Do machines know the meaning of a word?

Hung-yi Lee

Language Technology

spam detection

(http://spam-filter-review.toptenreviews.com/)

Part-of-speech Tagging

John saw the saw.

↓ ↓ ↓ ↓

PN V D N

Name Entity Recognition

這 位 是 李 宏 毅

Name of People

Sentiment Analysis

這部電影太糟了

Negative (負雷)

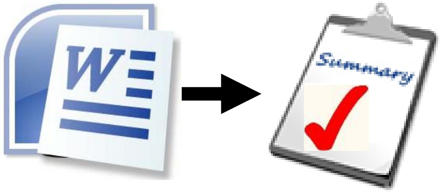
Translation

“Machine learning ……”

⬍

“機器學習 ……”

Summarization



document summary

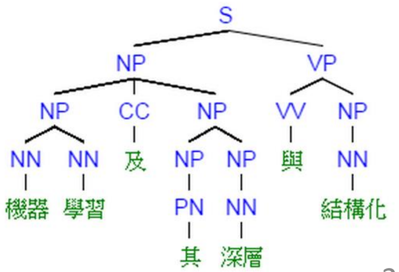
Retrieval

Google
台灣

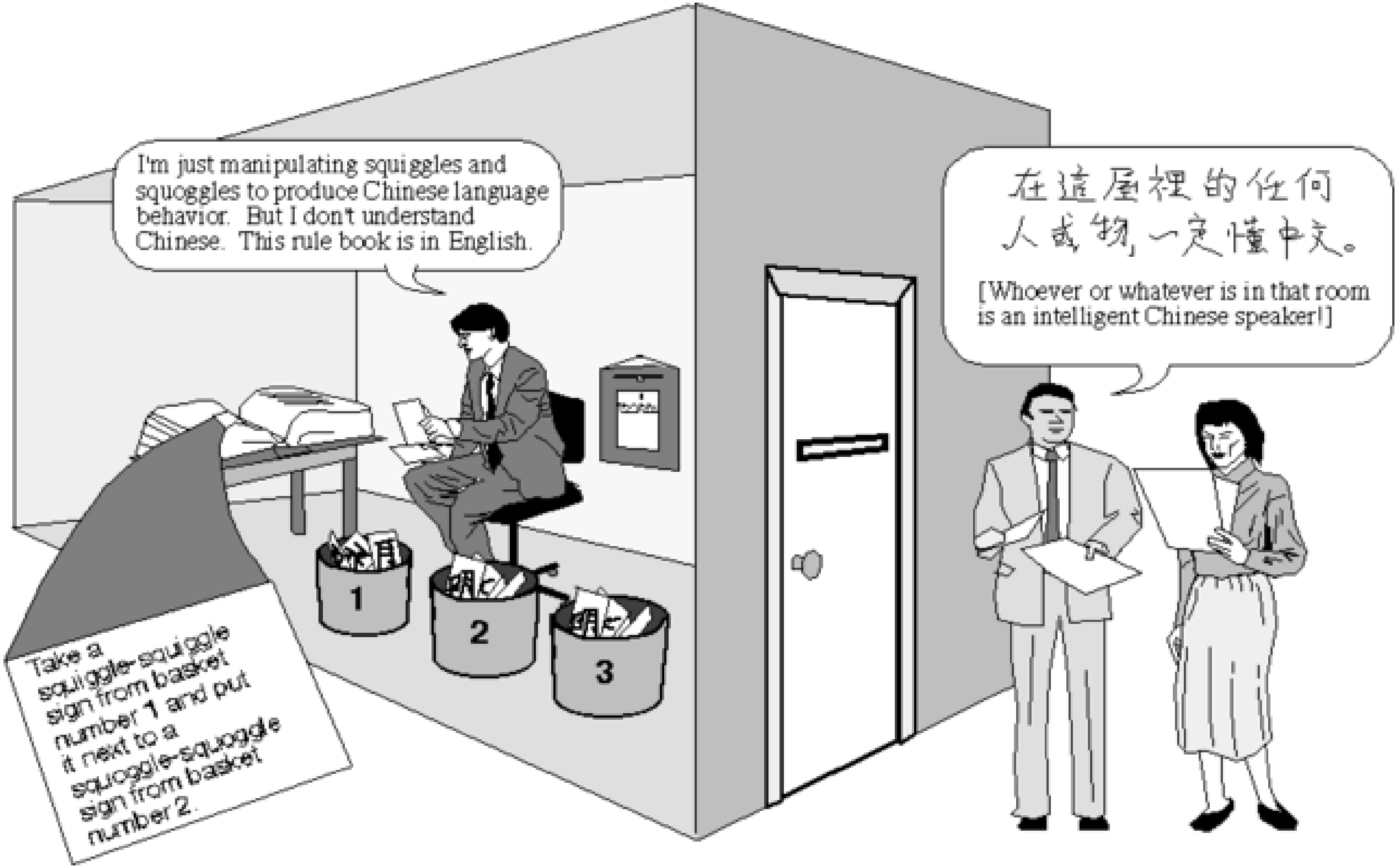
Speech Recognition

大家好……

Syntactic Analysis



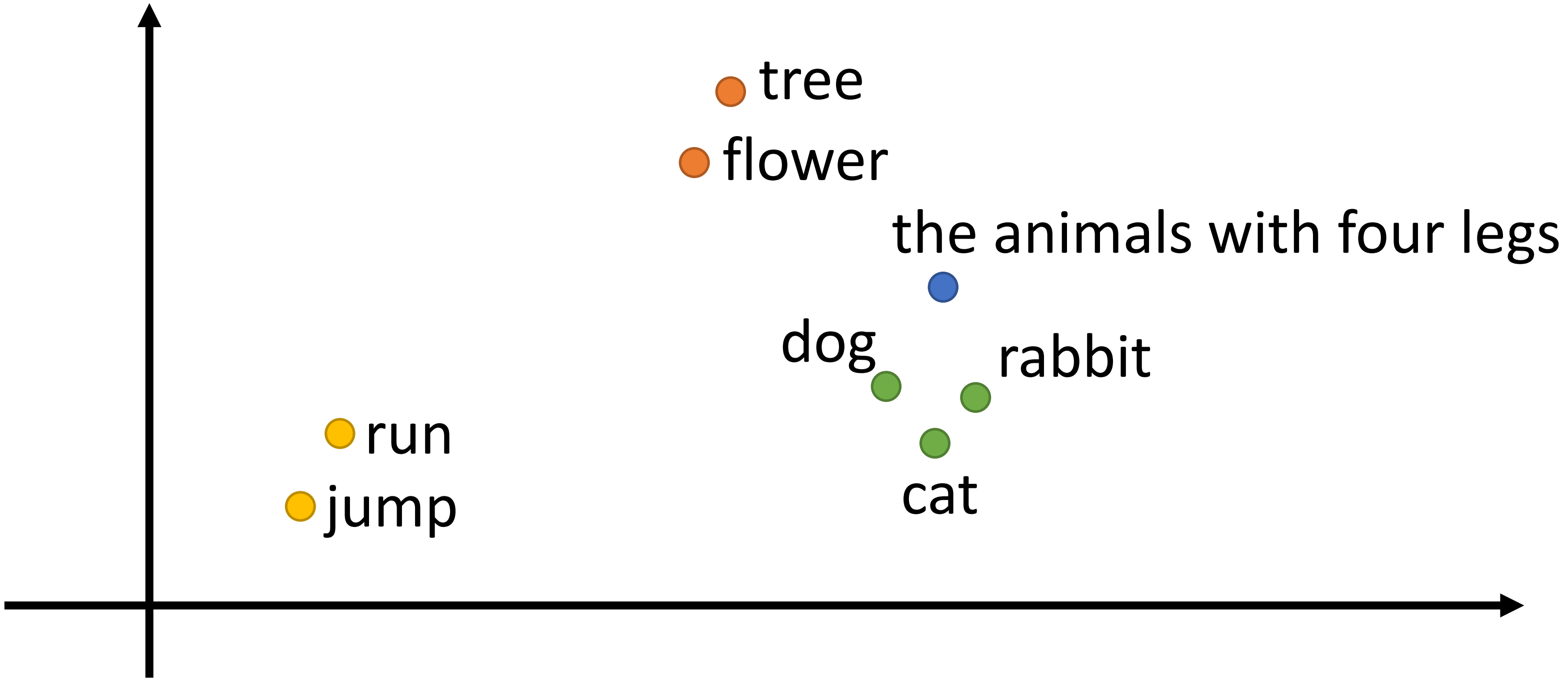
Do machine really understand human language?


I'm just manipulating squiggles and squoggles to produce Chinese language behavior. But I don't understand Chinese. This rule book is in English.
在這屋裡的任何人或物，一定懂中文。
[Whoever or whatever is in that room is an intelligent Chinese speaker!]
Take a squiggle-squiggle sign from basket number 1 and put it next to a squoggle-squoggle sign from basket number 2.
1
2
3

http://cse3521.artifice.cc/chinese-room.html

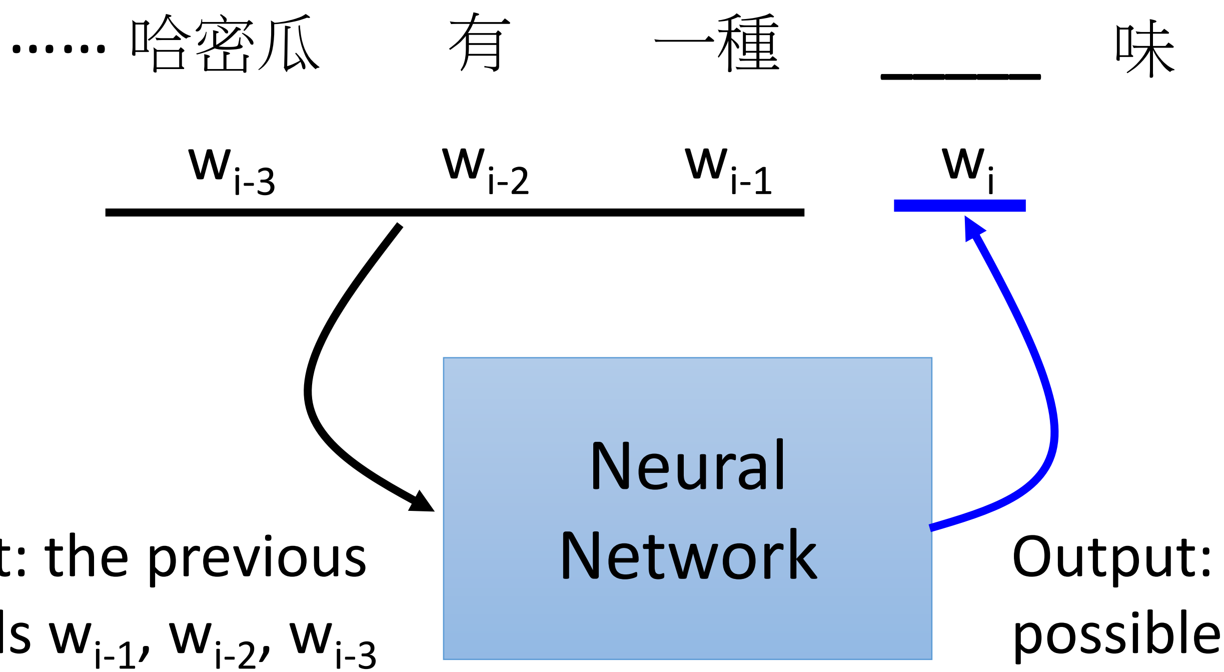
Meaning Representation

Do machine know the meaning of a word or word sequence?



