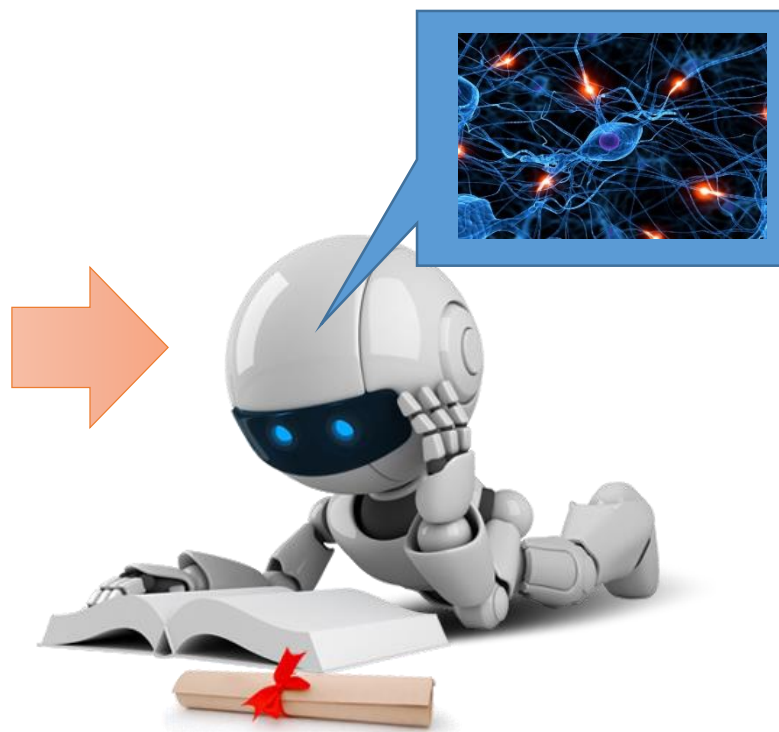$$V(king) - V(queen) \approx V(uncle) - V(aunt)$$

- Solving analogies

Rome : Italy = Berlin : ?

Compute $V(Berlin) - V(Rome) + V(Italy)$

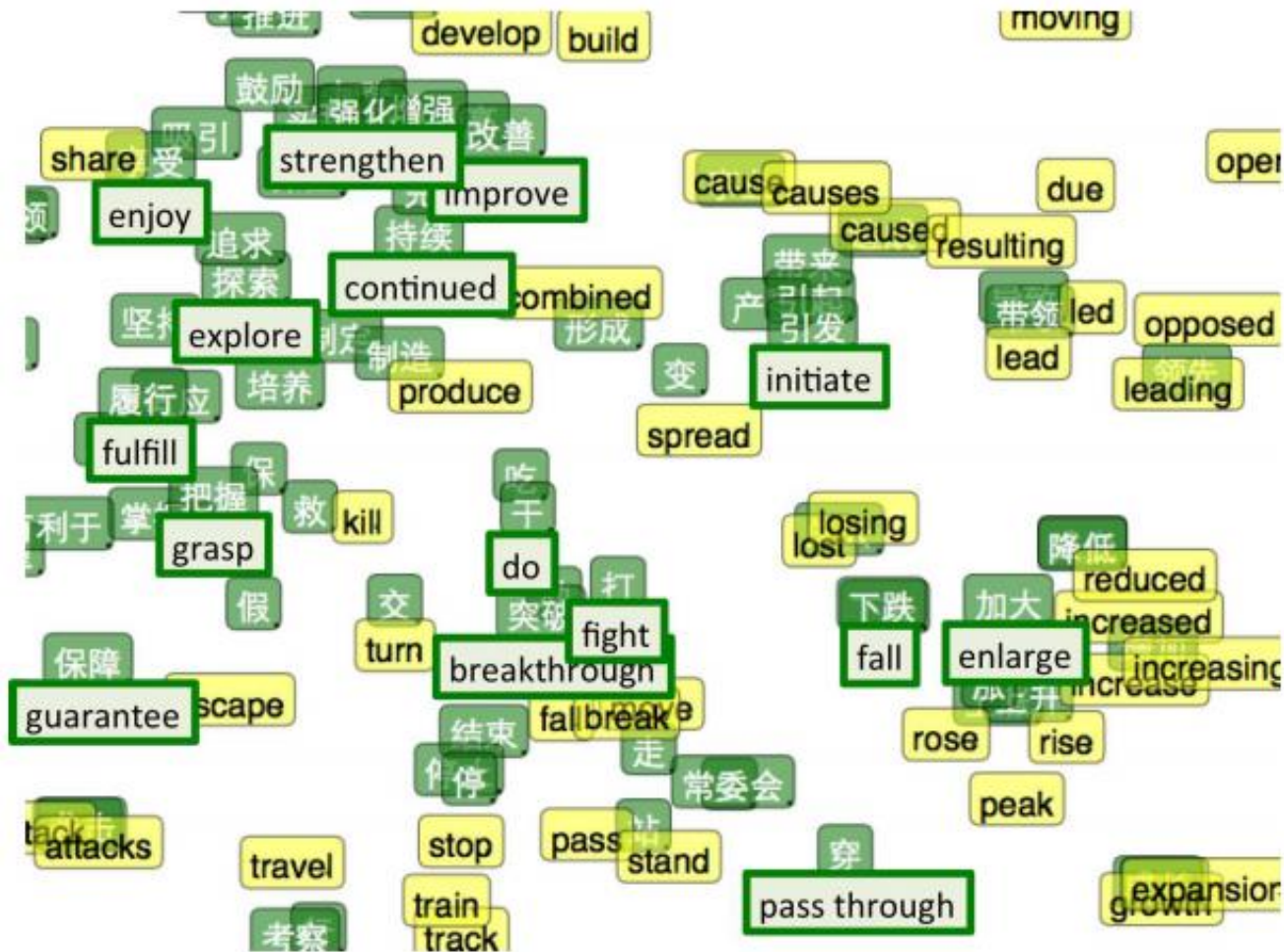Find the word w with the closest V(w)

