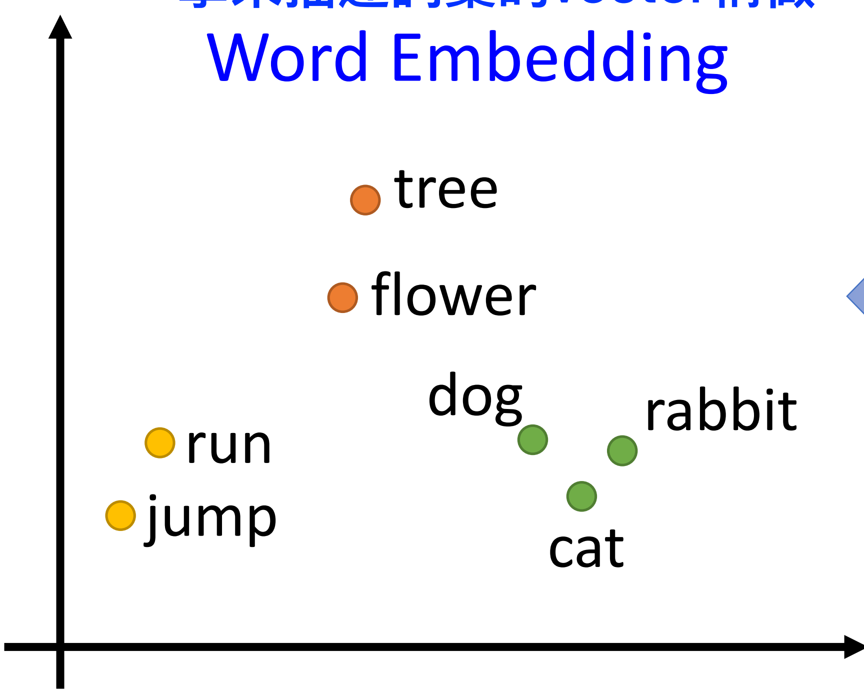
Unsupervised Learning: Word Embedding

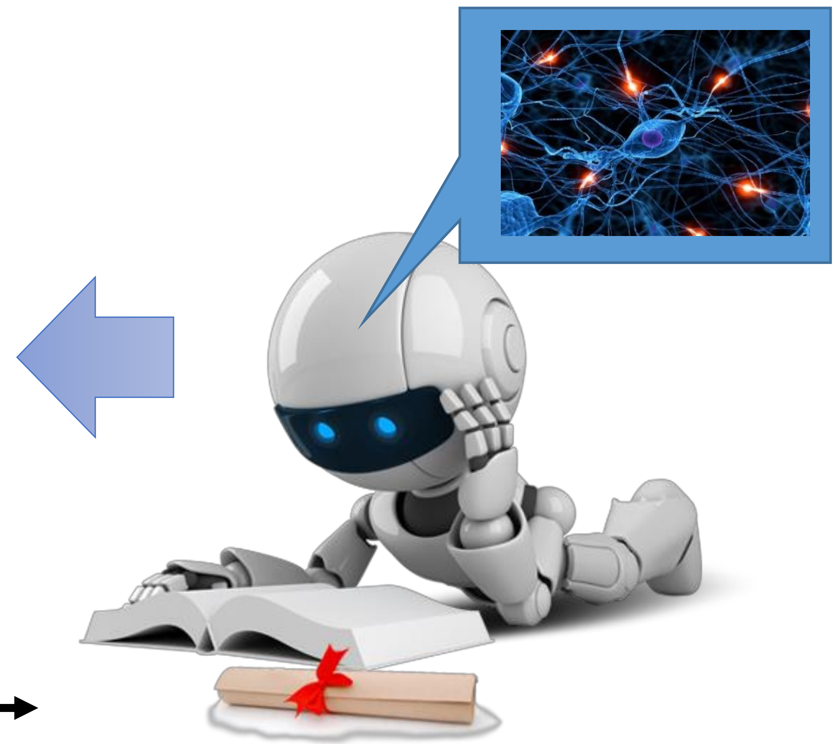
特别用在文字上的dimension reduction

Word Embedding

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

拿來描述詞彙的vector稱做
Word Embedding





用lexicon size的vector來描述一個詞彙

1-of-N Encoding

apple = [1 0 0 0 0]

bag = [0 1 0 0 0]

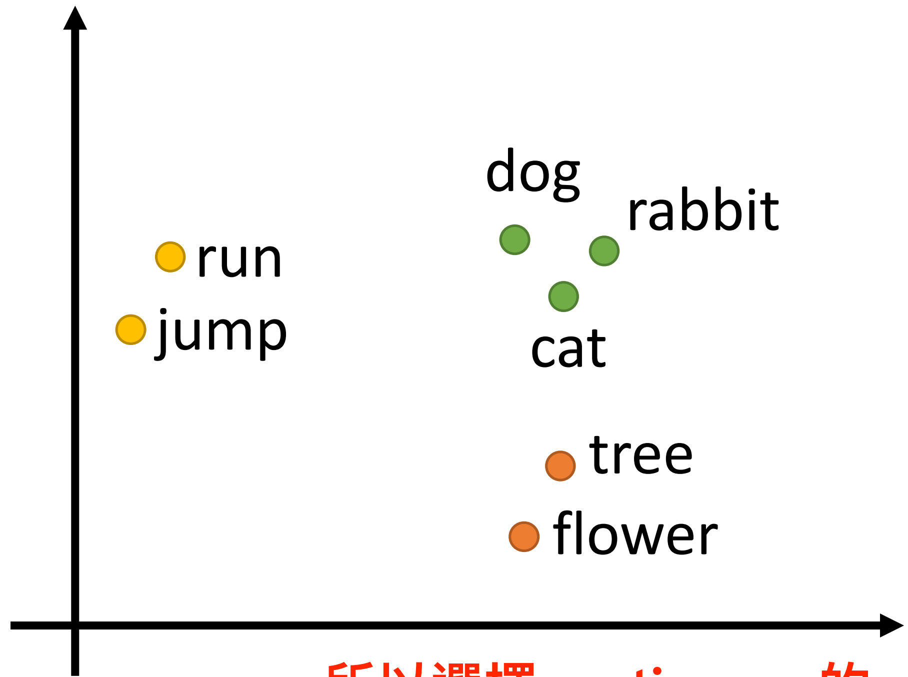
cat = [0 0 1 0 0]

dog = [0 0 0 1 0]

elephant = [0 0 0 0 1]

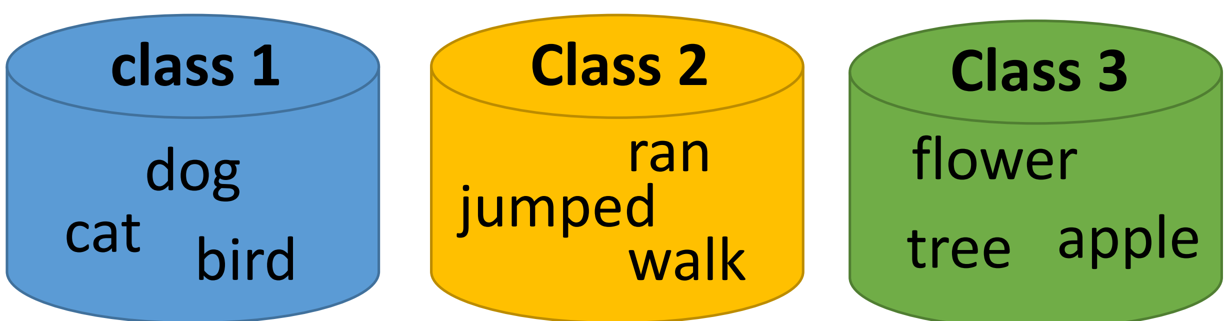
壞處是詞彙間的關係無法藉由vector傳遞出來

Word Embedding



所以選擇continuous的vector (word embedding)