Meaning of Word

Fill in the Blank



Input: the previous words w_{i-1}, w_{i-2}, w_{i-3}

Output: the most possible next word w_i

Each word should be represented as a feature vector.

Fill in the Blank

1-of-N Encoding

lexicon = {apple, bag, cat, dog, elephant}

apple = [1 0 0 0 0]

bag = [0 1 0 0 0]

cat = [0 0 1 0 0]

dog = [0 0 0 1 0]

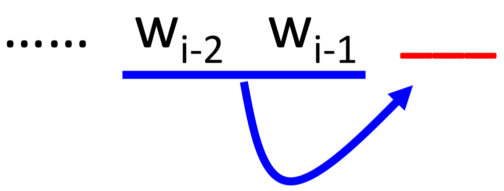
elephant = [0 0 0 0 1]

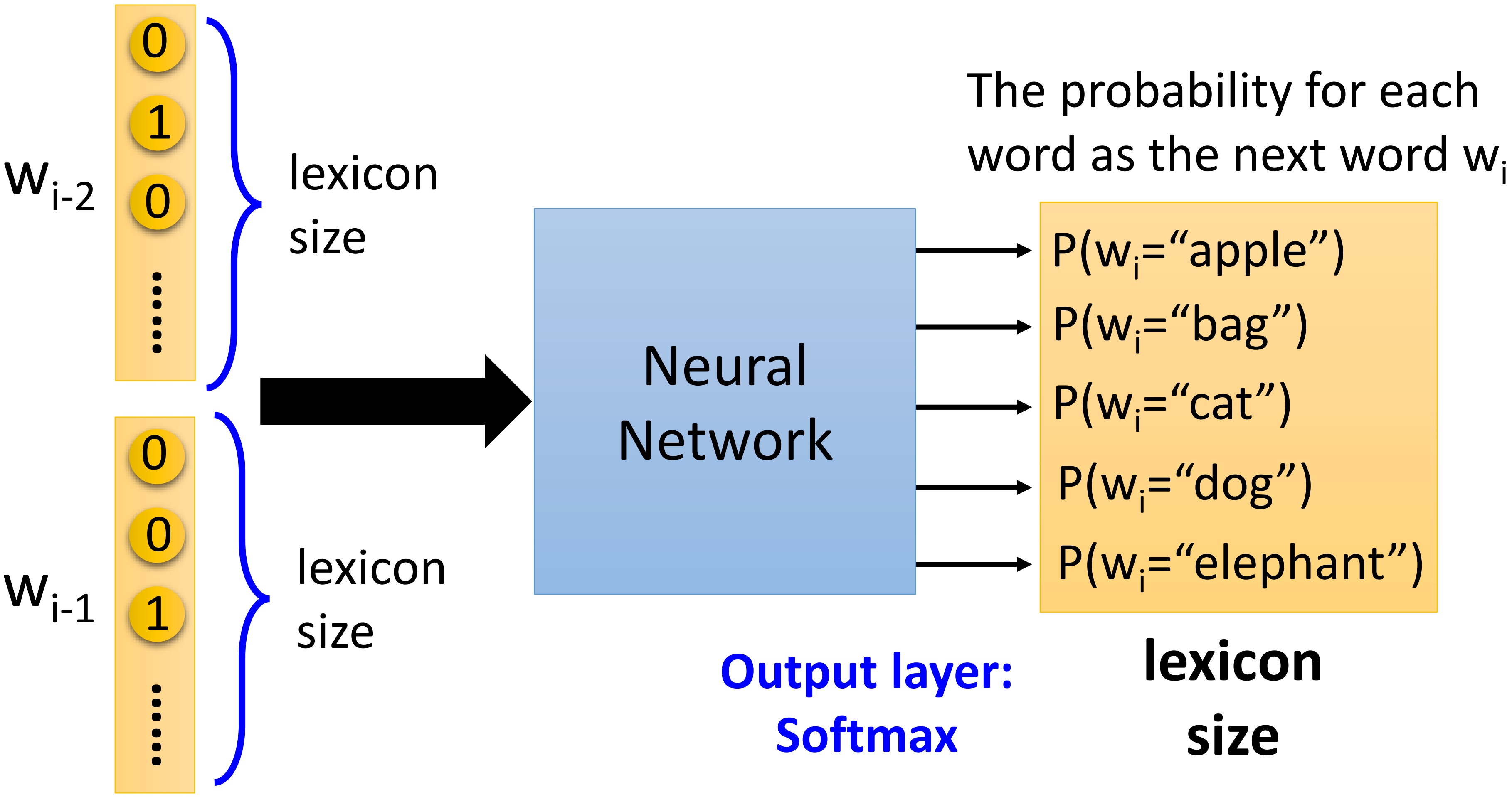
The vector is lexicon size.

Each dimension corresponds to a word in the lexicon

The dimension for the word is 1, and others are 0

Fill in the Blank





lexicon size

lexicon size

Neural Network

The probability for each word as the next word w_i

P(w_i=“apple”)

P(w_i=“bag”)

P(w_i=“cat”)

P(w_i=“dog”)

P(w_i=“elephant”)

Output layer: Softmax

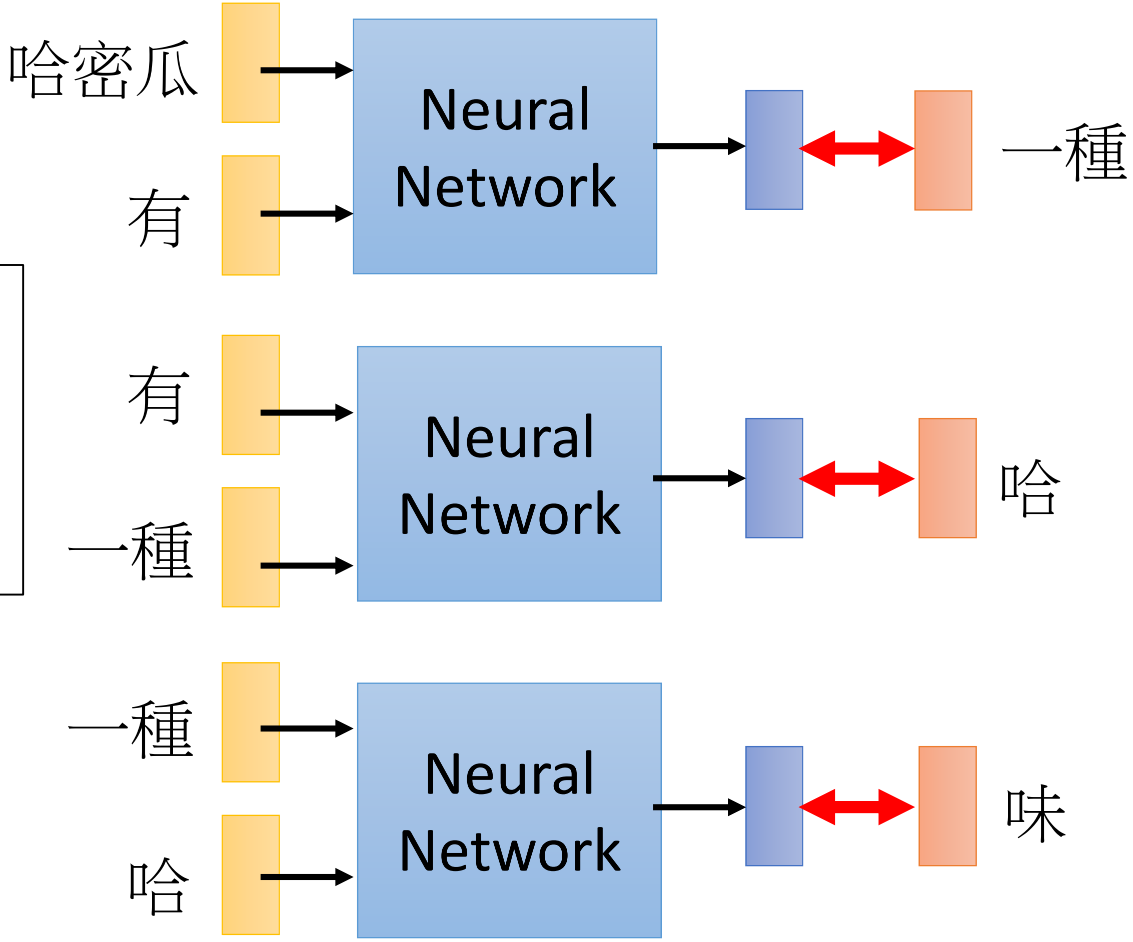
lexicon size

Fill in the Blank

- Training:

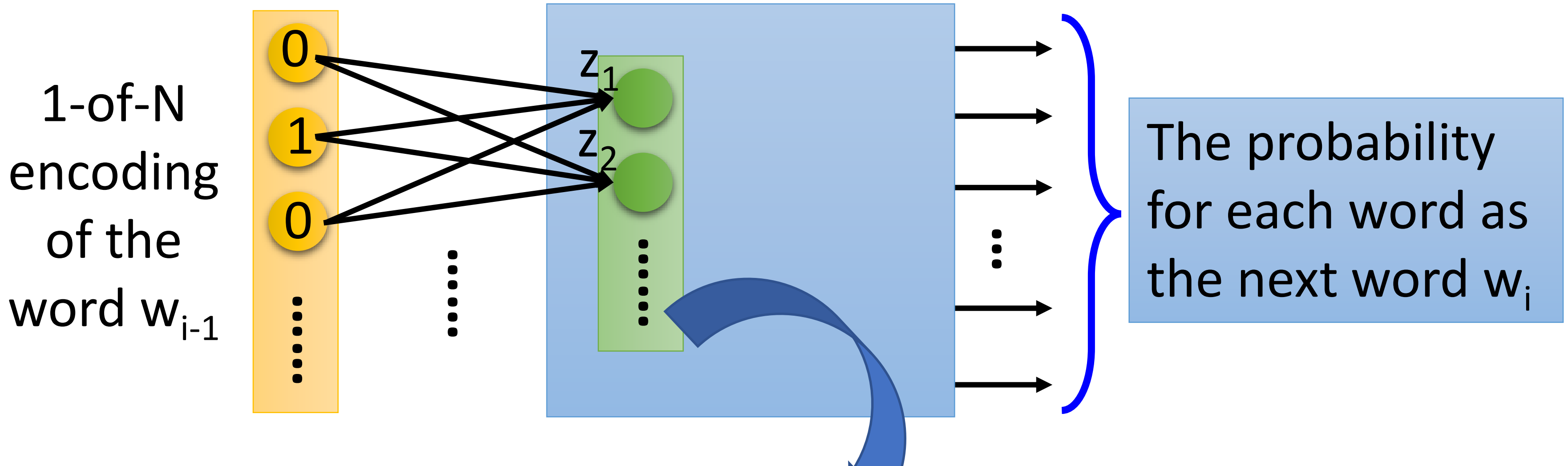
Collect data:

哈密瓜 有 一種 哈 味
不爽 不要 買
公道價 八萬 一
………

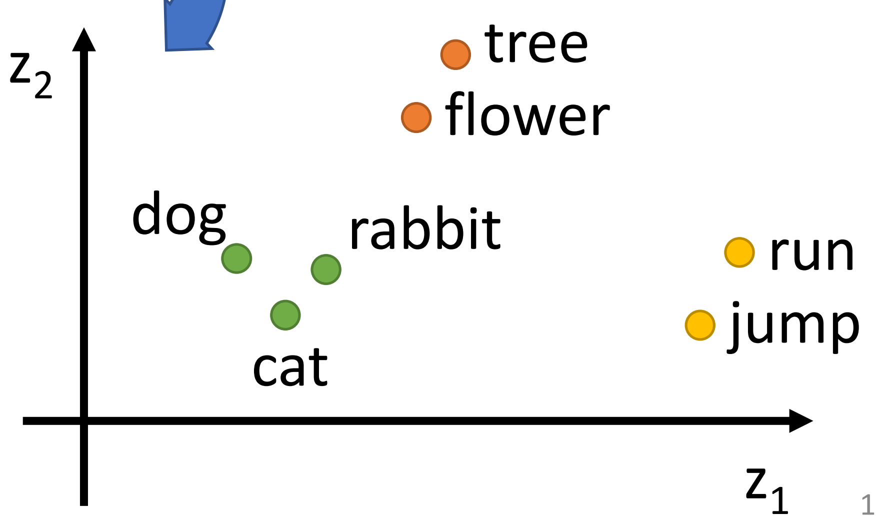


Minimizing cross entropy

Word Vector


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

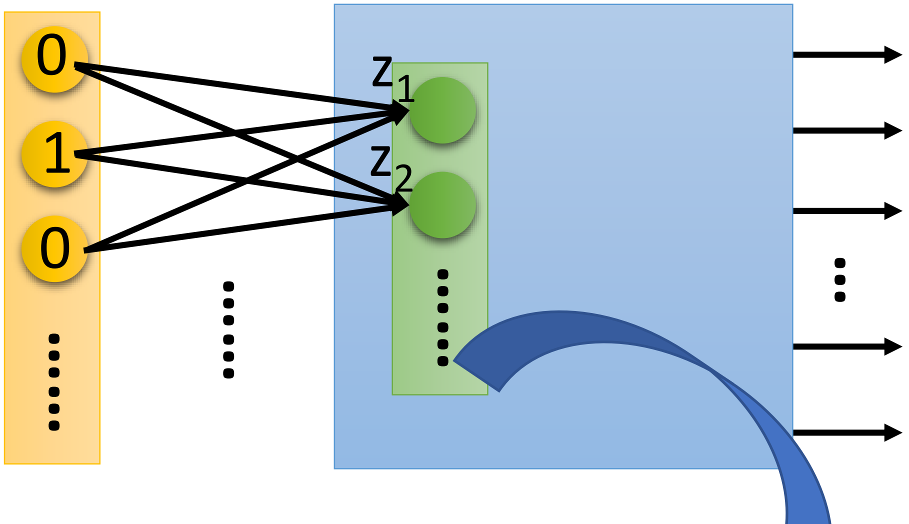
- ➢ Take out the input of the neurons in the first layer
- ➢ Use it to represent a word w
- ➢ Word vector, word embedding feature: V(w)



Word Vector

You shall know a word
by the company it keeps

李逍遙
or
酒劍仙



The probability for each word as the next word w_i

"御劍飛行"
should have large probability

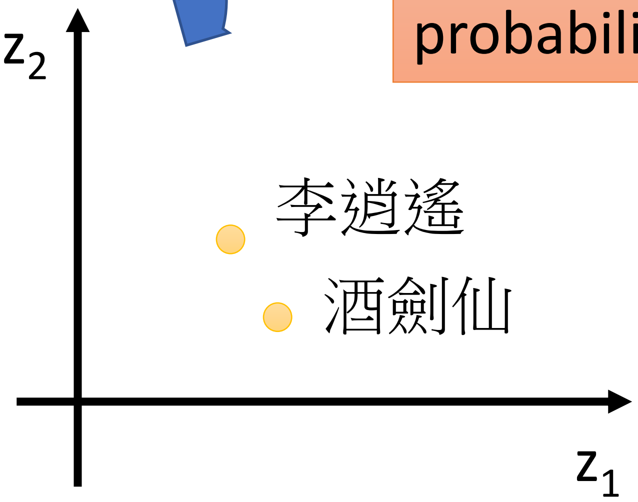
Training text:

…… 李逍遙 御劍飛行 ……

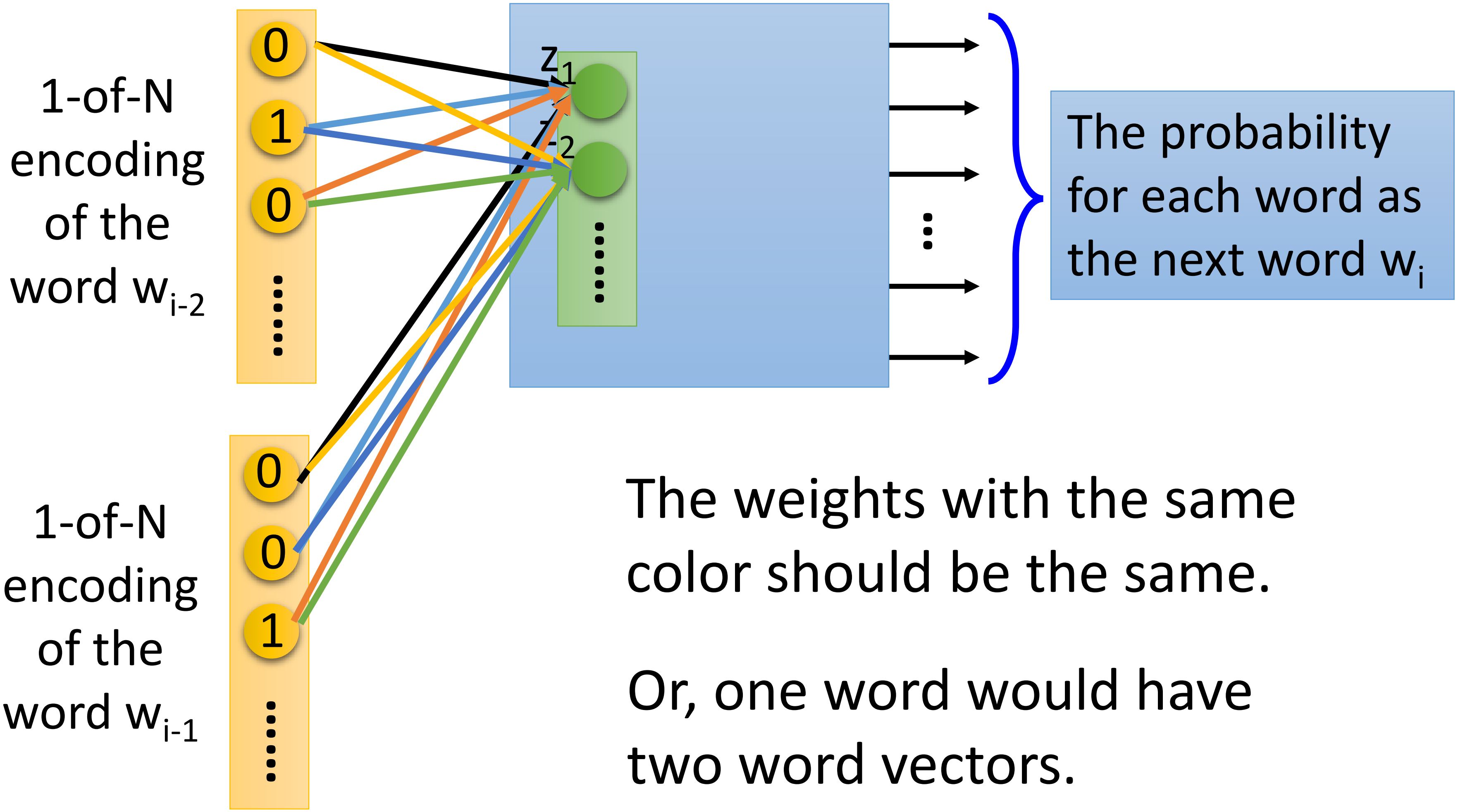
w_{i-1} w_i

…… 酒劍仙 御劍飛行 ……

w_{i-1} w_i



Word Vector – Sharing Parameters



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

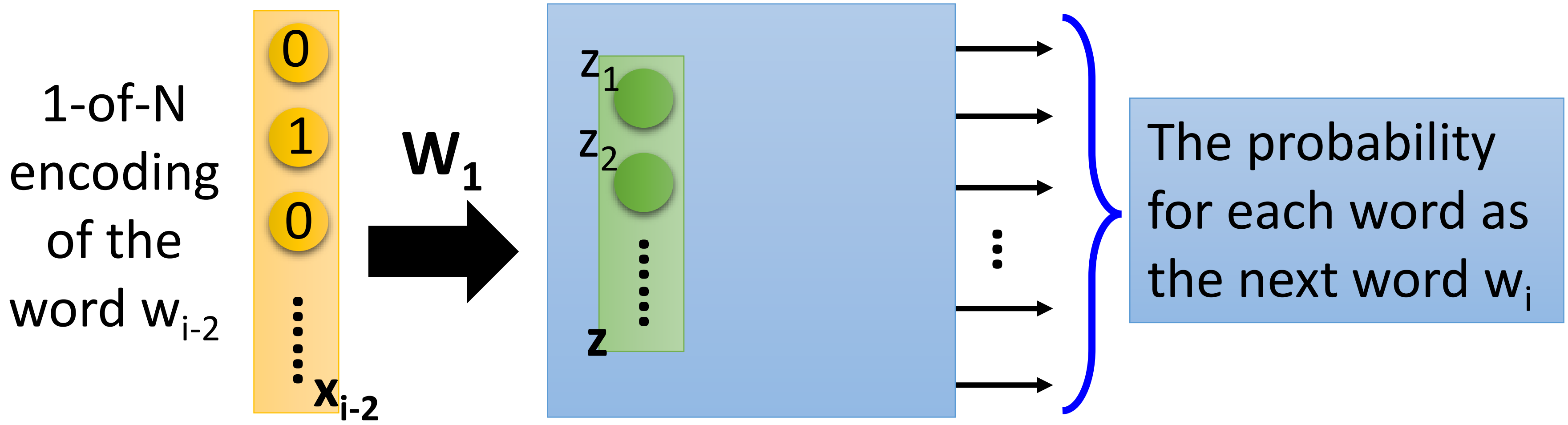
1-of-N encoding of the word w_{i-1}

The probability for each word as the next word w_i

The weights with the same color should be the same.

