Unsupervised Learning: Word Embedding

1-of-N Encoding

apple = [1 0 0 0 0]

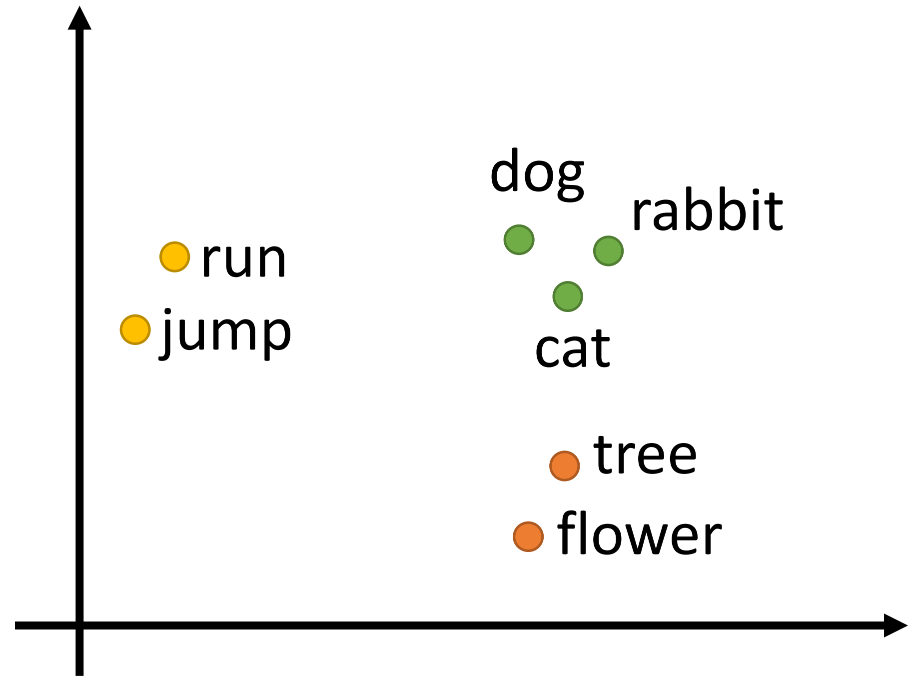
bag = [0 1 0 0 0]

cat = [0 0 1 0 0]

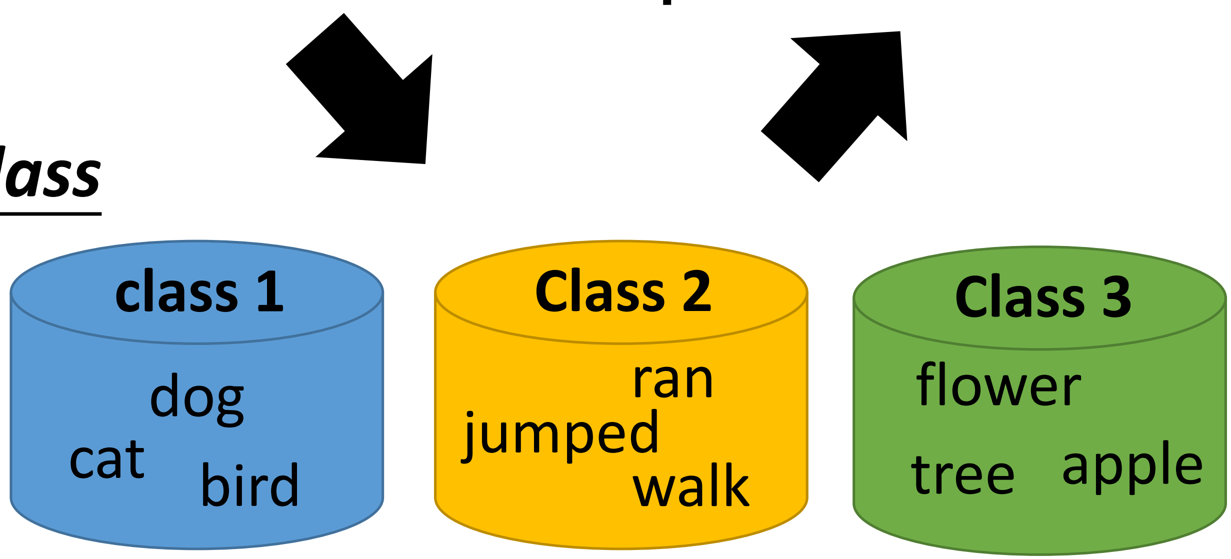
dog = [0 0 0 1 0]

elephant = [0 0 0 0 1]