Word Class



因此我們可以先做clustering，但這樣的分類方式太粗糙

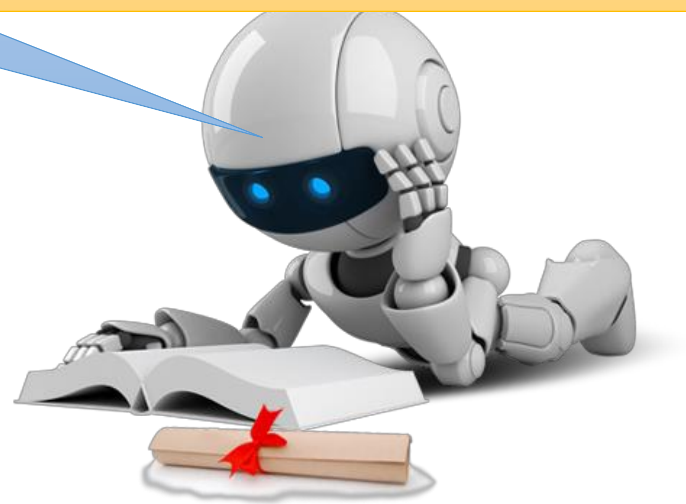
Word Embedding

想法：每一個word皆可以用上下文來判斷其語意

- 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

有類似的content，可能vector某個dimension會一樣

蔡英文 520宣誓就職

How to exploit the context?

- **Count based** 原則：若wi,wj常常一起出現，則他們的vector繼算相似度應該要很接近
  - 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/

越接近越好

$V(w_i) . V(w_j)$ ⟷ $N_{i,j}$ N(i,j): 一起出現的次數

Inner product

Number of times w_i and w_j in the same document

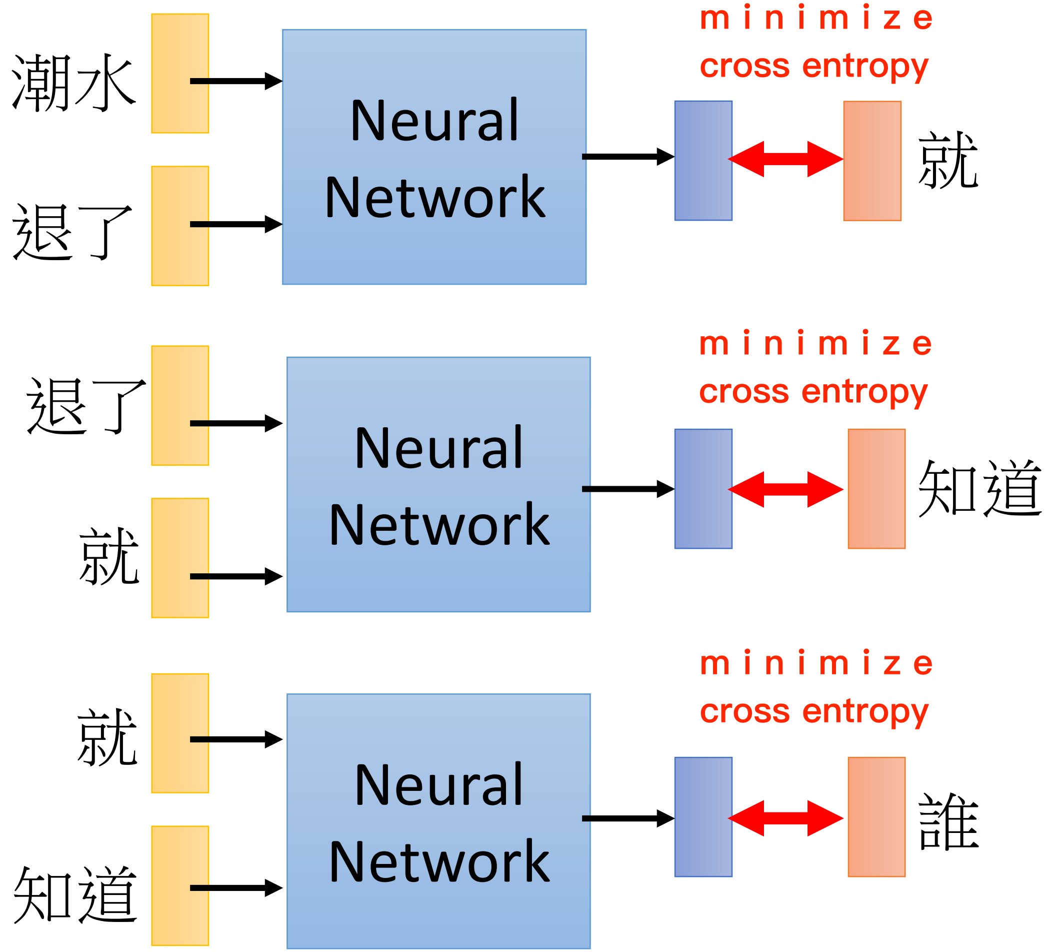
- **Perdition based**

Prediction-based – Training

首先讓machine predict接下來出現的詞彙

Collect data: 首先要先做斷詞

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)