Demo

- Machine learns the meaning of words from reading a lot of documents without supervision

Demo

- Model used in demo is provided by 陳仰德
  - Part of the project done by 陳仰德、林資偉
  - TA: 劉元銘
  - Training data is from PTT (collected by 葉青峰)

Multi-lingual Embedding

Bilingual Word Embeddings for Phrase-Based Machine Translation, Will Zou, Richard Socher, Daniel Cer and Christopher Manning, EMNLP, 2013
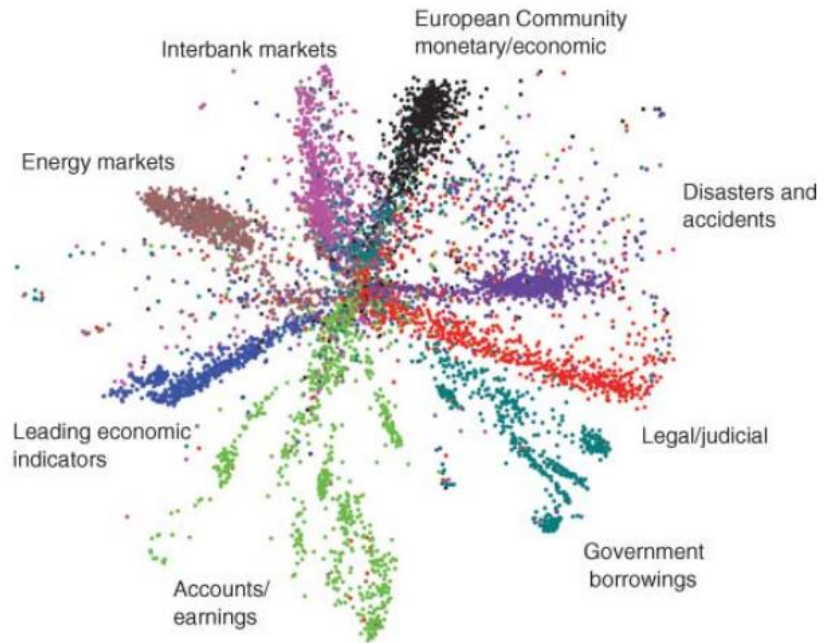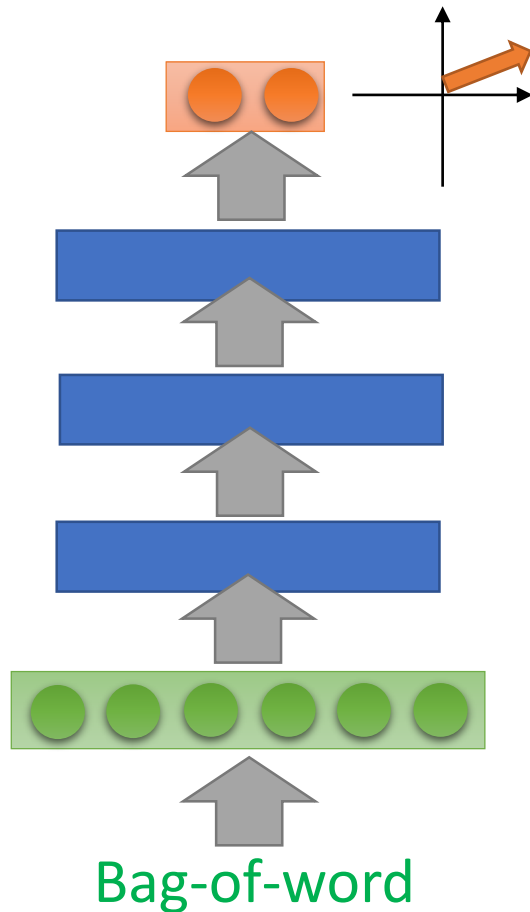
Document Embedding