Word Embedding

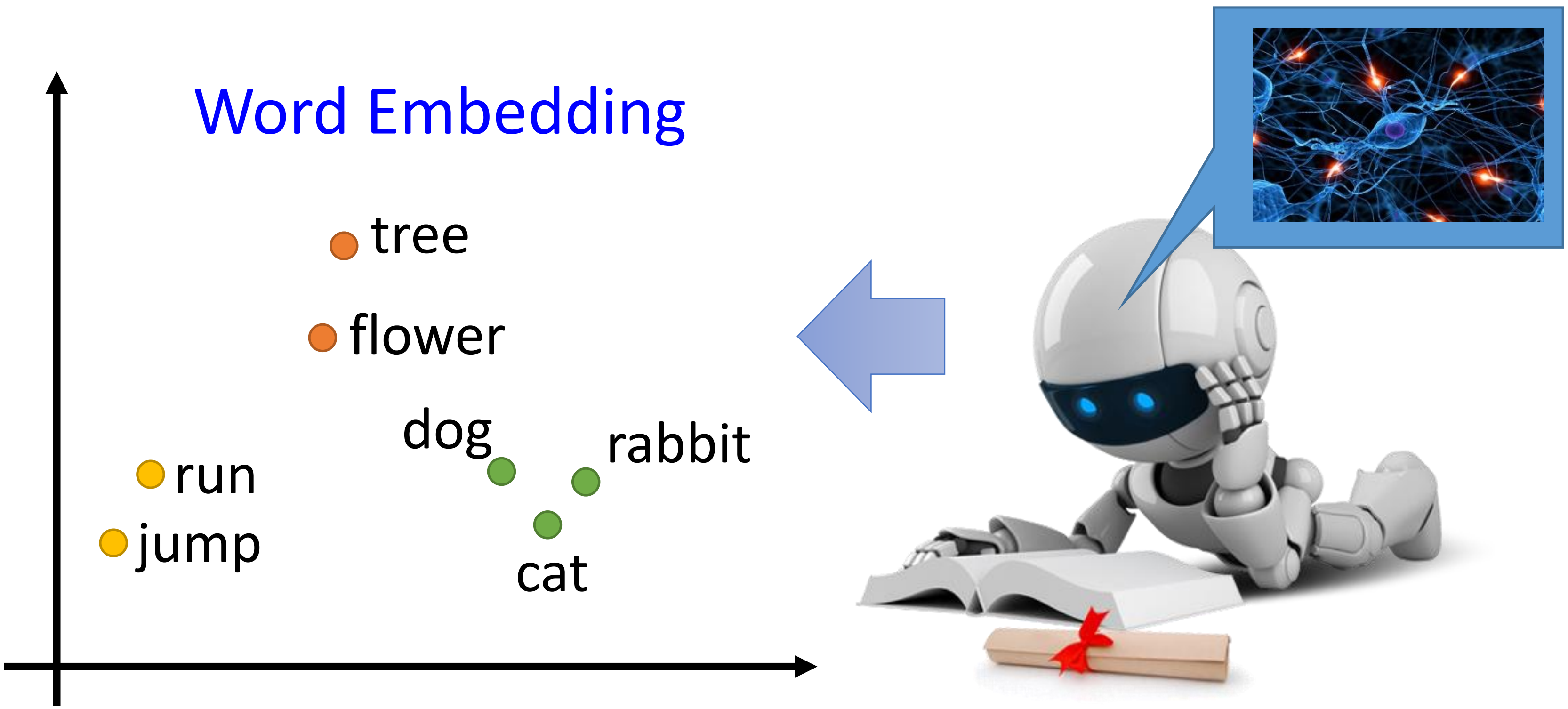


Word Class



Word Embedding

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

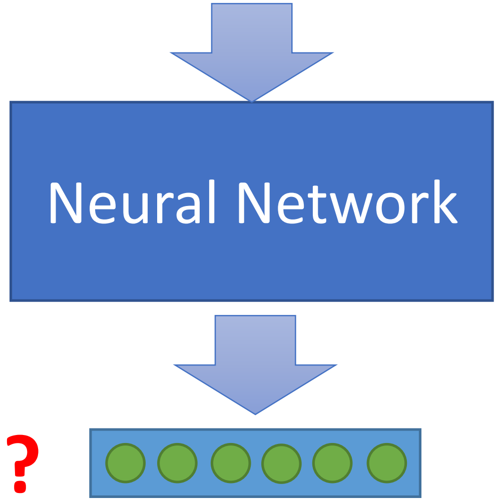


Word Embedding

How about auto-encoder?

- Generating Word Vector is unsupervised

Apple



Training data is a lot of text

https://garavato.files.wordpress.com/2011/11/stacksdocuments.jpg?w=490

Word Embedding

- Machine learn 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/

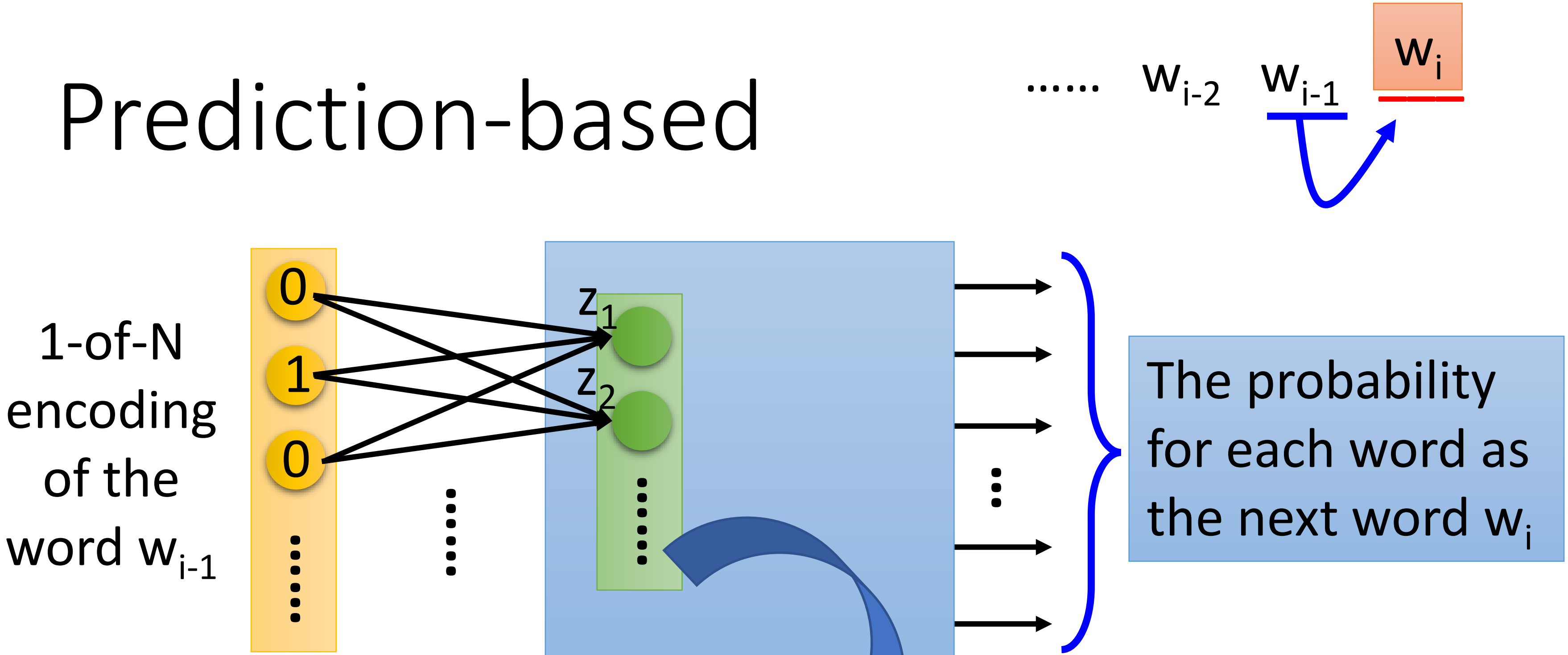
$$V(w_i) . V(w_j) \longleftrightarrow N_{i,j}$$

Inner product

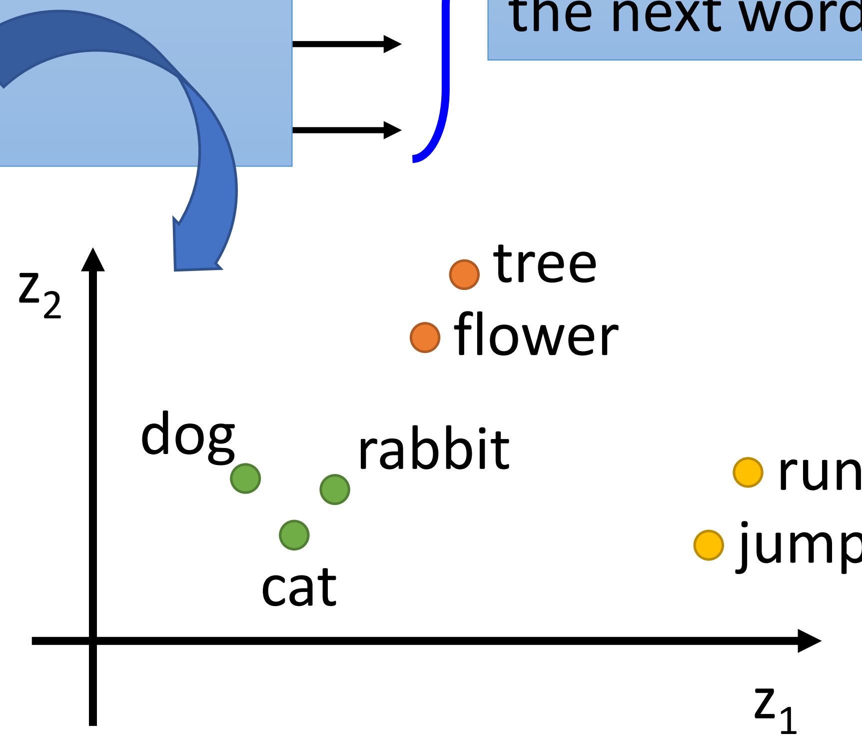
Number of times w_i and w_j in the same document