– 機器翻譯
– 語音辨識

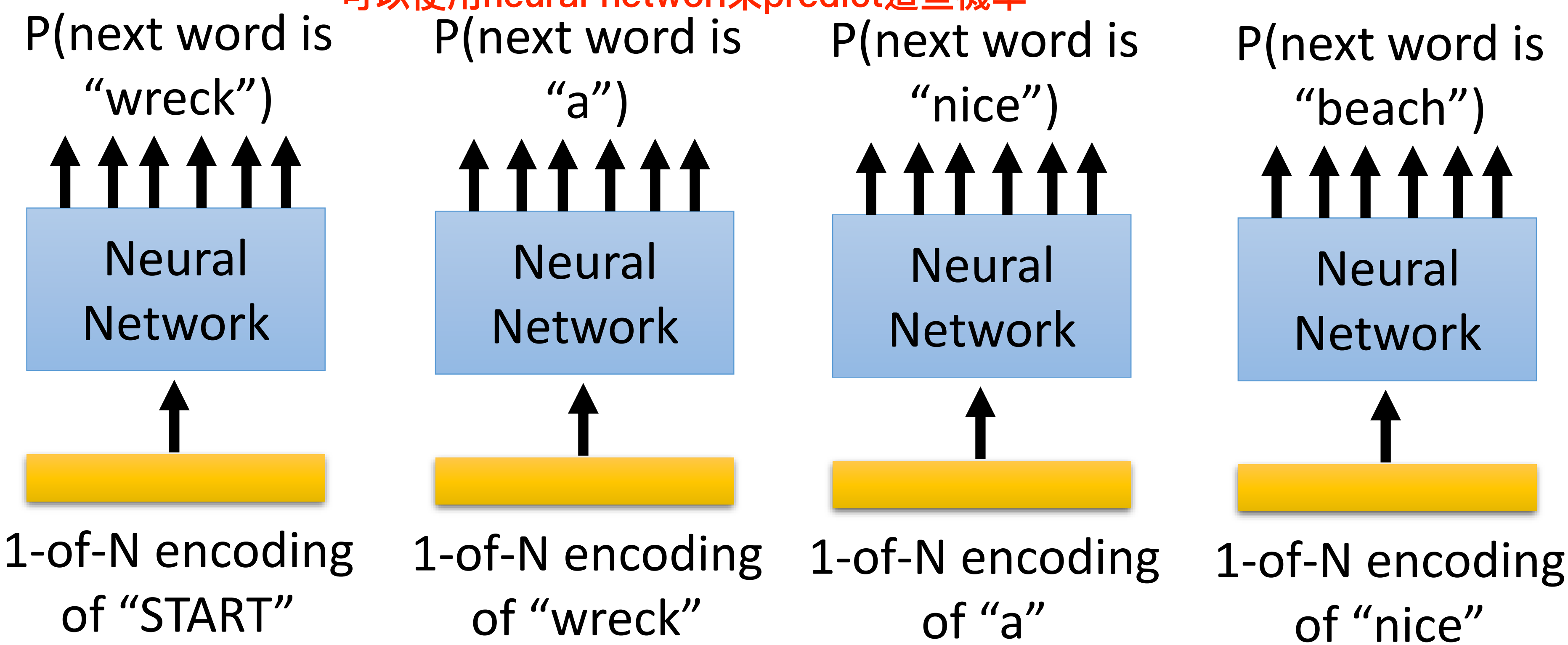
Prediction-based
– Language Modeling

P(“wreck a nice beach”) 即使搜集了大量文章再去算這個句子出現的機率，仍然有機會等於零

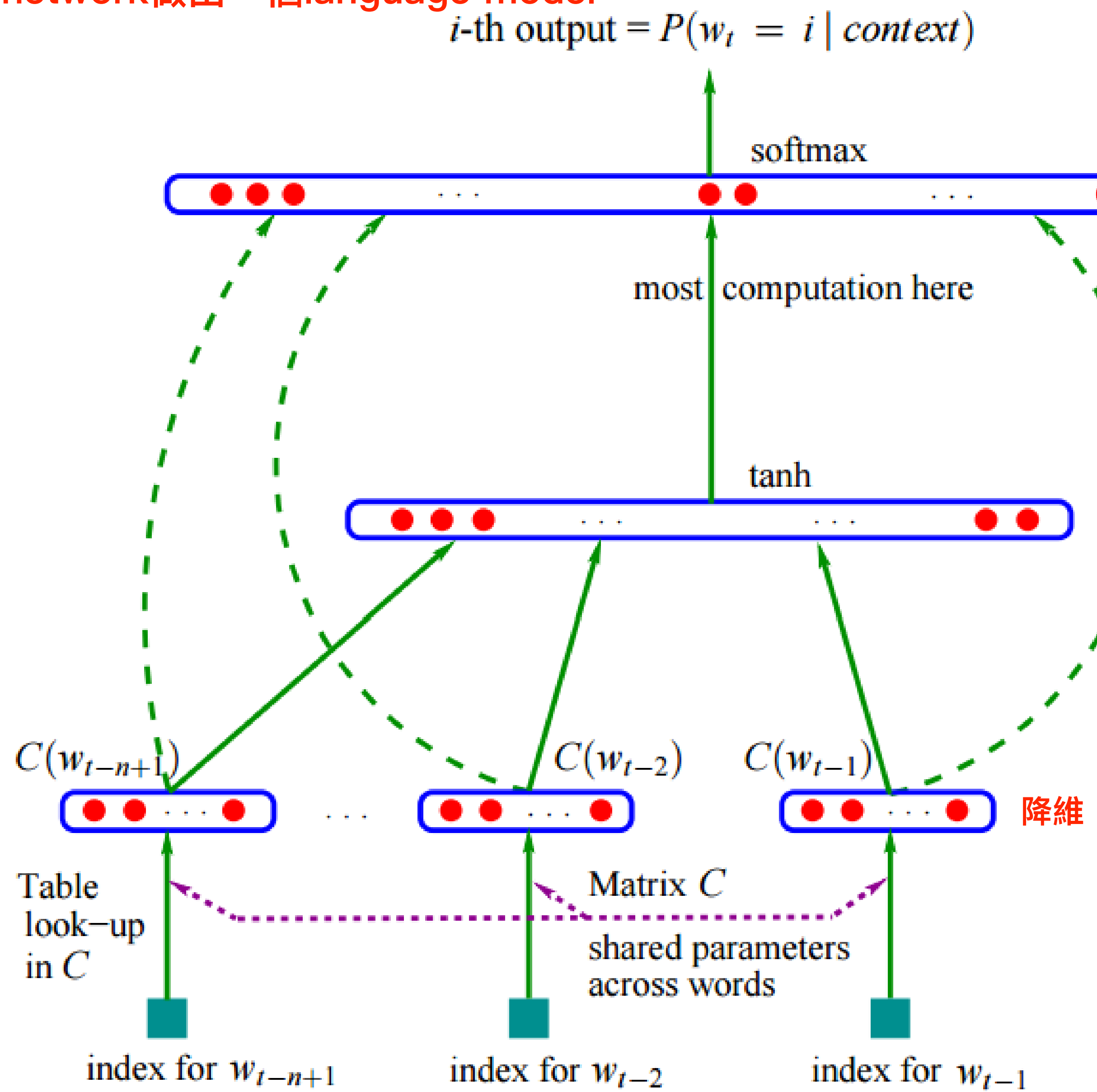
=P(wreck|START)P(a|wreck)P(nice|a)P(beach|nice)