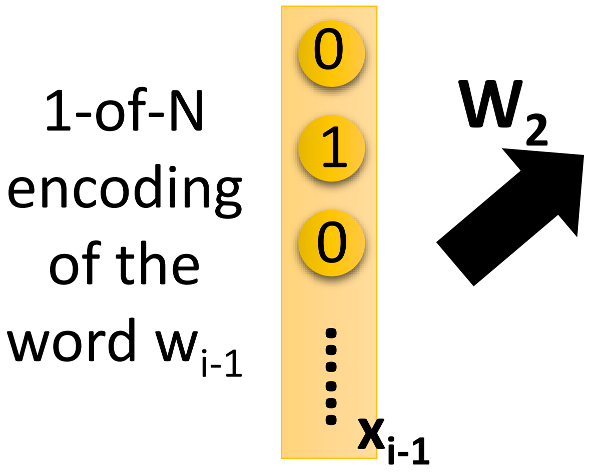
Or, one word would have two word vectors.

Word Vector – Sharing Parameters



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

The probability for each word as the next word w_i



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

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

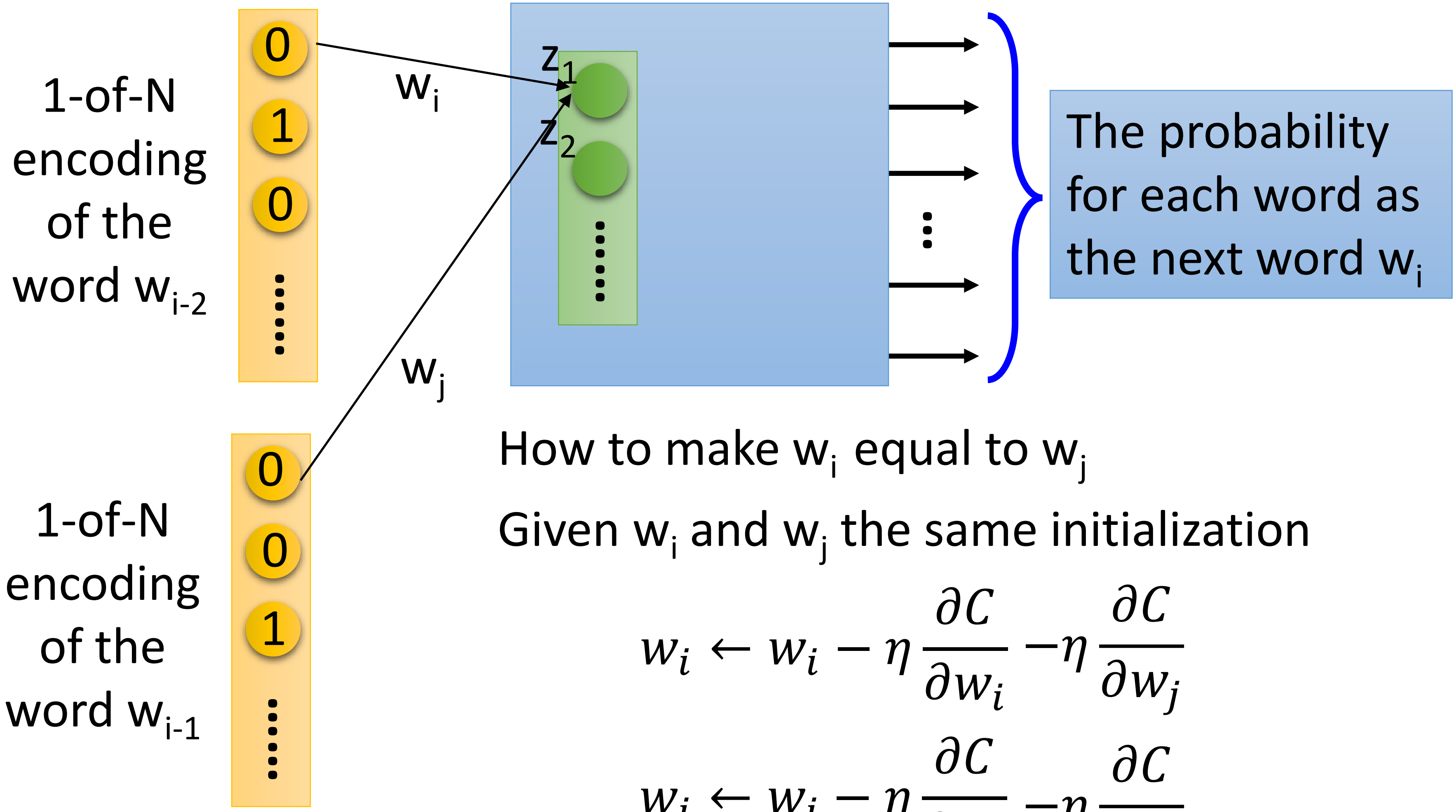
The length of $\mathbf{z}$ is |Z|.

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

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

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

Word Vector – Sharing Parameters



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

The probability for each word as the next word w_i

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

How to make w_i equal to w_j

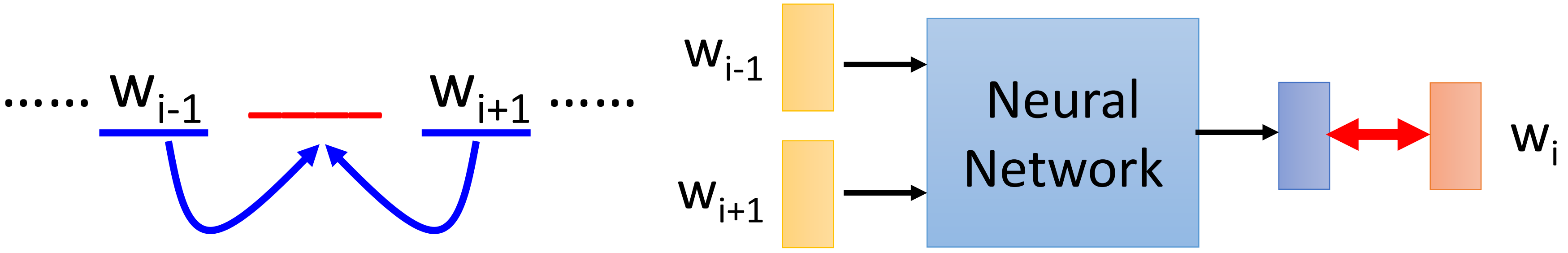
Given w_i and w_j the same initialization

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

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

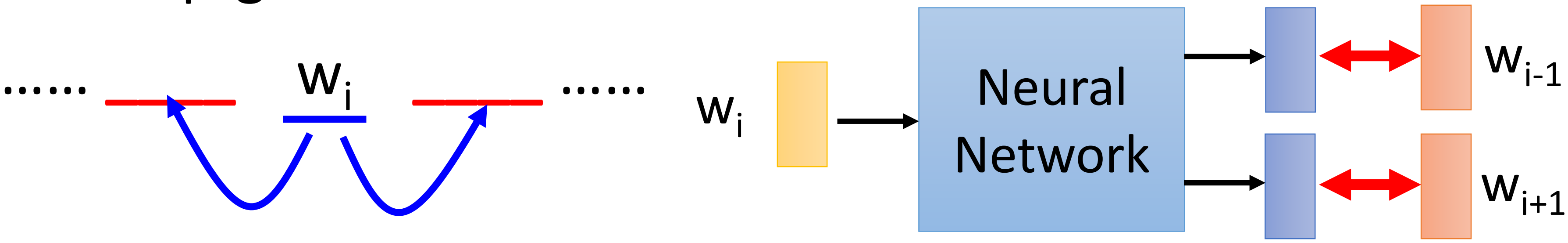
Word Vector
– Various Architectures

- Continuous bag of word (CBOW) model



predicting the word given its context

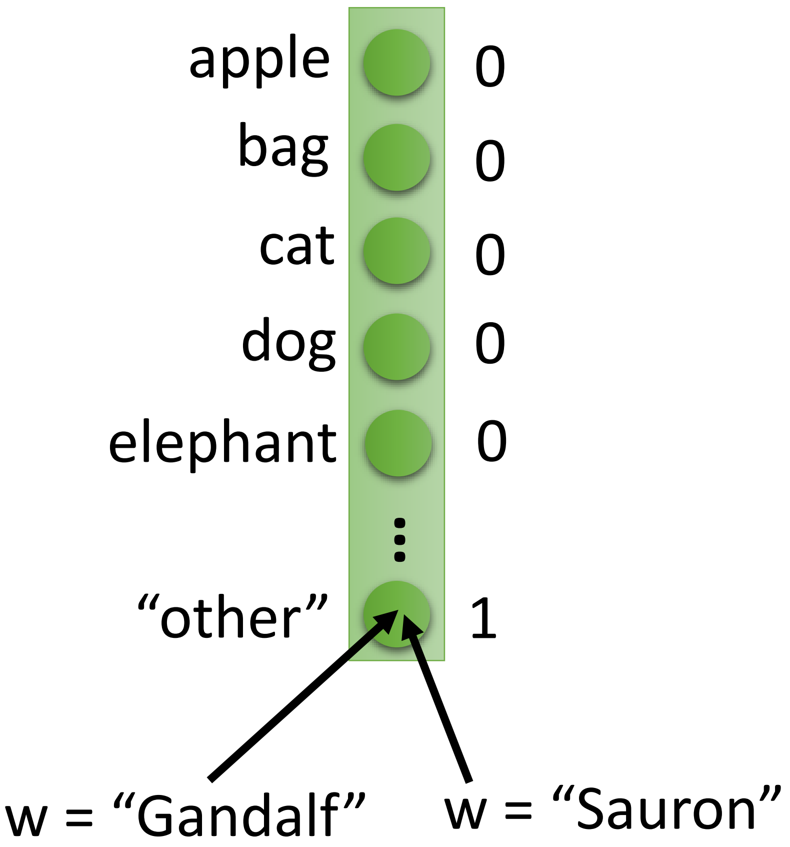
- Skip-gram



predicting the context given a word

Beyond 1-of-N encoding

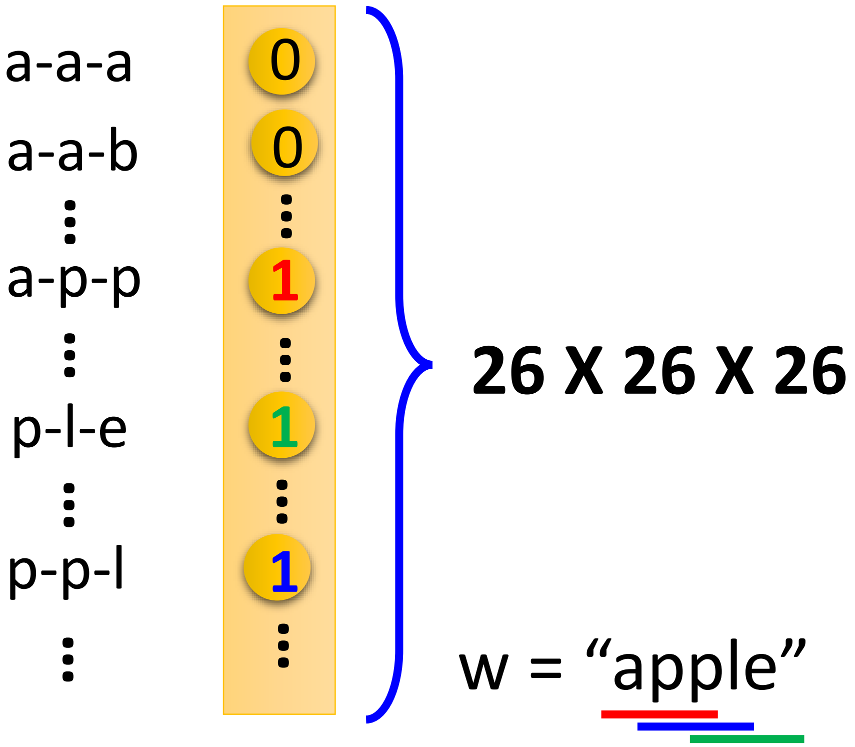
Dimension for "Other"



apple 0
bag 0
cat 0
dog 0
elephant 0
⋮
"other" 1

w = "Gandalf"
w = "Sauron"