- **Perdition based**

Prediction-based

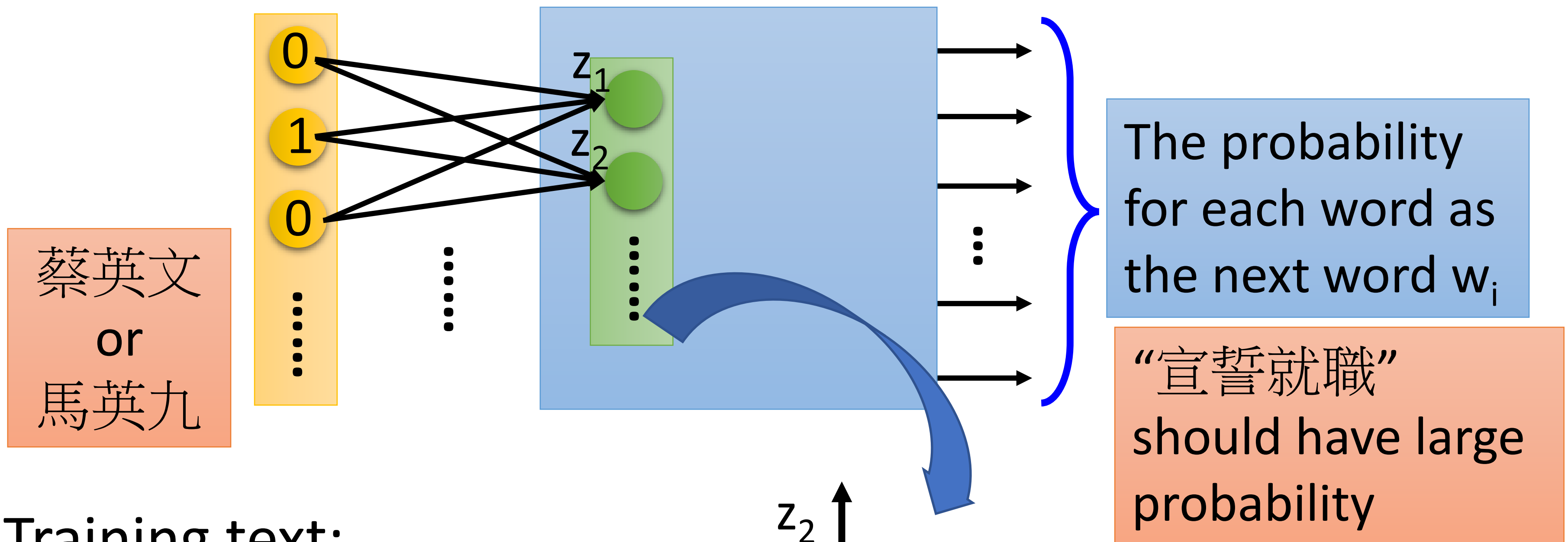


- Take out the input of the neurons in the first layer
- Use it to represent a word w
- Word vector, word embedding feature: V(w)



Prediction-based

You shall know a word by the company it keeps



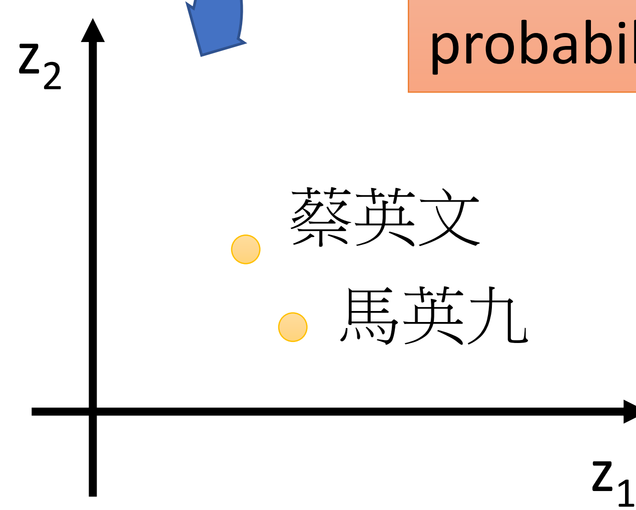
Training text:

…… 蔡英文 宣誓就職 ……

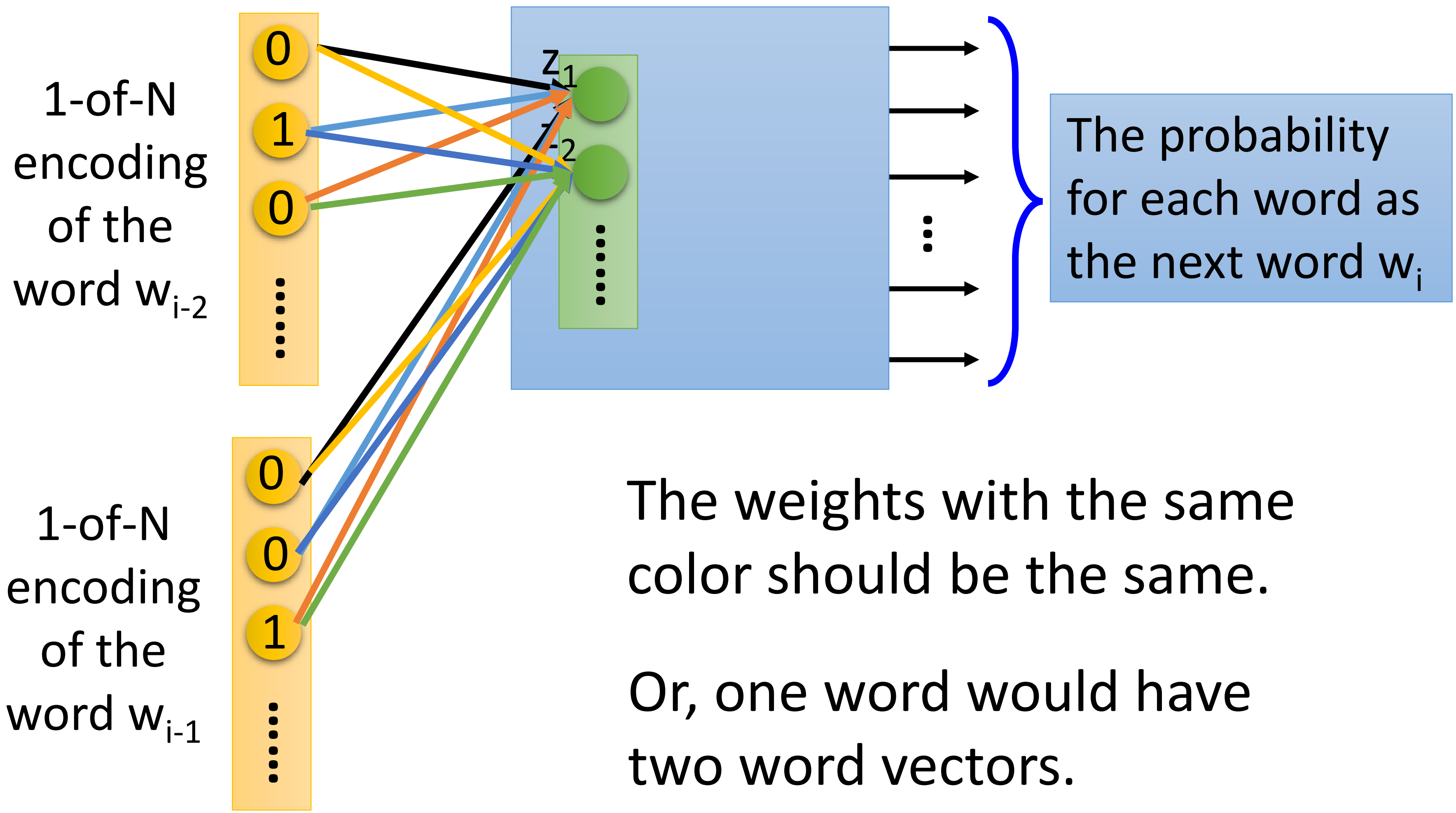
w_{i-1} w_i

…… 馬英九 宣誓就職 ……

w_{i-1} w_i



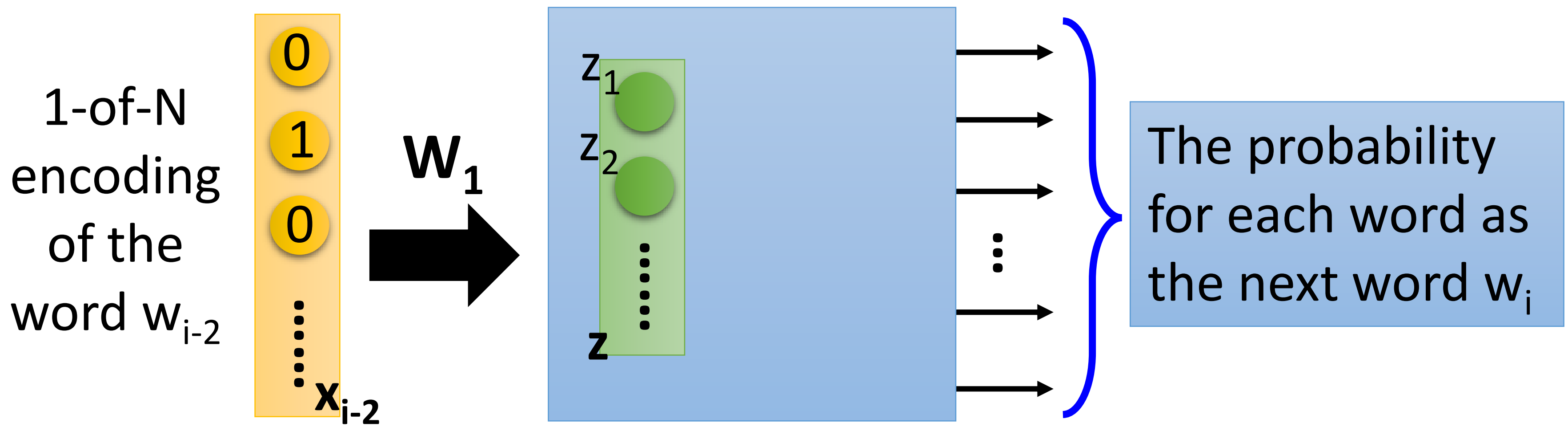
Prediction-based – Sharing Parameters

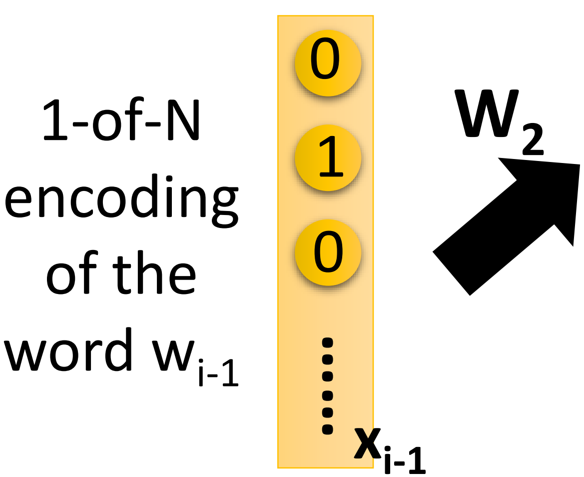


The weights with the same color should be the same.

Or, one word would have two word vectors.

Prediction-based – Sharing Parameters





The length of $\mathbf{x}_{i-1}$ and $\mathbf{x}_{i-2}$ are both |V|.

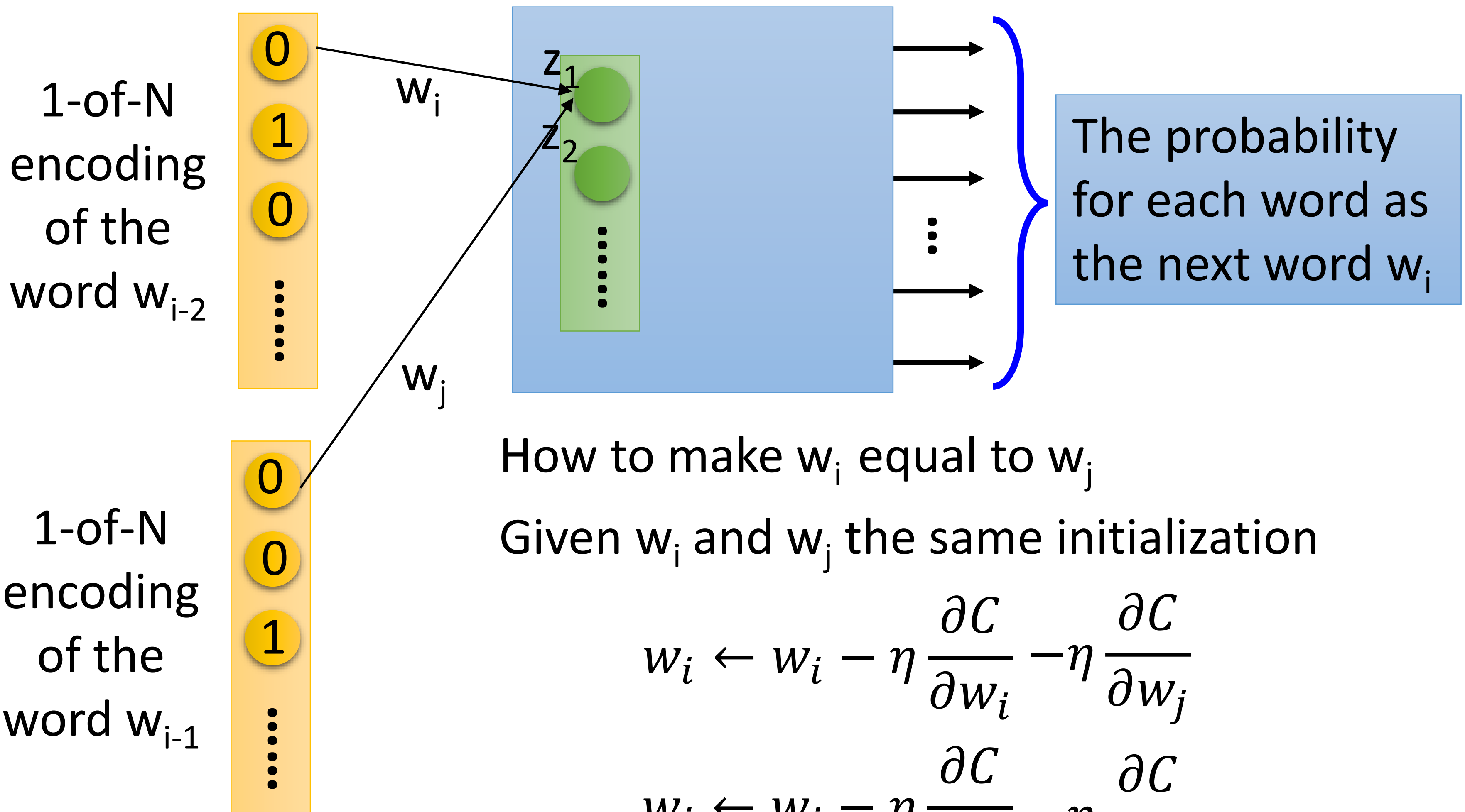
The length of $\mathbf{z}$ is |Z|.

$$\mathbf{z} = \mathbf{W}_1\, \mathbf{x}_{i-2} + \mathbf{W}_2\, \mathbf{x}_{i-1}$$

The weight matrix $\mathbf{W}_1$ and $\mathbf{W}_2$ are both |Z|X|V| matrices.

$$\mathbf{W}_1 = \mathbf{W}_2 = \mathbf{W} \Rightarrow \mathbf{z} = \mathbf{W}\,(\mathbf{x}_{i-2} + \mathbf{x}_{i-1})$$