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

可以使用neural networl來predict這些機率



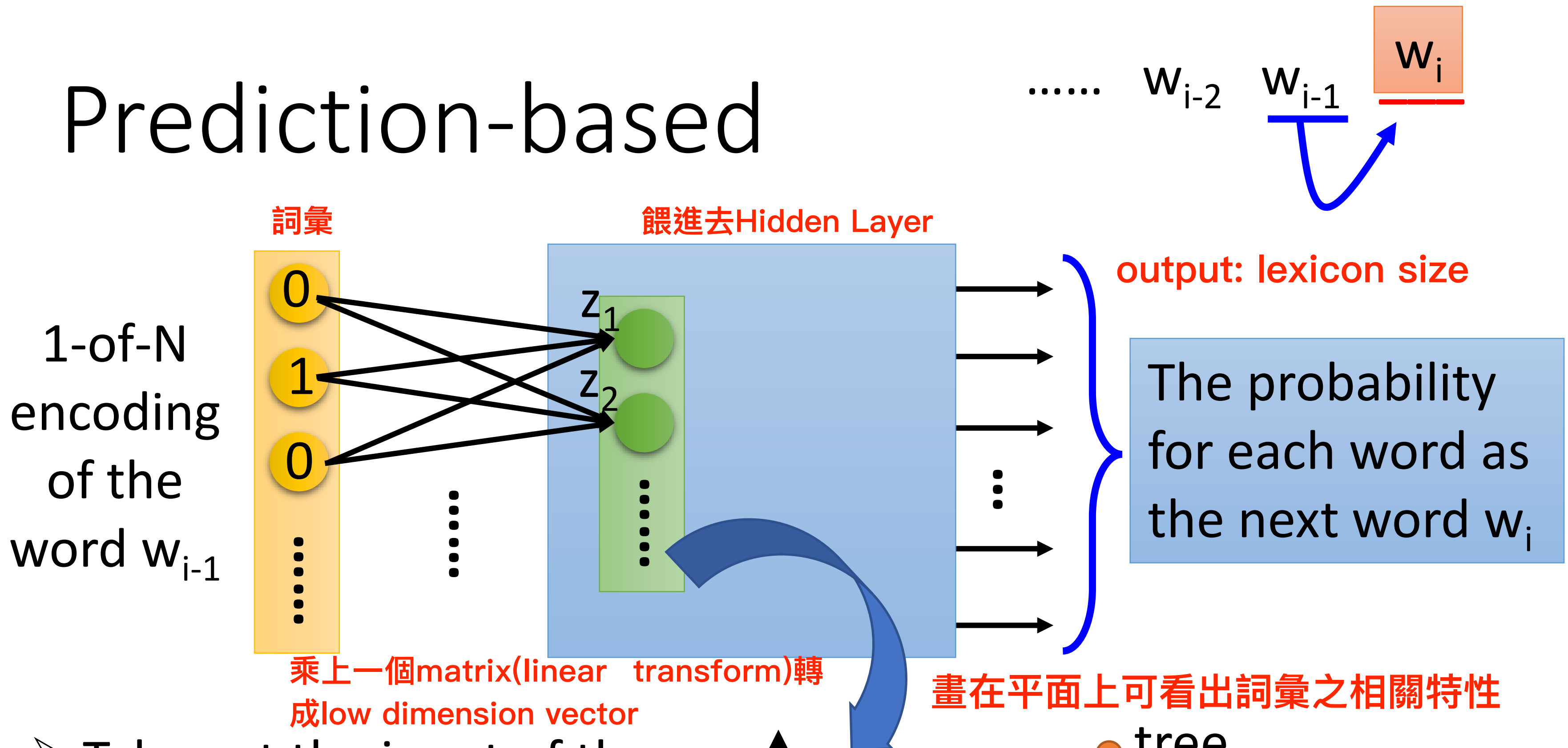
利用neural network做出一個language model



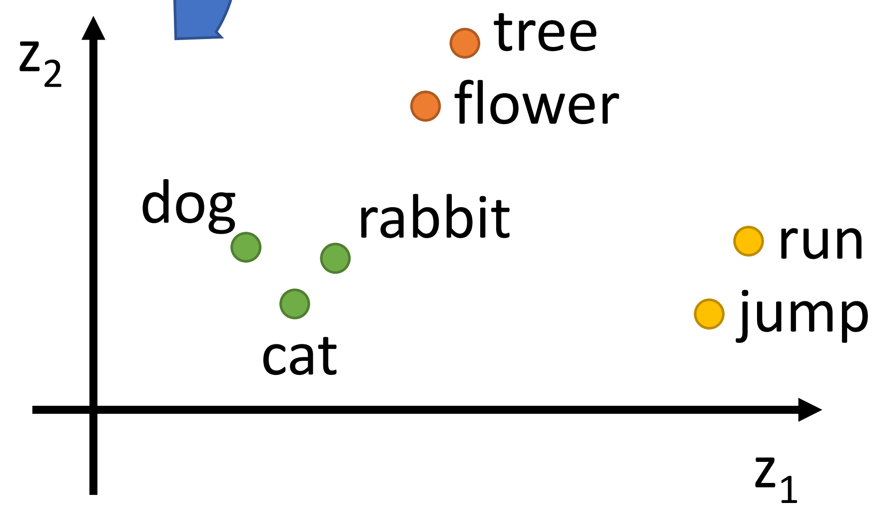
Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, *3*(Feb), 1137-1155.

第一篇用NN解language model

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)

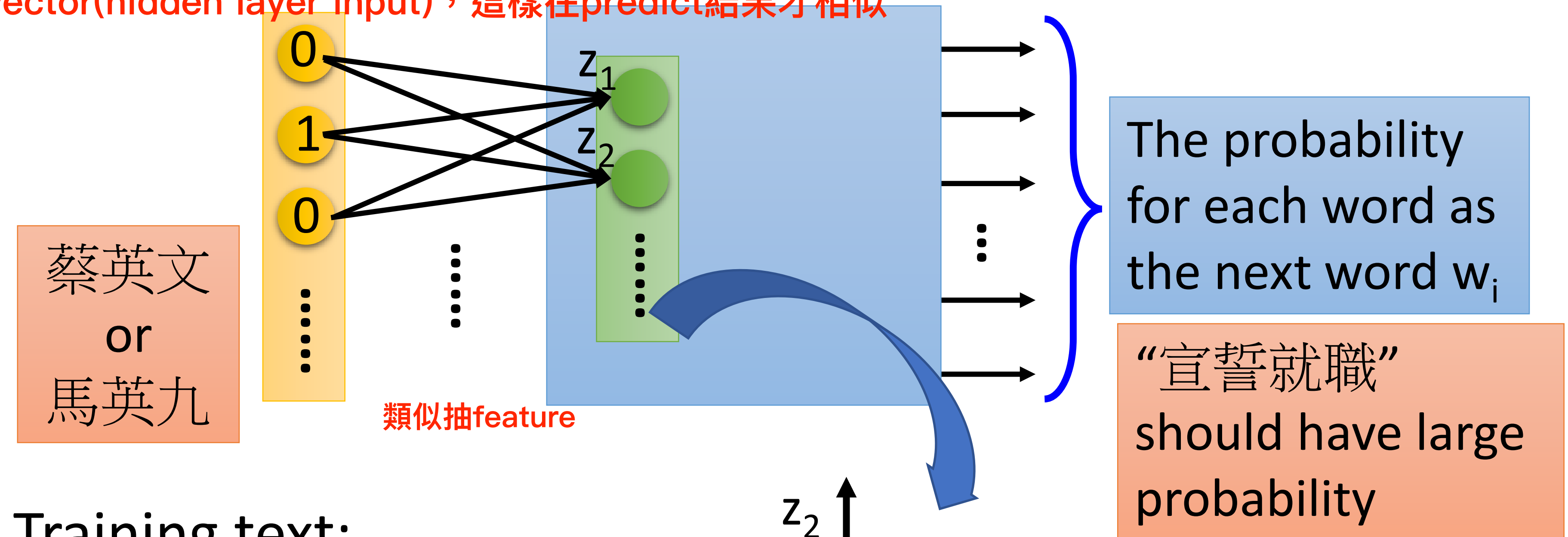


在做word embedding與其做一個deep的model，不如用一個shallow的linear model，類似抽feature（PCA）

Prediction-based

You shall know a word by the company it keeps

將不同的input做transform後能夠得到相似的vector(hidden layer input)，這樣在predict結果才相似



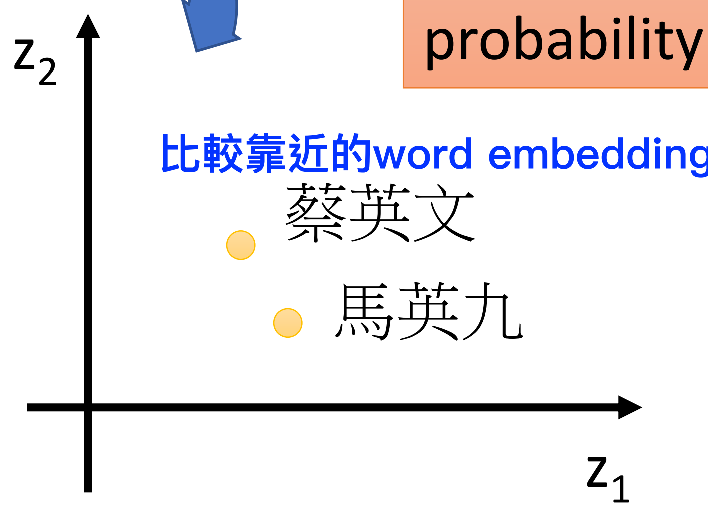
Training text:

...... 蔡英文 宣誓就職 ......

w_{i-1} w_i

...... 馬英九 宣誓就職 ......