Word hashing

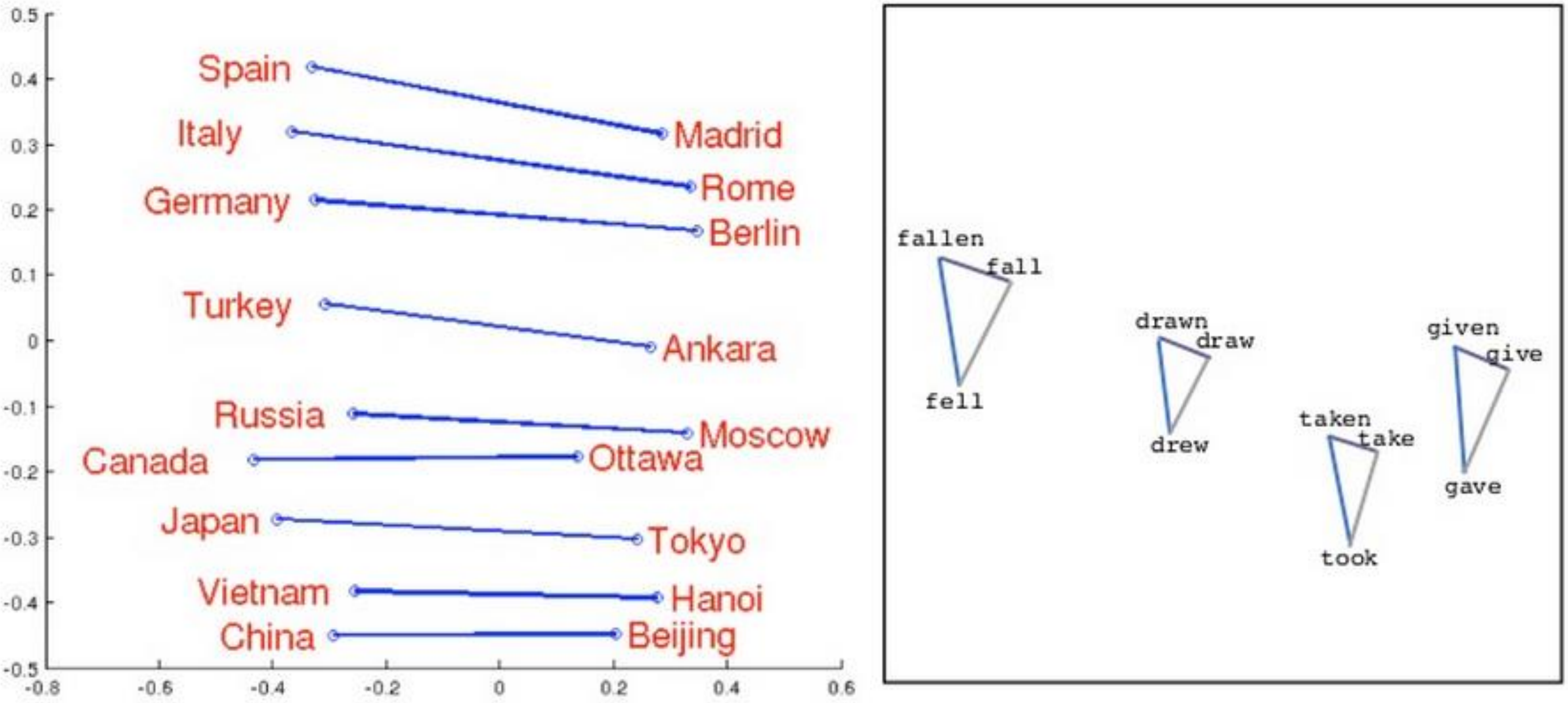


a-a-a 0
a-a-b 0
⋮
a-p-p 1
⋮
p-l-e 1
⋮
p-p-l 1
⋮

26 X 26 X 26

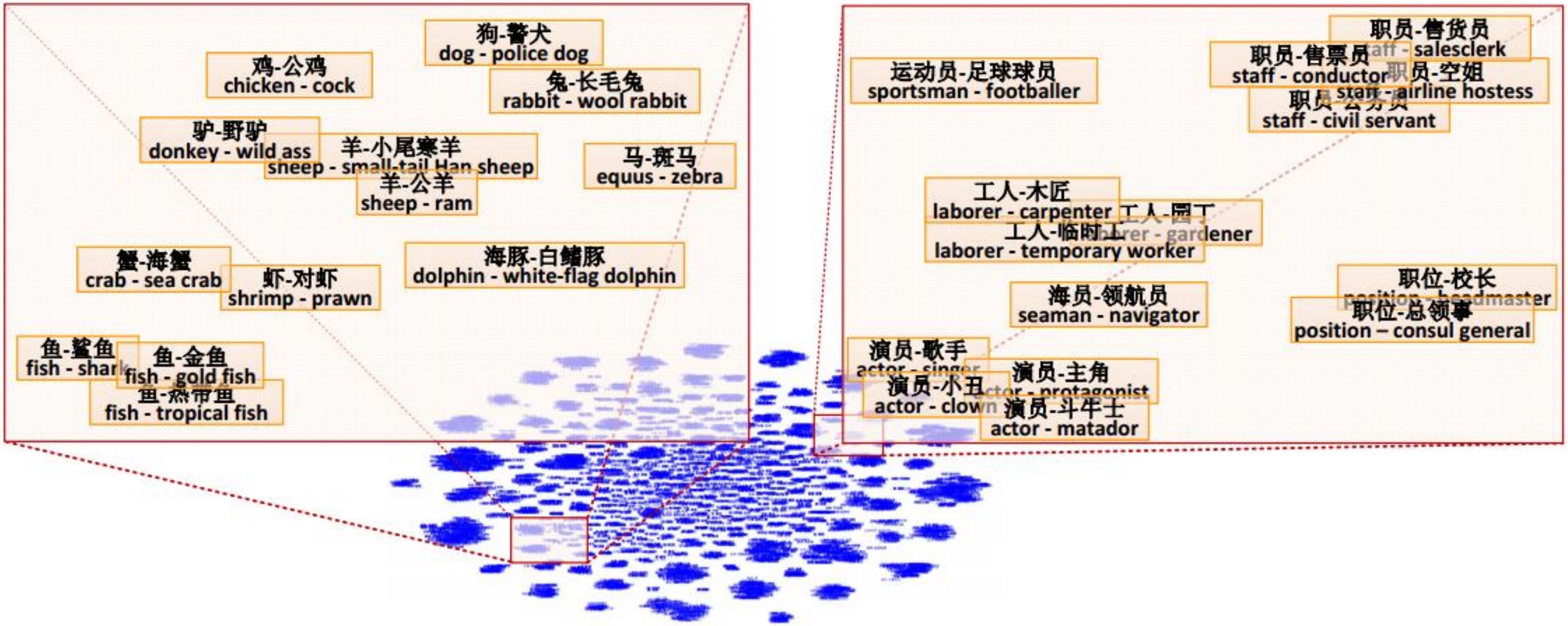
w = "apple"

Word Vector



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

Word Vector



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 Vector

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

- Characteristics

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

- Solving analogies

Rome : Italy = Berlin : ?

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

Find the word w with the closest V(w)

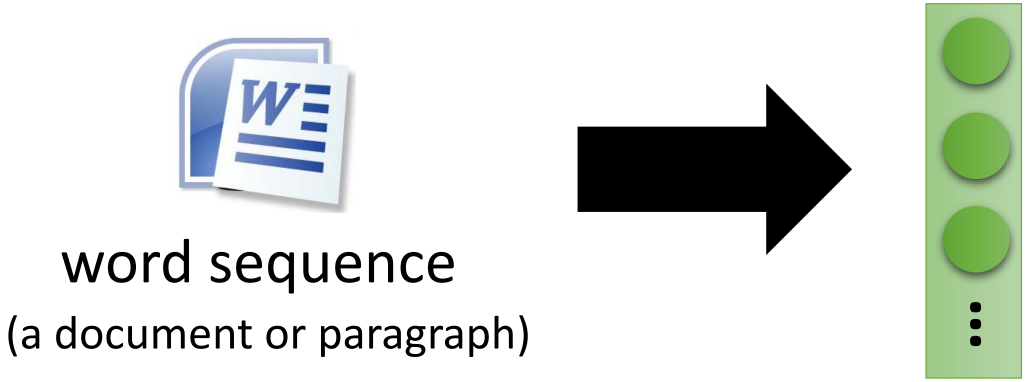
Demo

- 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)

Meaning of Word Sequence

Meaning of Word Sequence

- 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)

Outline

Deep Structured Semantic Model (DSSM)

- Application: Information Retrieval (IR)

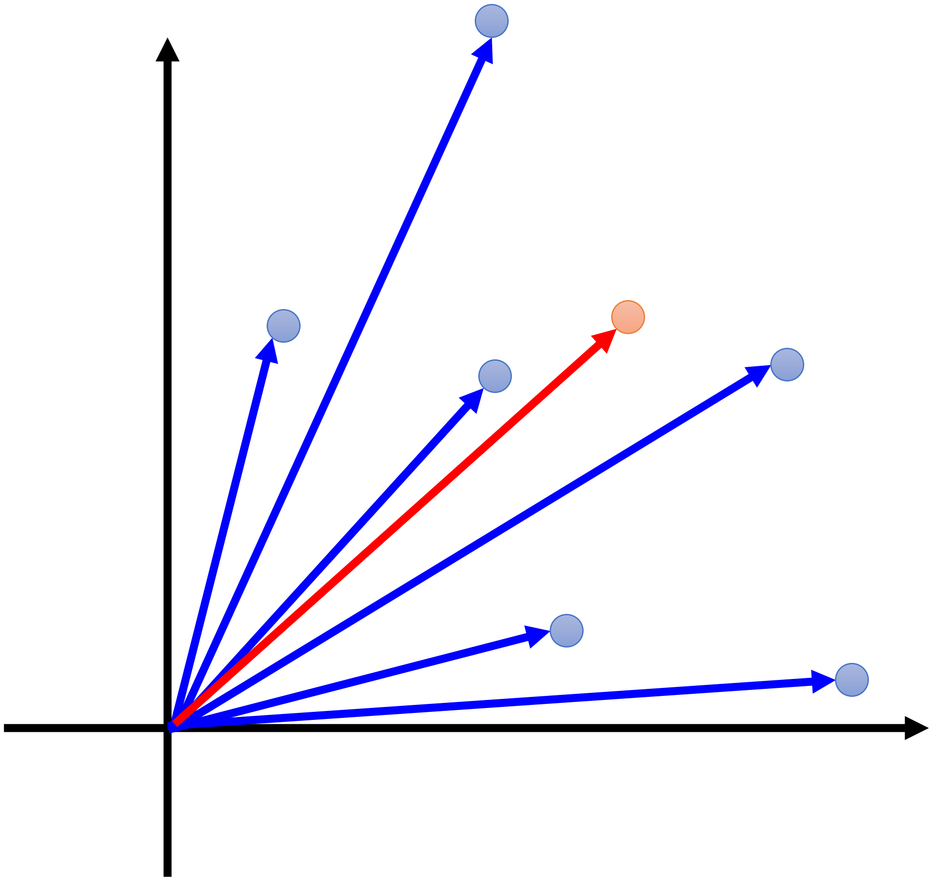
Recursive Neural Network

- Application: Sentiment Analysis, Sentence Relatedness

Unsupervised

- Paragraph Vector
- Sequence-to-sequence auto-encoder

Information Retrieval (IR)



Vector Space Model

The documents are vectors in the space.