Prediction-based – Sharing Parameters



How to make w_i equal to w_j

Given w_i and w_j the same initialization

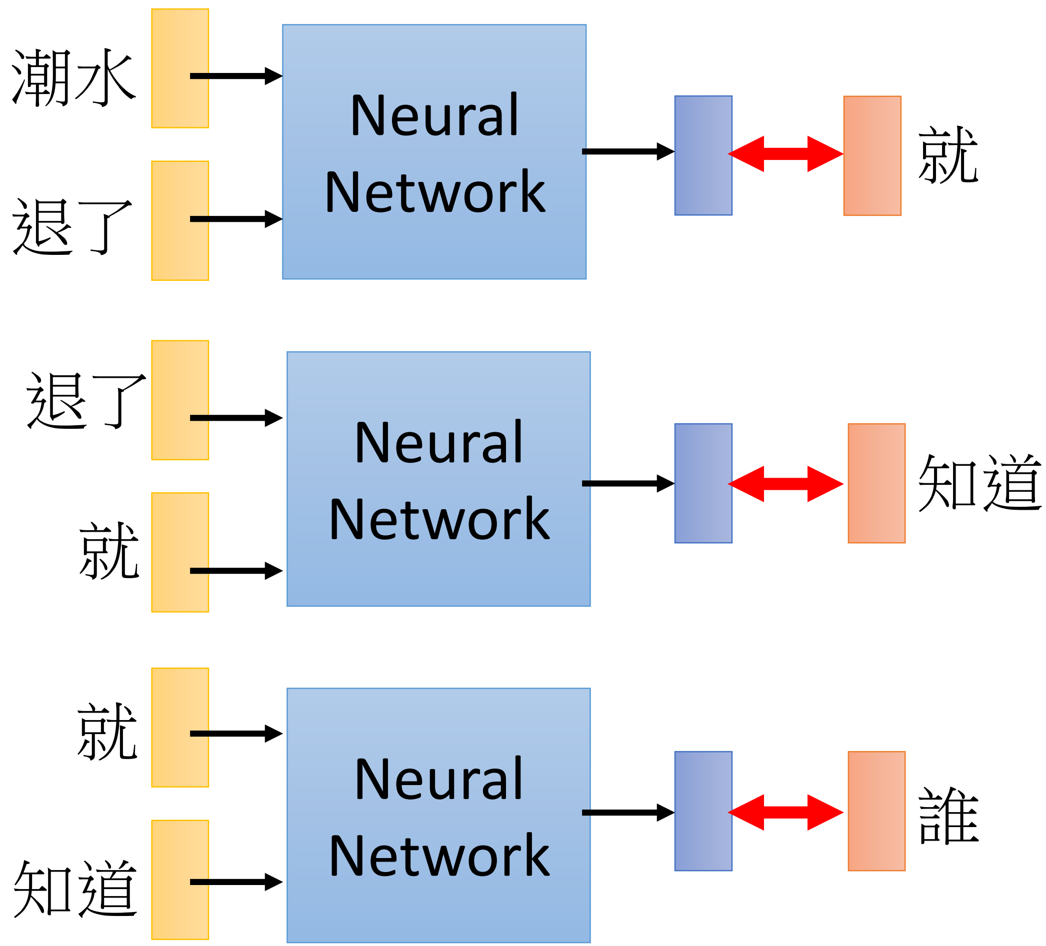
$$w_i \leftarrow w_i - \eta \frac{\partial C}{\partial w_i} - \eta \frac{\partial C}{\partial w_j}$$

$$w_j \leftarrow w_j - \eta \frac{\partial C}{\partial w_j} - \eta \frac{\partial C}{\partial w_i}$$

Prediction-based – Training

Collect data:

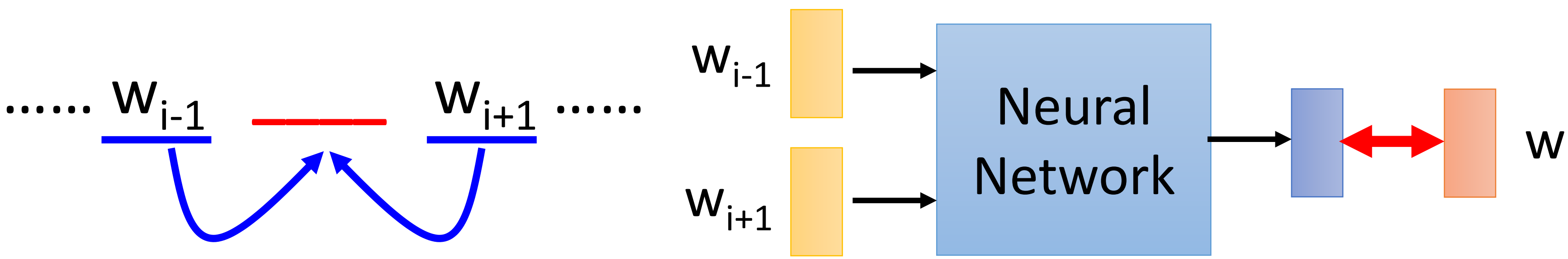
潮水 退了 就 知道 誰 …
不爽 不要 買 …
公道價 八萬 一 …
………



Minimizing cross entropy

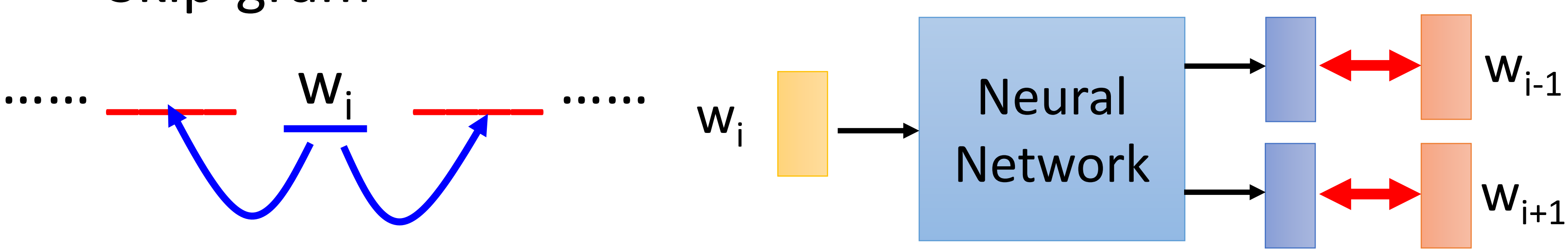
Prediction-based
– Various Architectures