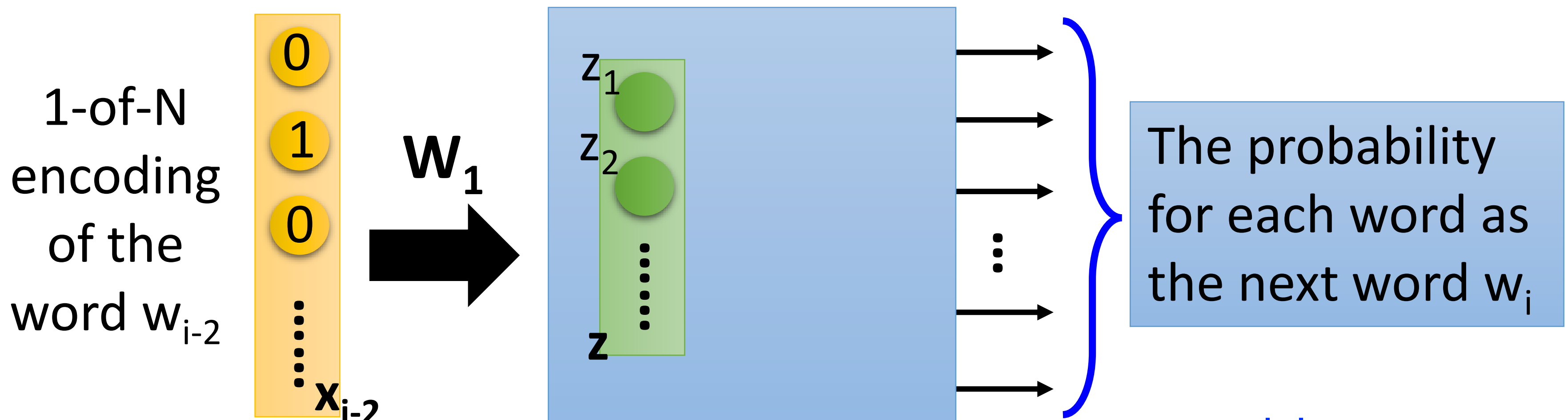
w_{i-1} w_i

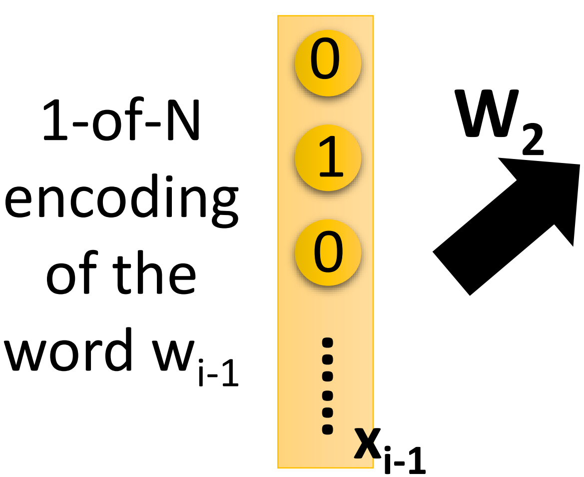


不同的詞彙如果有同樣的attribute，在做word embedding，他們在某幾個dimension會有接近的值

Prediction-based – Sharing Parameters

如同在做CNN的時候，用同個filter做convolution即為共用參數考慮相同的部分





|V| = lexicon size

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

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

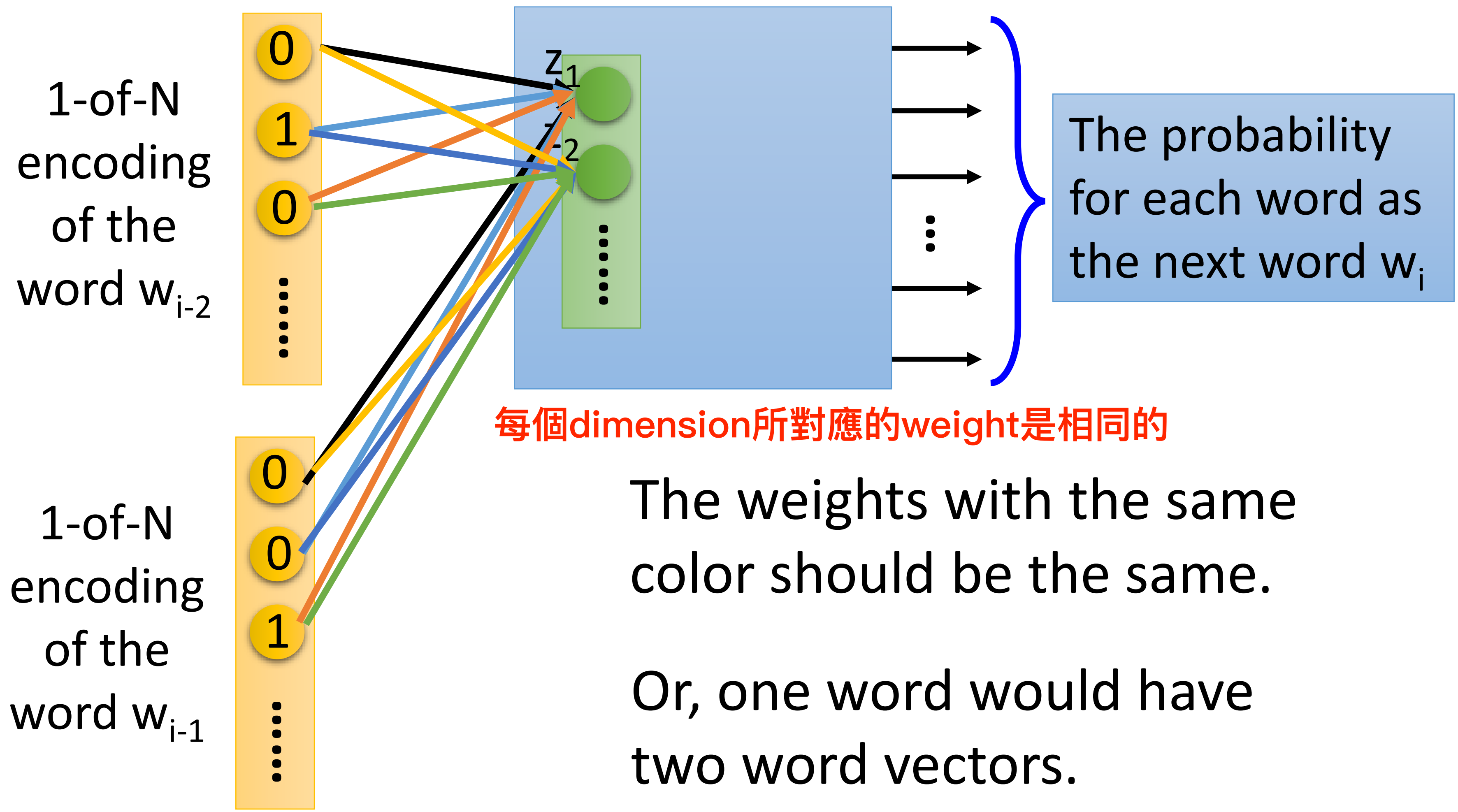
$$\mathbf{z} = \mathbf{W}_1 \mathbf{x}_{i-2} + \mathbf{W}_2 \mathbf{x}_{i-1}$$