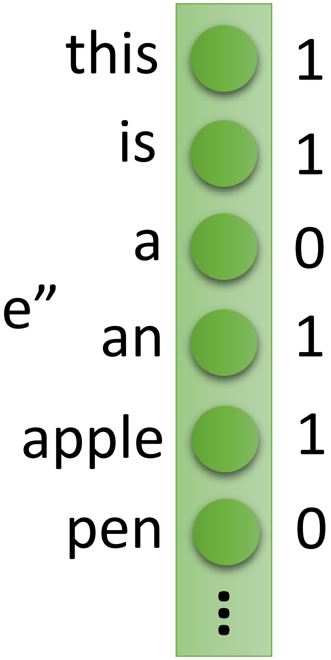
The query is also a vector.

How to use a vector to represent word sequences

Information Retrieval (IR)

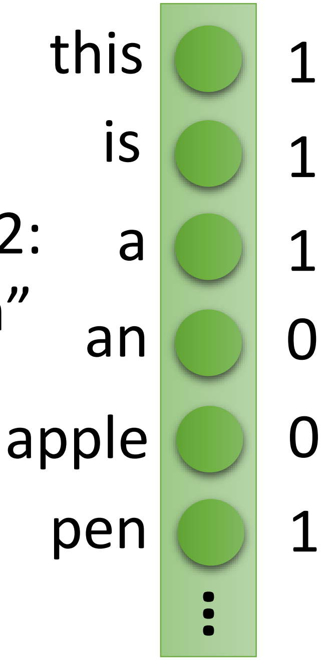
Bag-of-word

word string s1:
“This is an apple”



this	1
is	1
a	0
an	1
apple	1
pen	0
⋮	

word string s2:
“This is a pen”

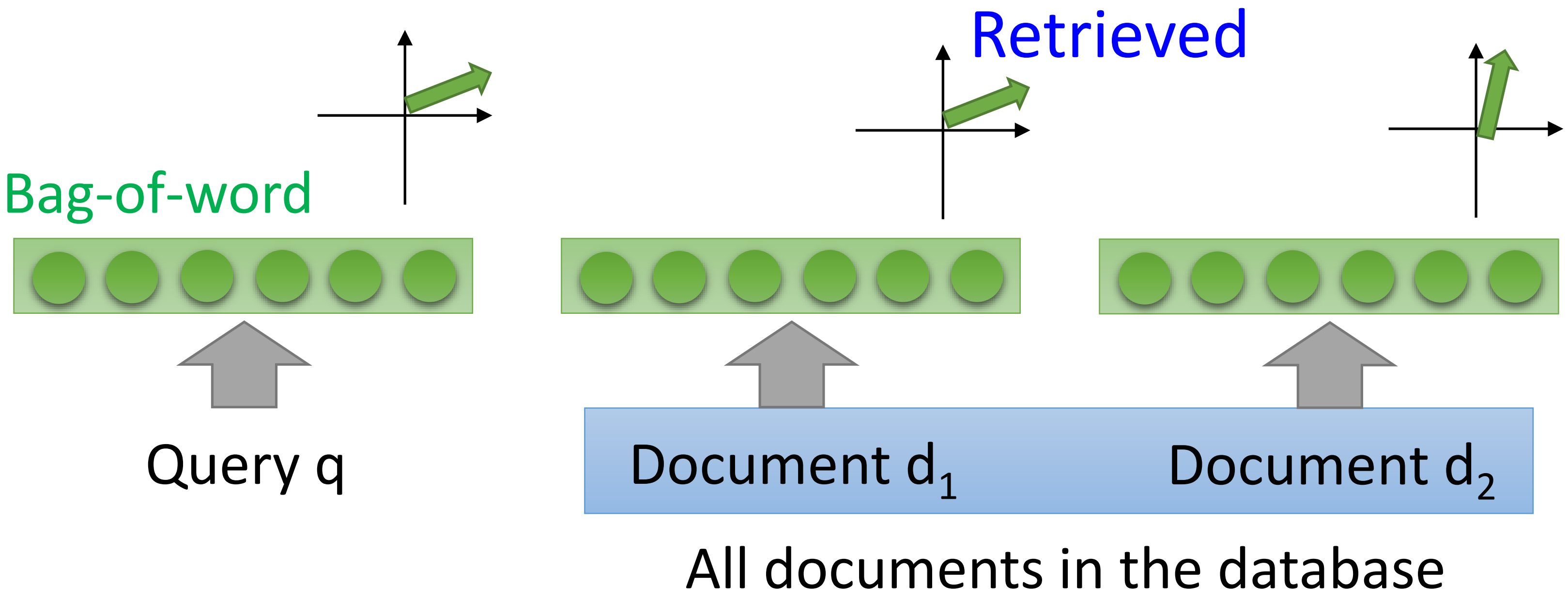


this	1
is	1
a	1
an	0
apple	0
pen	1
⋮	

Weighted by IDF

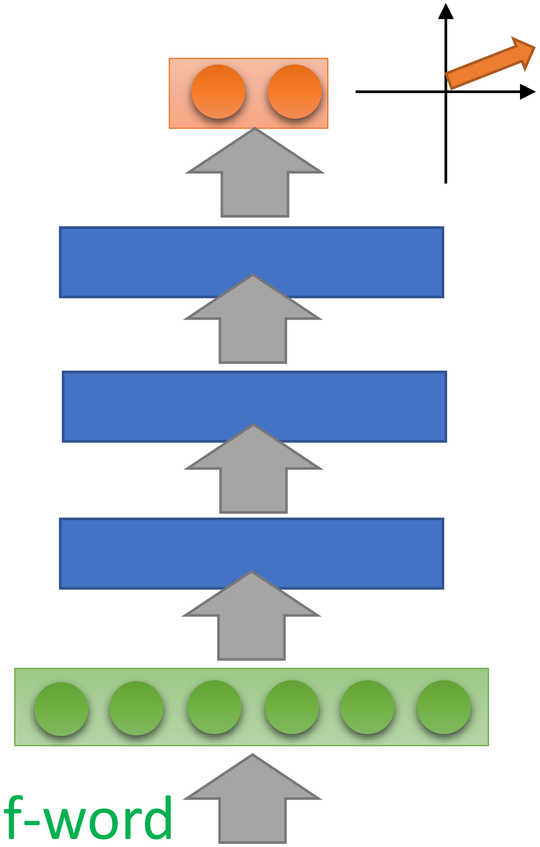
Information Retrieval (IR)

Vector Space Model + Bag-of-word



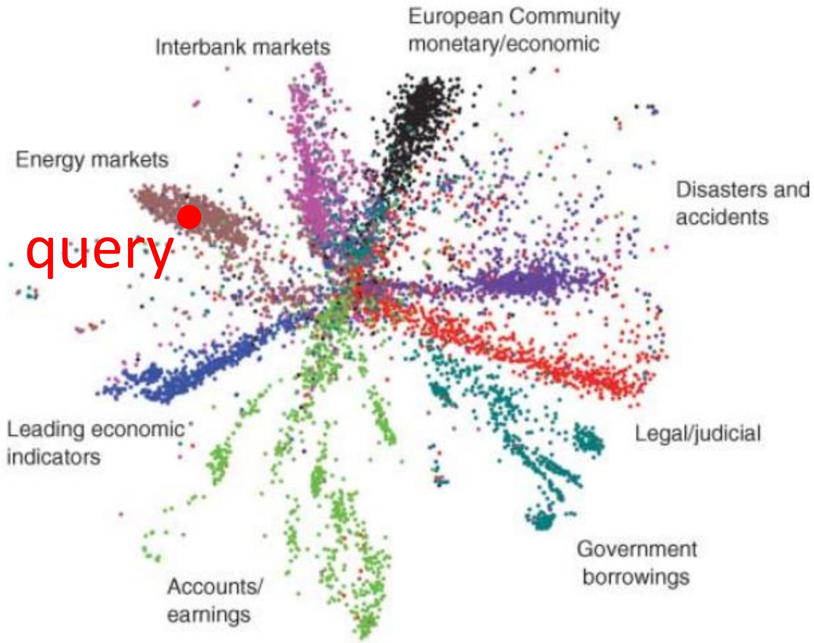
- ➢ All the words are treated as discrete tokens.
- ➢ Never considered: Different words can have the same meaning, and the same word can have different meanings.