- Continuous bag of word (CBOW) model



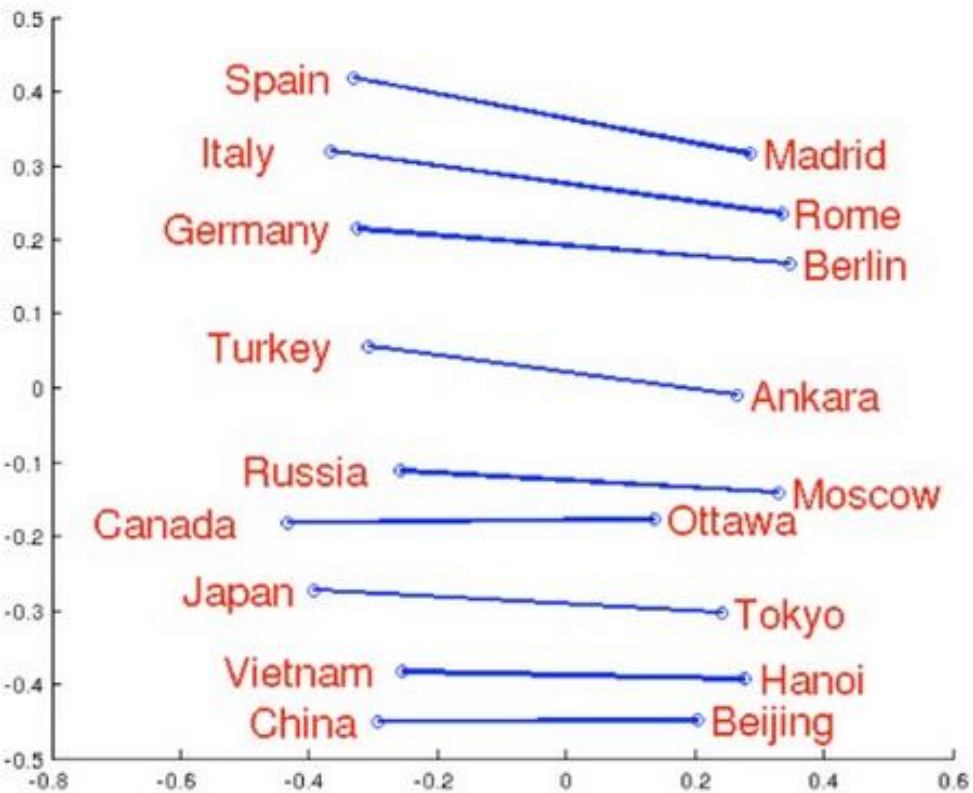
predicting the word given its context

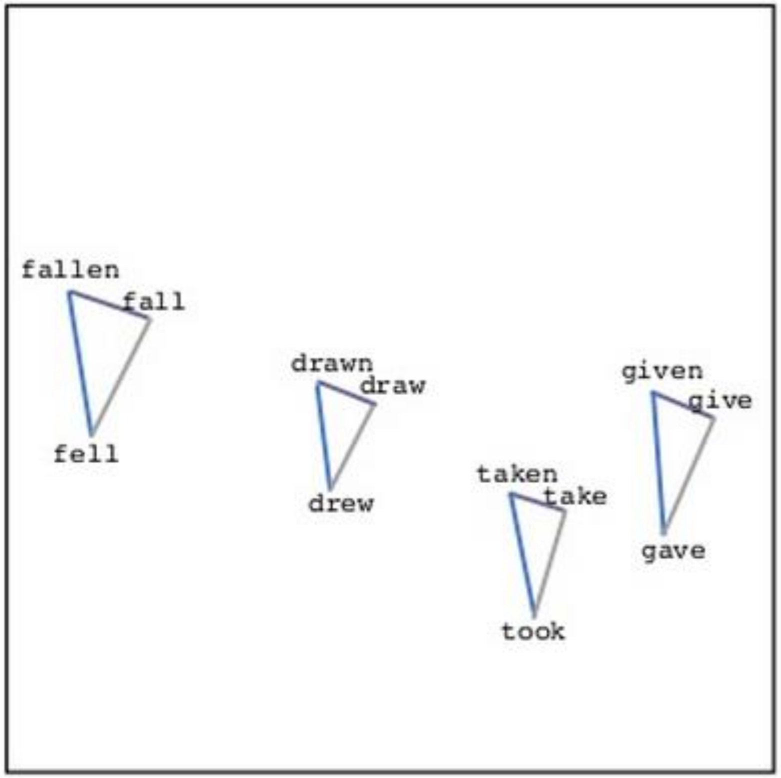
- Skip-gram



predicting the context given a word

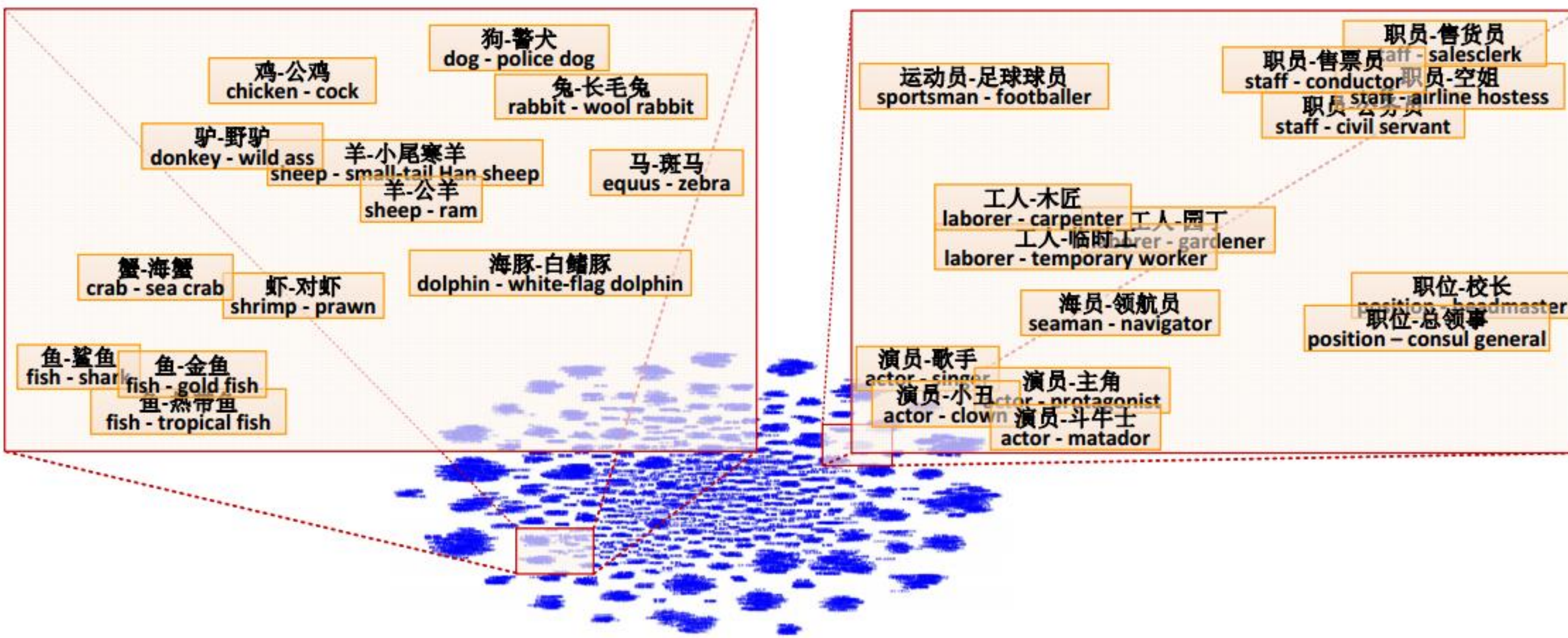
Word Embedding





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

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.

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 : ?