The weight matrix $\mathbf{W}_1$ and $\mathbf{W}_2$ are both |Z|X|V| matrices. 將w1,w2 share 參數

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

考慮長一點的word才能有好的結果(至少10~20) 共用參數，使得vector dimension不會增加

Prediction-based – Sharing Parameters



每個dimension所對應的weight是相同的

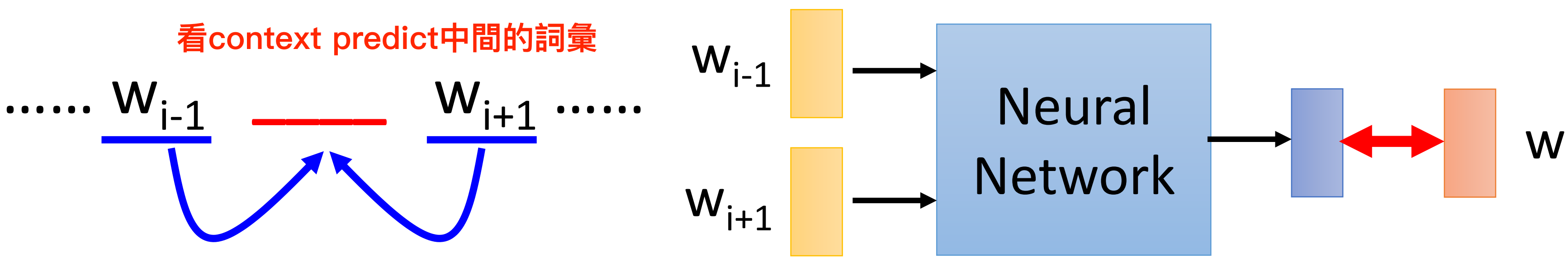
The weights with the same color should be the same.

Or, one word would have two word vectors.

由於將參數tight在一起，因此同個詞彙放在不同位置得到的結果是相同的

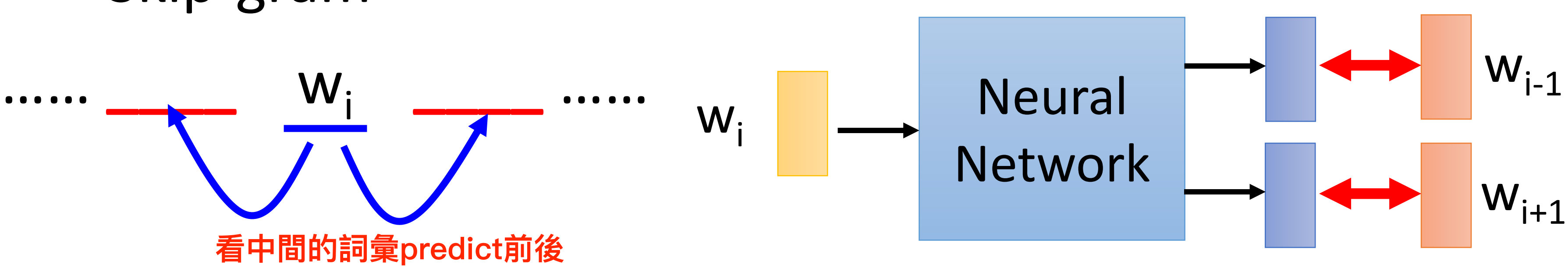
Prediction-based
– Various Architectures