- word sequences with different lengths → the vector with the same length
  - The vector representing the meaning of the word sequence
  - A word sequence can be a document or a paragraph
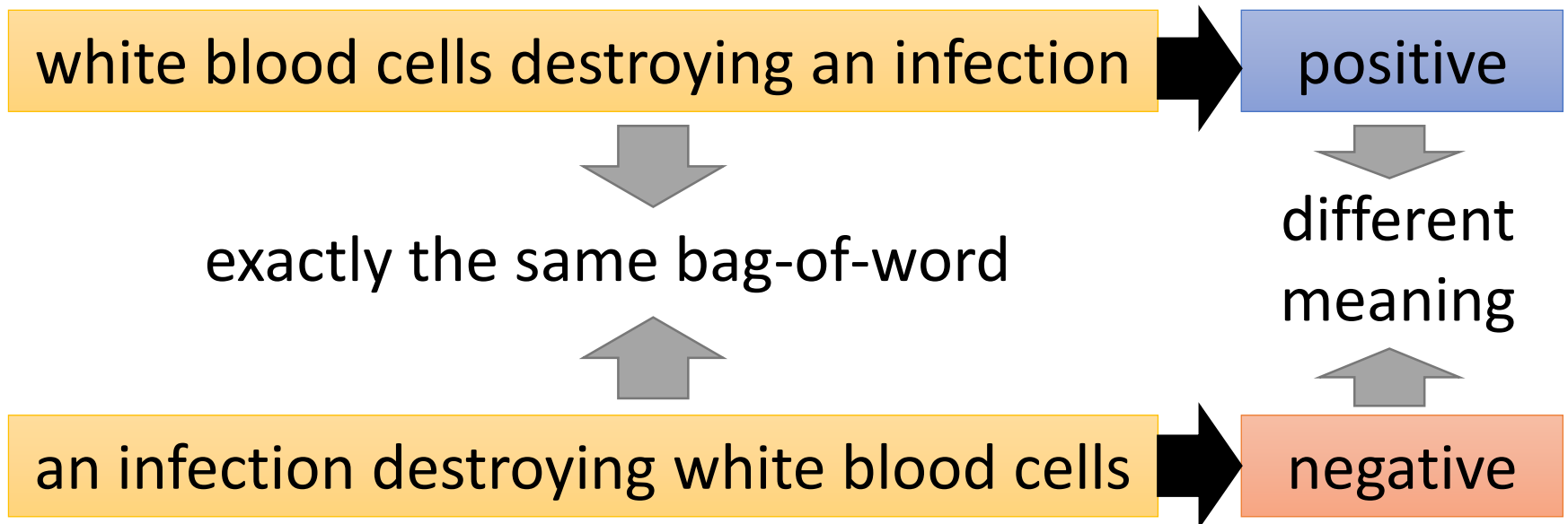
word sequence
(a document or paragraph)
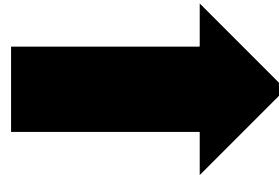
Semantic Embedding

Bag-of-word

Reference: Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks." *Science* 313.5786 (2006): 504-507

Beyond Bag of Word

- To understand the meaning of a word sequence, the order of the words can not be ignored.

white blood cells destroying an infection	➡	positive

exactly the same bag-of-word

different meaning

an infection destroying white blood cells	➡	negative

Beyond Bag of Word

- ***Paragraph Vector***: Le, Quoc, and Tomas Mikolov. "Distributed Representations of Sentences and Documents." ICML, 2014

- ***Seq2seq Auto-encoder***: Li, Jiwei, Minh-Thang Luong, and Dan Jurafsky. "A hierarchical neural autoencoder for paragraphs and documents." arXiv preprint, 2015

- ***Skip Thought***: Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, Sanja Fidler, "Skip-Thought Vectors" arXiv preprint, 2015.