IR - Semantic Embedding



Bag-of-word

word string
(document or query)



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

How to achieve that? (No target ……)

DSSM

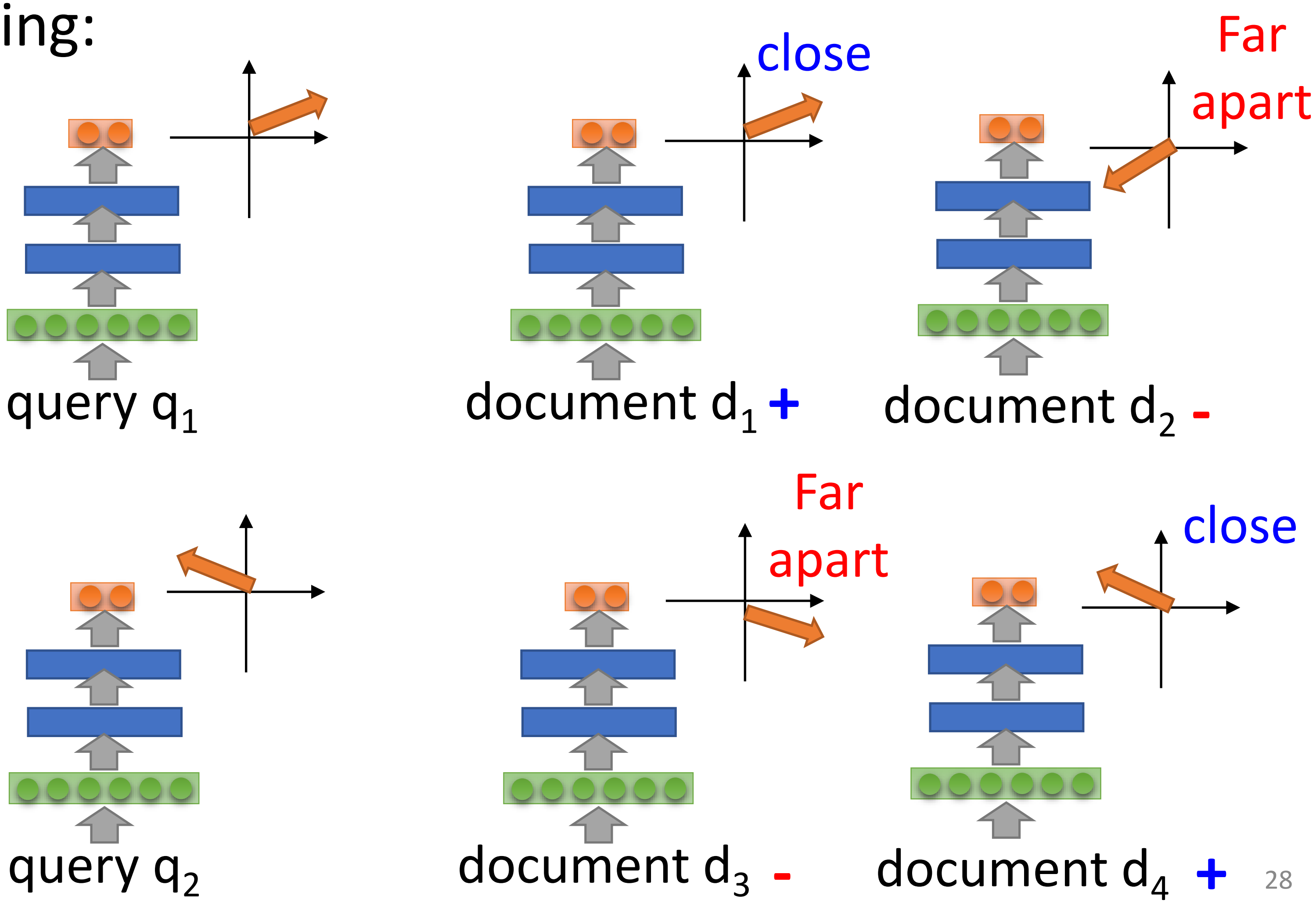
Click-through data:

q_1 → d_1 : + d_2 : -

q_2 → d_3 : - d_4 : +

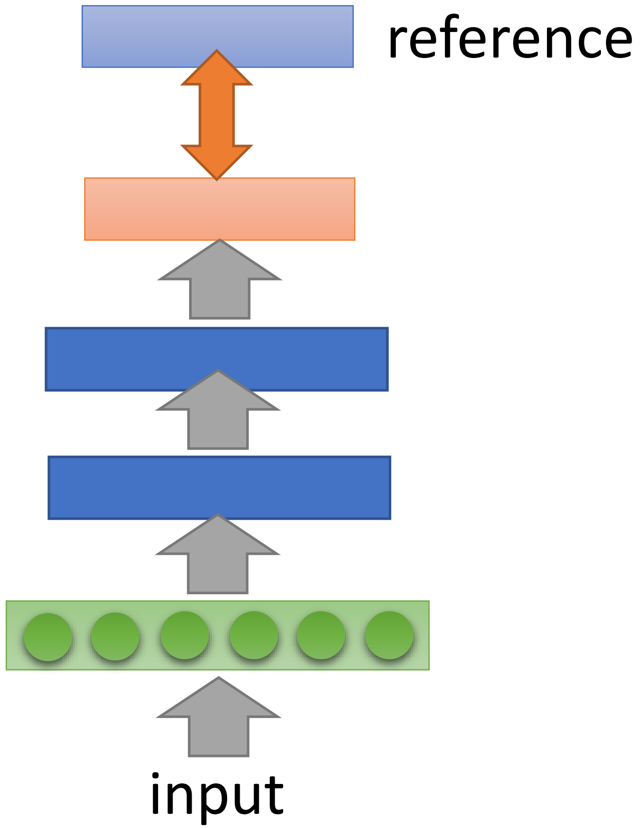
……

Training:

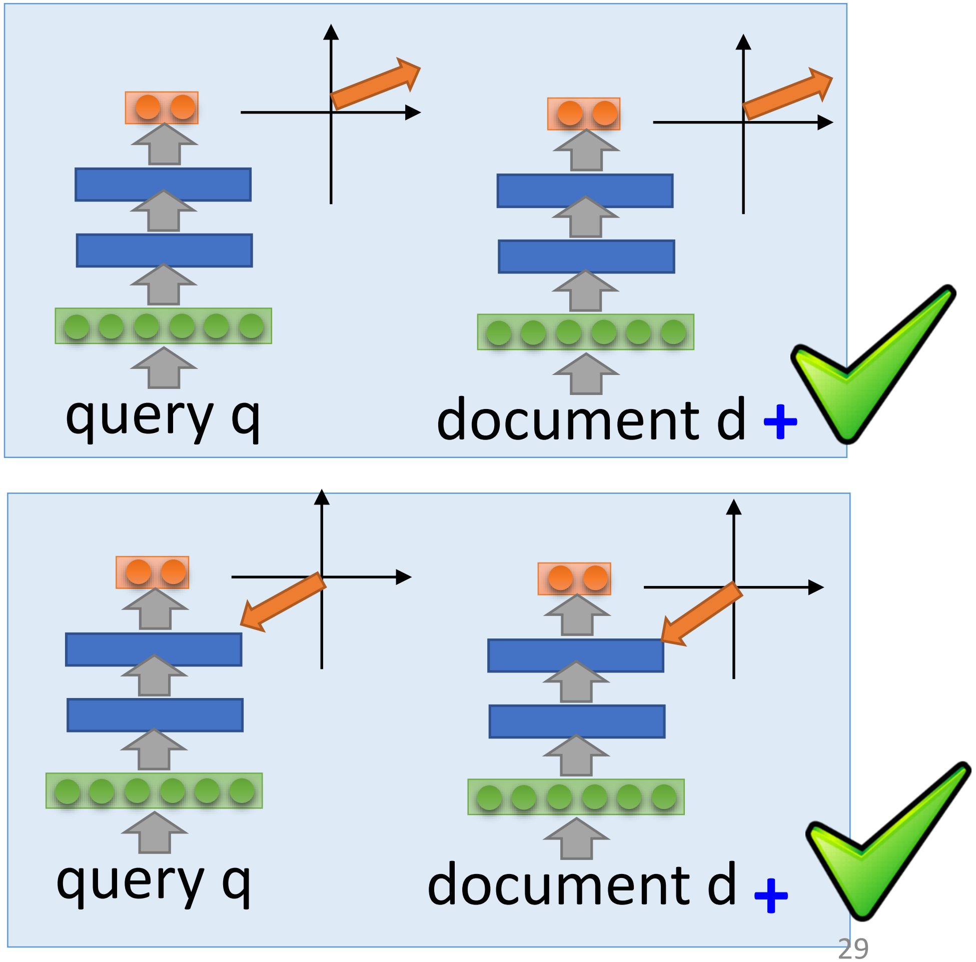


DSSM v.s. Typical DNN

Typical DNN



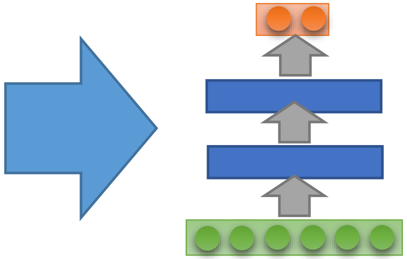
DSSM



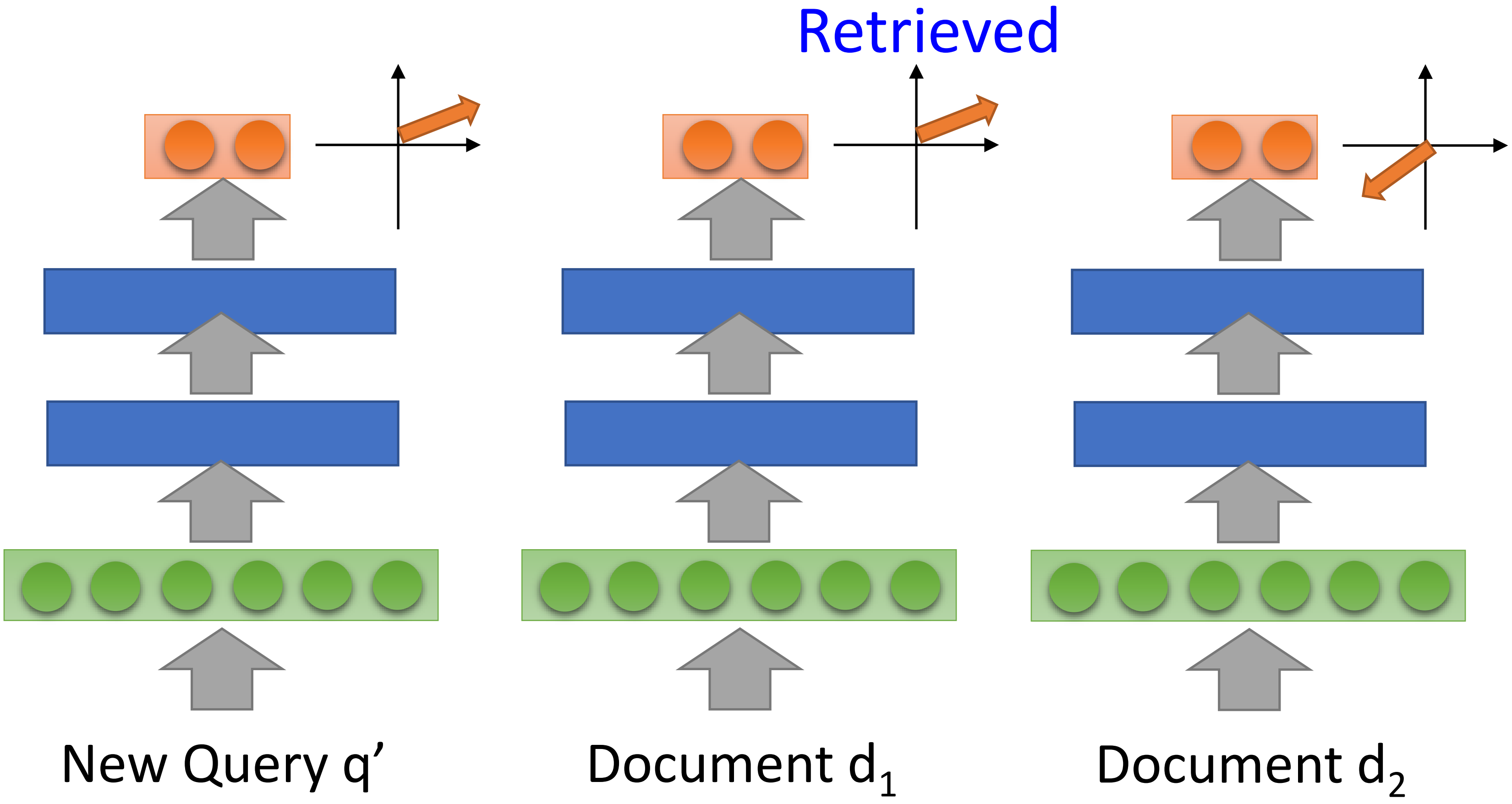
Click-through data: $q_1 \longrightarrow d_1$: + d_2 : -