- Continuous bag of word (CBOW) model



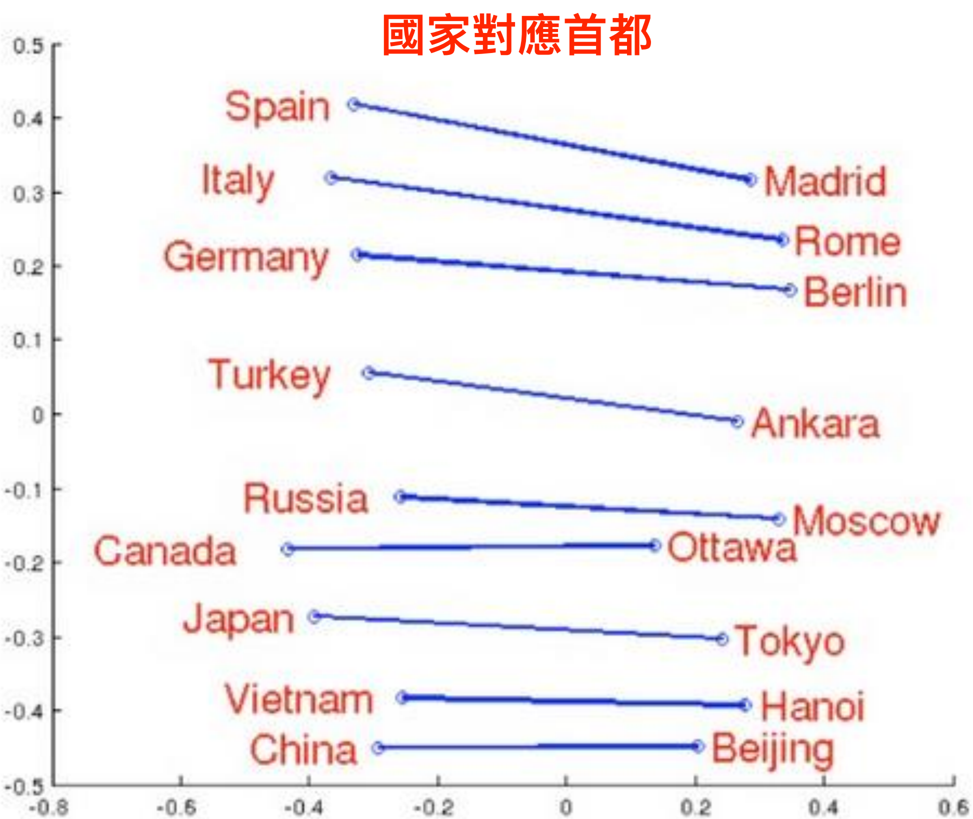
predicting the word given its context

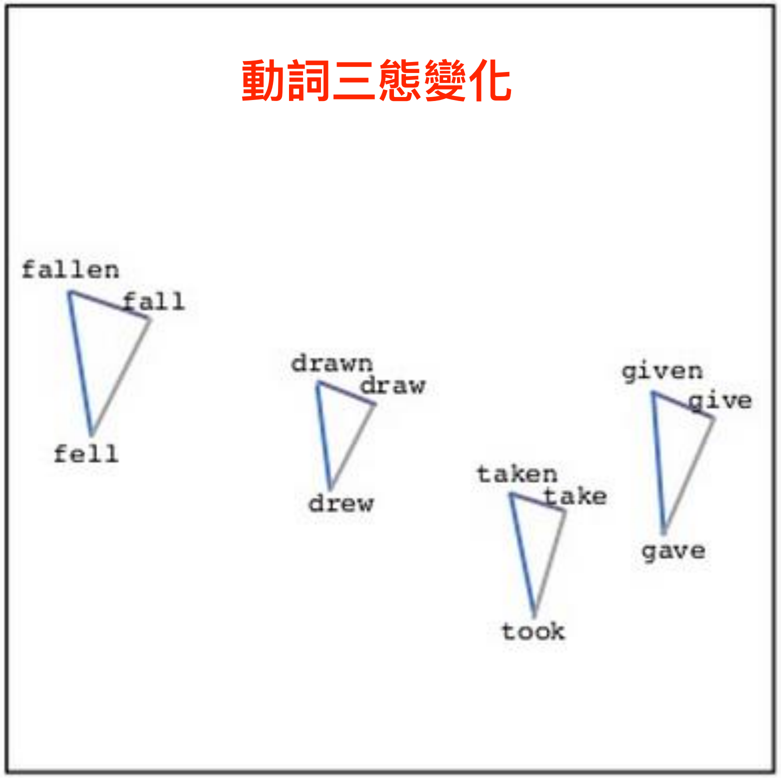
- Skip-gram



predicting the context given a word

Word Embedding

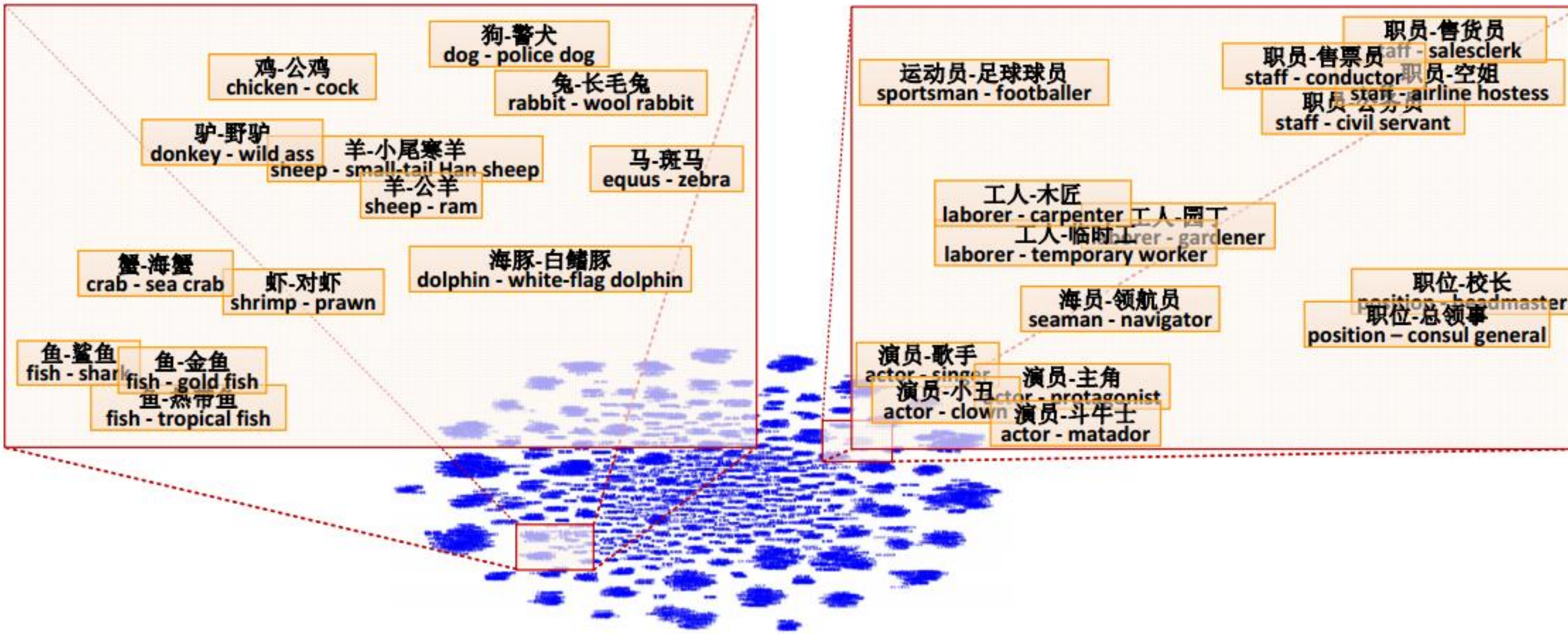




Source: http://www.slideshare.net/hustwj/cikm-keynotenov2014

Word Embedding

找出詞彙間的從屬關係



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

羅馬之於意大利=柏林之於？

Rome : Italy = Berlin : ?

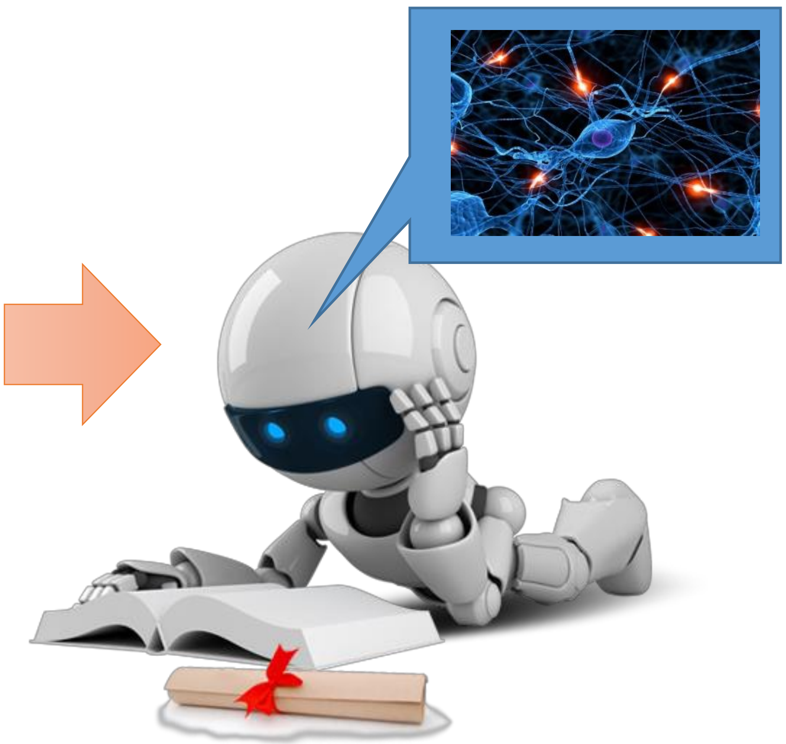
Compute $\underline{V(Berlin) - V(Rome) + V(Italy)}$

Find the word w with the closest V(w)