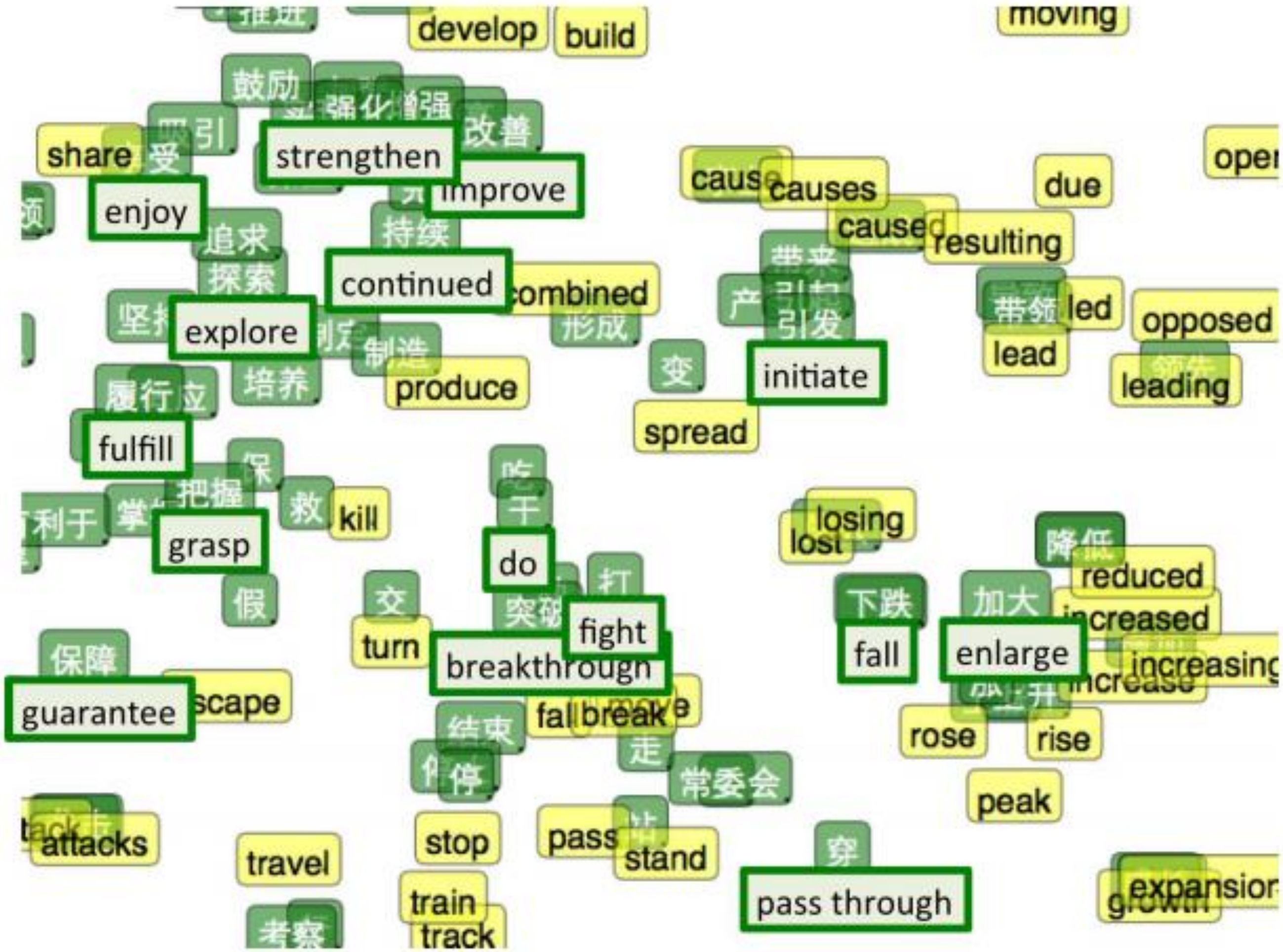
Compute $\underline{V(Berlin) - V(Rome) + V(Italy)}$

Find the word w with the closest V(w)

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)

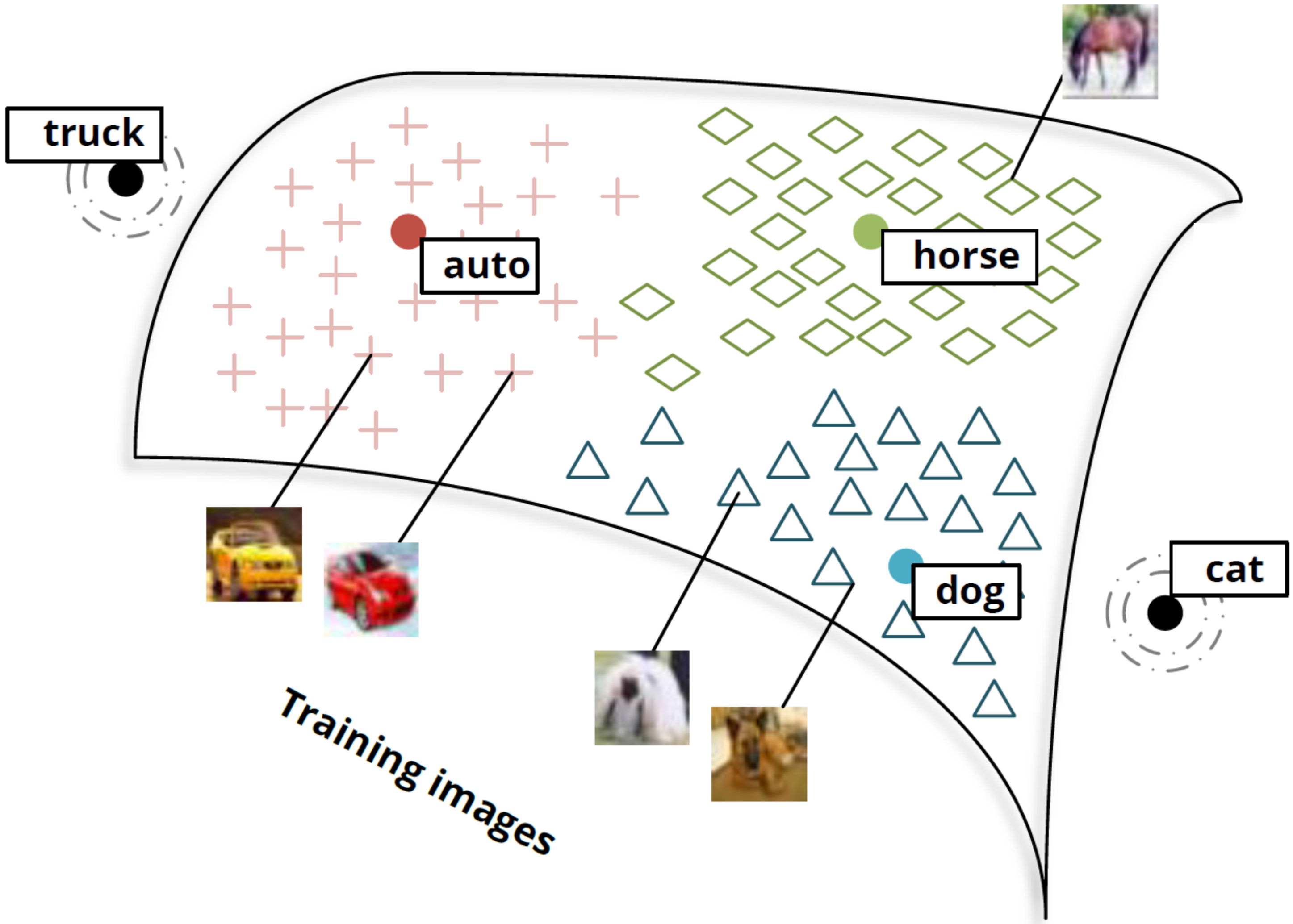
Multi-lingual Embedding



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

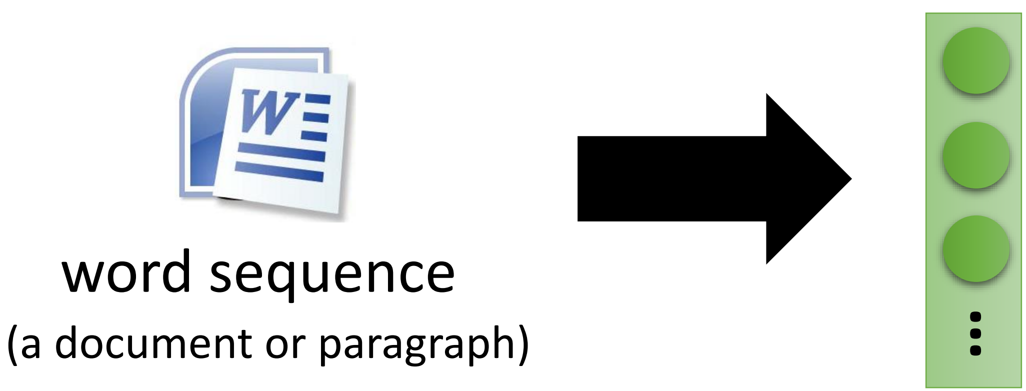
Multi-domain Embedding

Richard Socher, Milind Ganjoo, Hamsa Sridhar, Osbert Bastani, Christopher D. Manning, Andrew Y. Ng, Zero-Shot Learning Through Cross-Modal Transfer, NIPS, 2013