算出vector後找最相近的vector

Demo

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



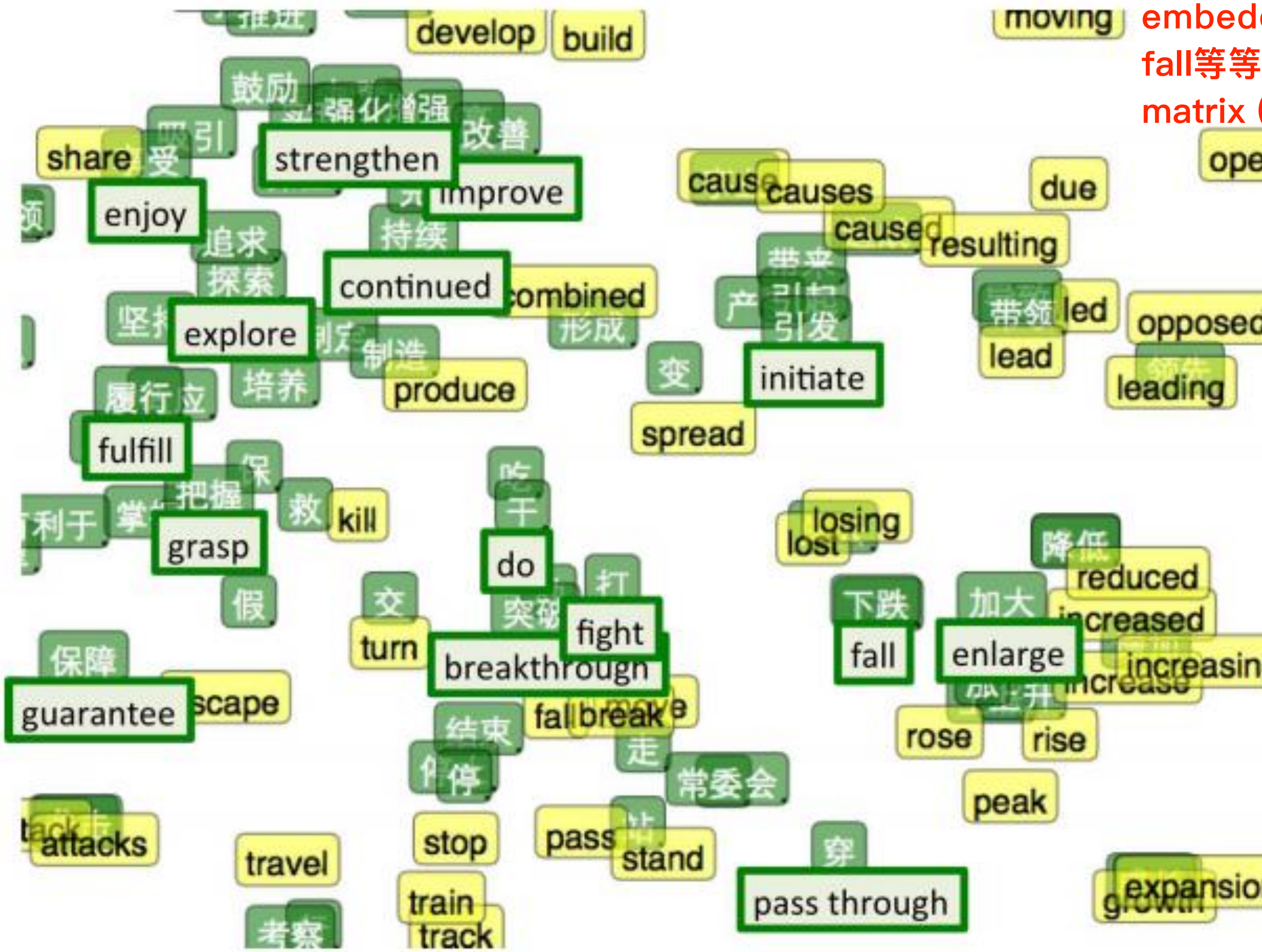
Demo

- 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)

word embedding是unsupervised的，因此我們並不清楚每個dimension代表什麼意思

中文跟英文無法一起train，舉例來說中文的第三維可能是動物英文的第七維才是，因此會亂掉
因此要先有一些已經可以相互對應的word embedding，如 下跌v.s. fall等等，找出transform的matrix (mapping)

Multi-lingual Embedding



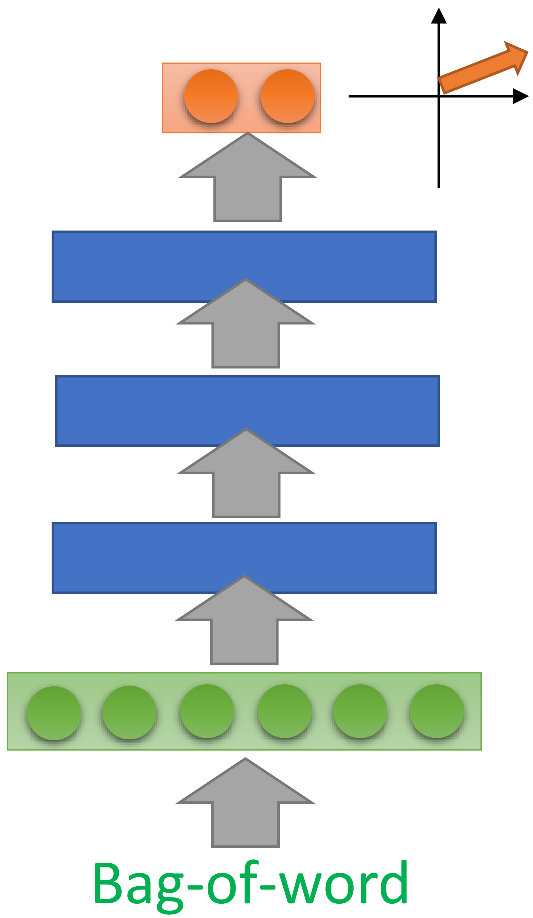
Bilingual Word Embeddings for Phrase-Based Machine Translation, Will Zou, Richard Socher, Daniel Cer and Christopher Manning, EMNLP, 2013

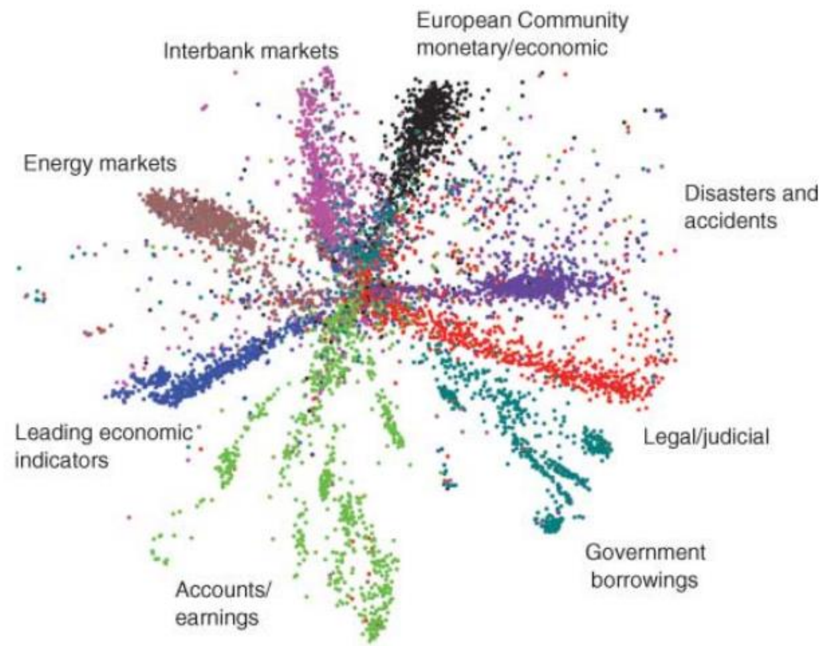
Document Embedding

- word sequences with different lengths → the vector with the same length
  - The vector representing the meaning of the word sequence 不同文章的長度不同
  - A word sequence can be a document or a paragraph

把文章對應至一個vector

Semantic Embedding





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

先將詞彙統計成Bag of Word，然後利用auto-encoder降維至二維平面

Beyond Bag of Word

需要用到RNN

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

即使Bag of Word相同，但是詞彙順序不同會影響其語意

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

- ***<u>Paragraph Vector</u>***: Le, Quoc, and Tomas Mikolov. "Distributed Representations of Sentences and Documents." ICML, 2014
- ***<u>Seq2seq Auto-encoder</u>***: Li, Jiwei, Minh-Thang Luong, and Dan Jurafsky. "A hierarchical neural autoencoder for paragraphs and documents." arXiv preprint, 2015
- ***<u>Skip Thought</u>***: Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, Sanja Fidler, “Skip-Thought Vectors” arXiv preprint, 2015.