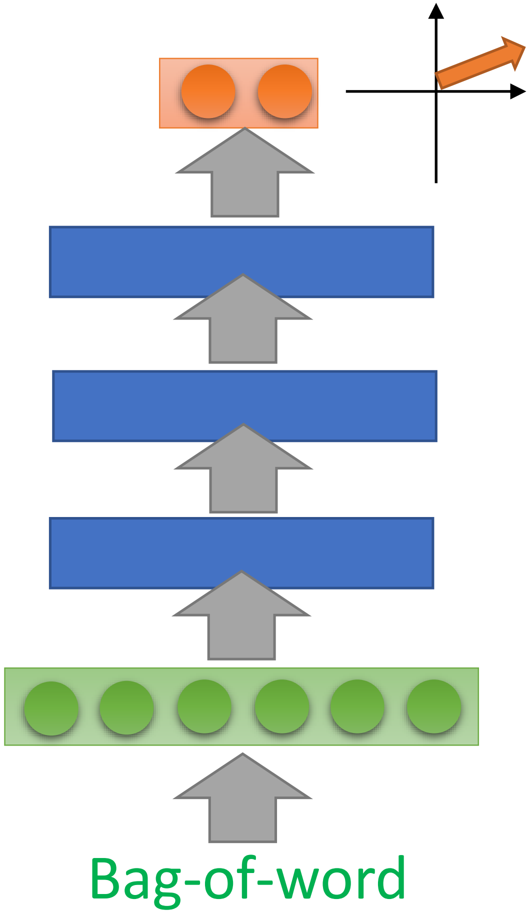


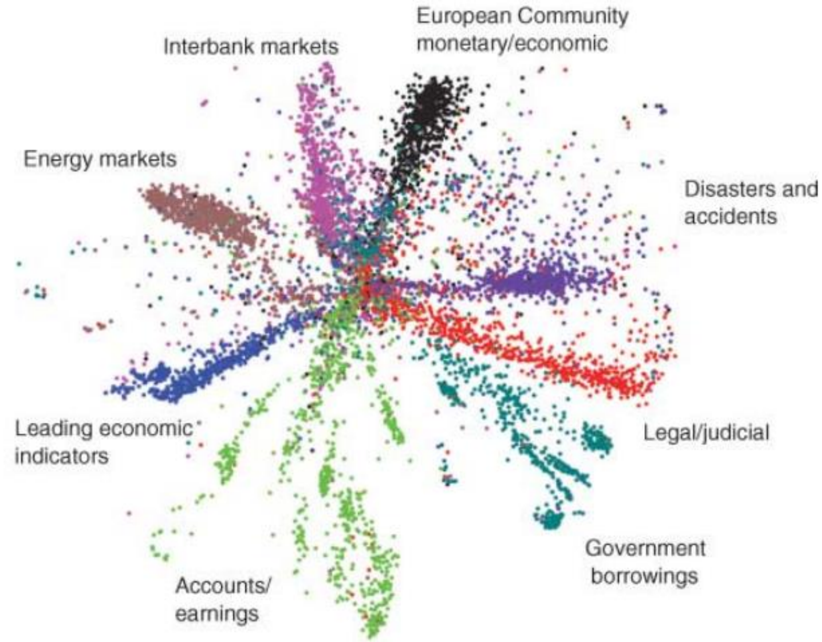
Document Embedding

- 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



Semantic Embedding

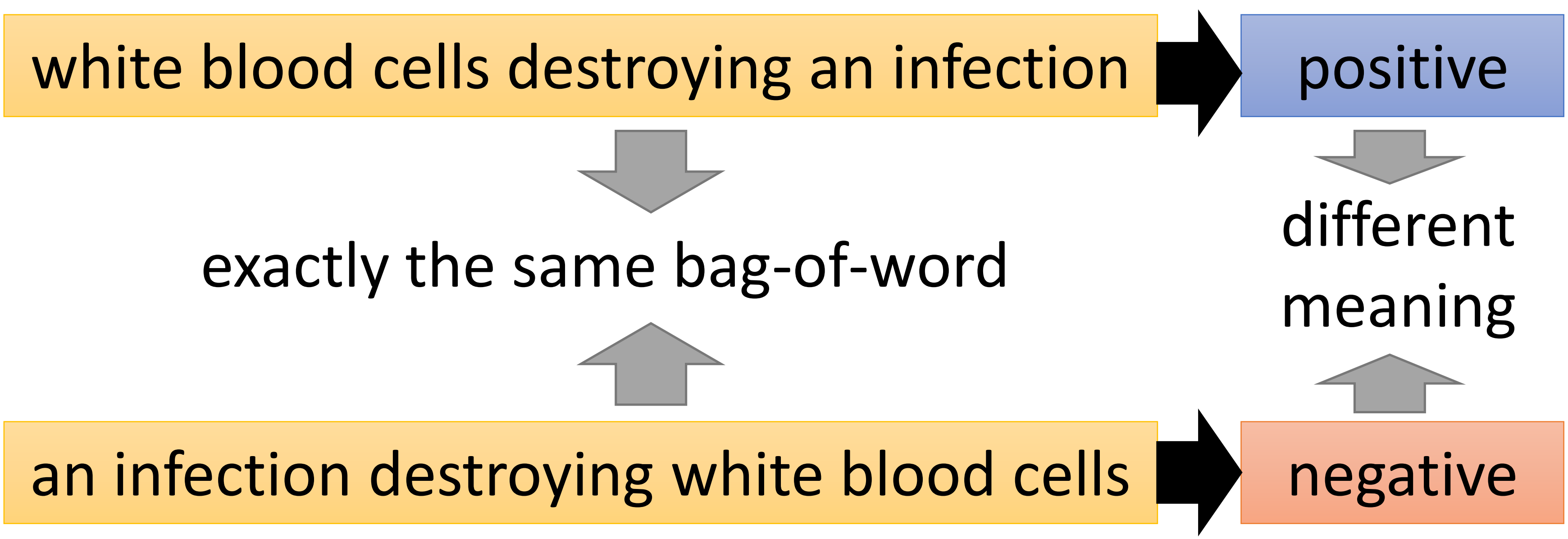




Reference: Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks." *Science* 313.5786 (2006): 504-507

Beyond Bag of Word

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



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.
- Exploiting other kind of labels:
  - Huang, Po-Sen, et al. "Learning deep structured semantic models for web search using clickthrough data." ACM, 2013.
  - Shen, Yelong, et al. "A latent semantic model with convolutional-pooling structure for information retrieval." ACM, 2014.
  - Socher, Richard, et al. "Recursive deep models for semantic compositionality over a sentiment treebank." EMNLP, 2013.
  - Tai, Kai Sheng, Richard Socher, and Christopher D. Manning. "Improved semantic representations from tree-structured long short-term memory networks." arXiv preprint, 2015.