$q_2 \longrightarrow d_3$: - d_4 : +

……

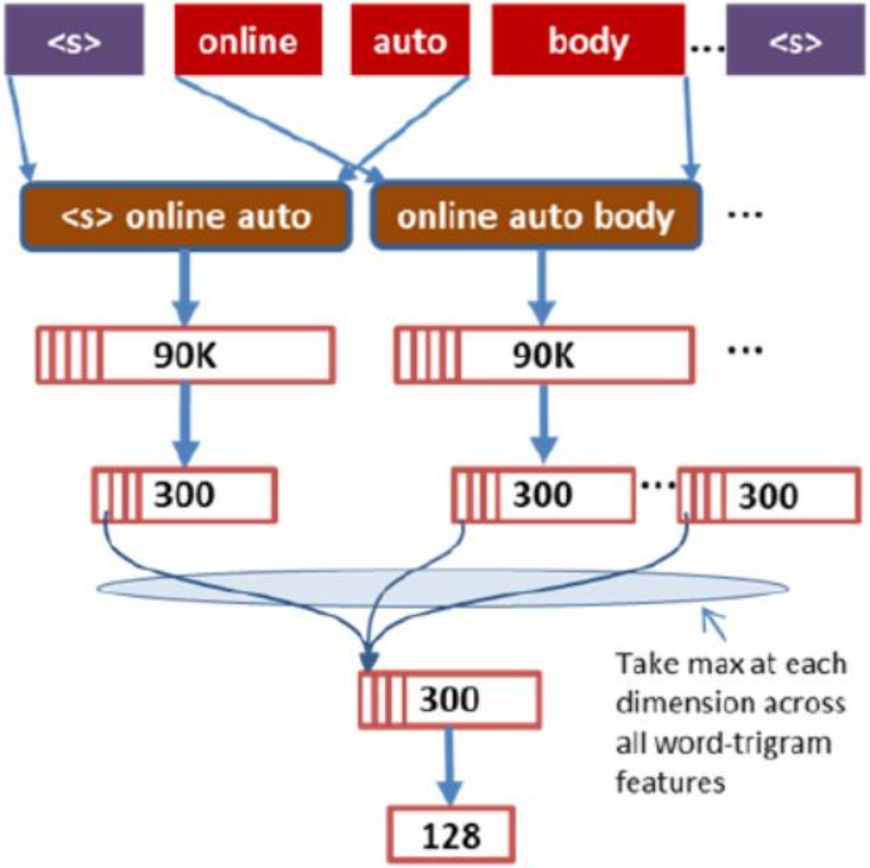


- How to do retrieval?



Reference

- 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.



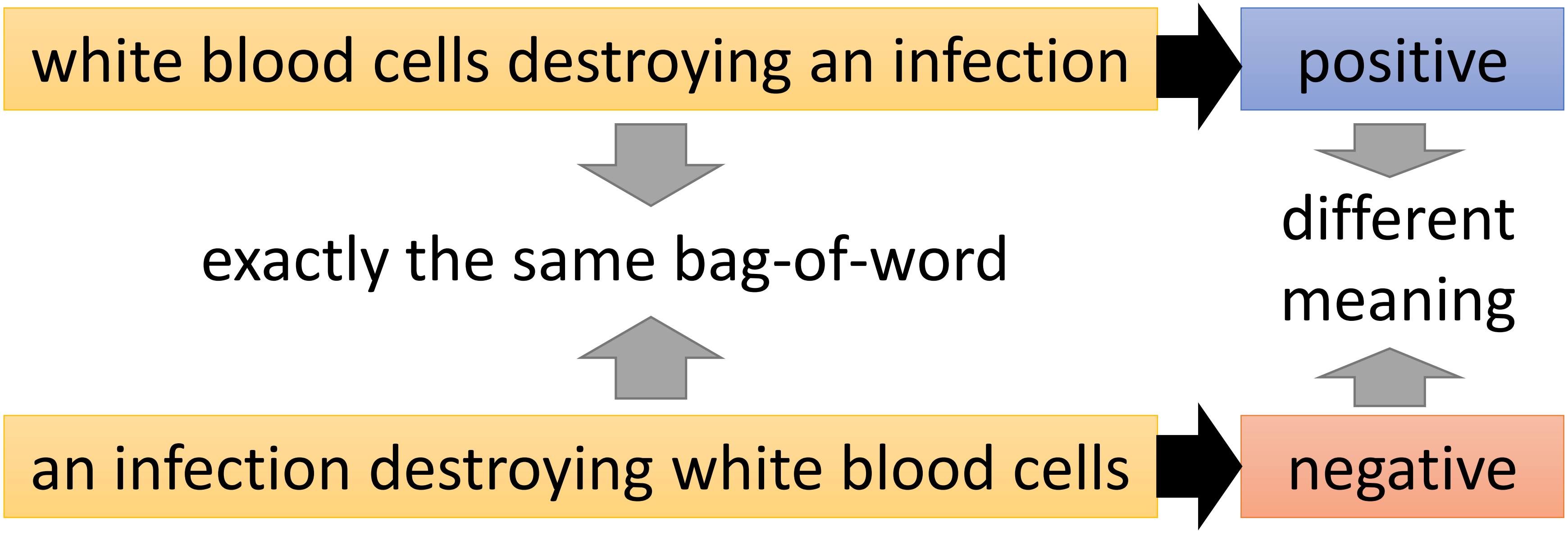
Take max at each dimension across all word-trigram features

Outline

Deep Structured Semantic Model (DSSM)	• Application: Information Retrieval (IR)
Recursive Neural Network	• Application: Sentiment Analysis, Sentence Relatedness
Unsupervised	• Paragraph Vector • Sequence-to-sequence auto-encoder

Recursive Deep Model

- 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

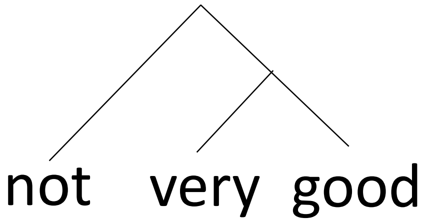
different meaning

an infection destroying white blood cells

negative

Recursive Deep Model

syntactic structure

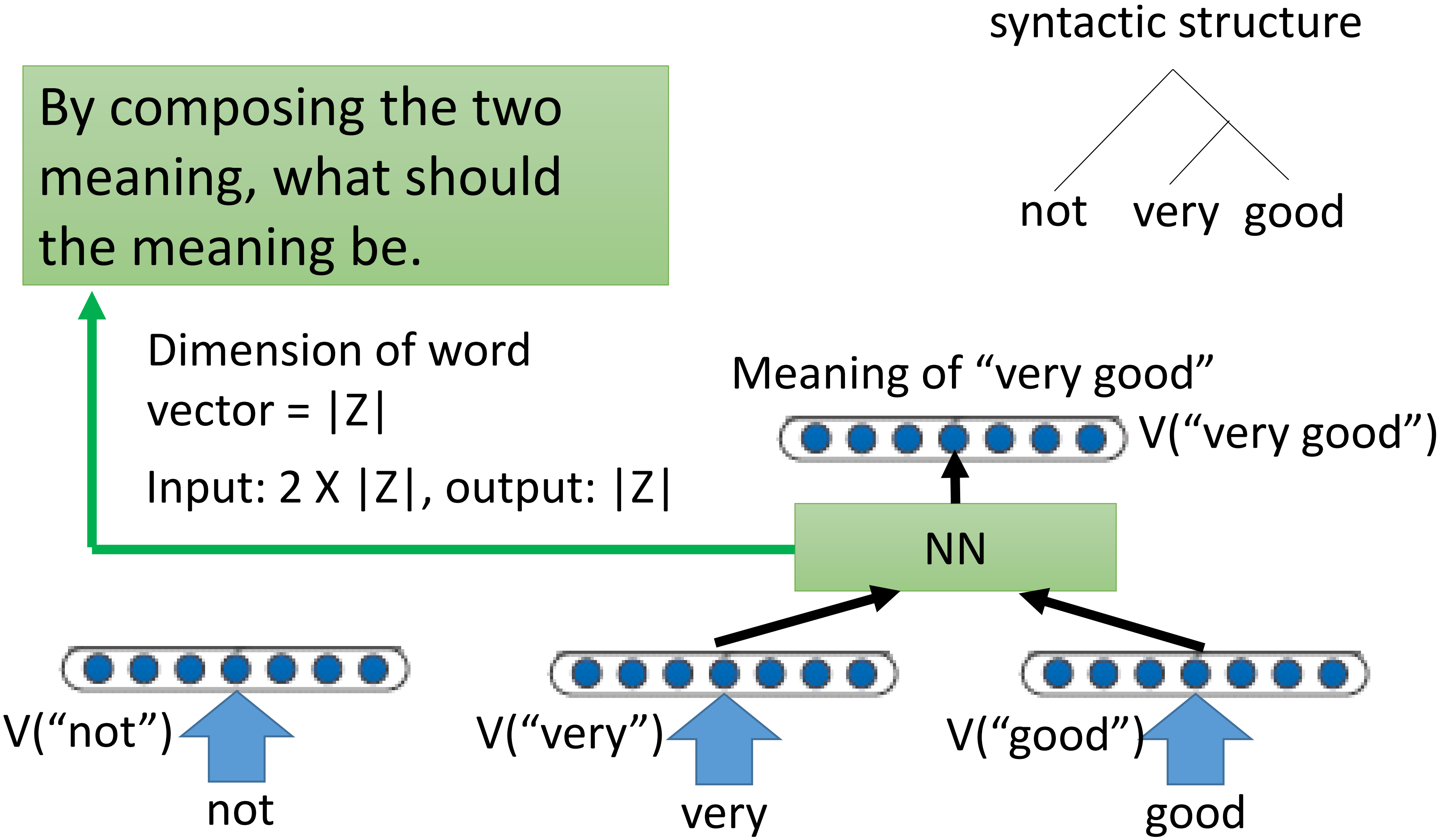


How to do it is out of the scope

word sequence:

not very good

Recursive Deep Model



syntactic structure

not very good

By composing the two meaning, what should the meaning be.

Dimension of word vector = |Z|

Input: 2 X |Z|, output: |Z|

Meaning of “very good”

V(“very good”)

NN

V(“not”)

not

V(“very”)

very

V(“good”)

good

Recursive Deep Model

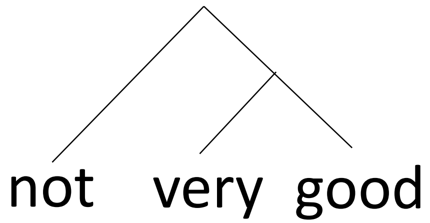
$V(w_A\, w_B) \neq V(w_A) + V(w_B)$

“not”: neutral

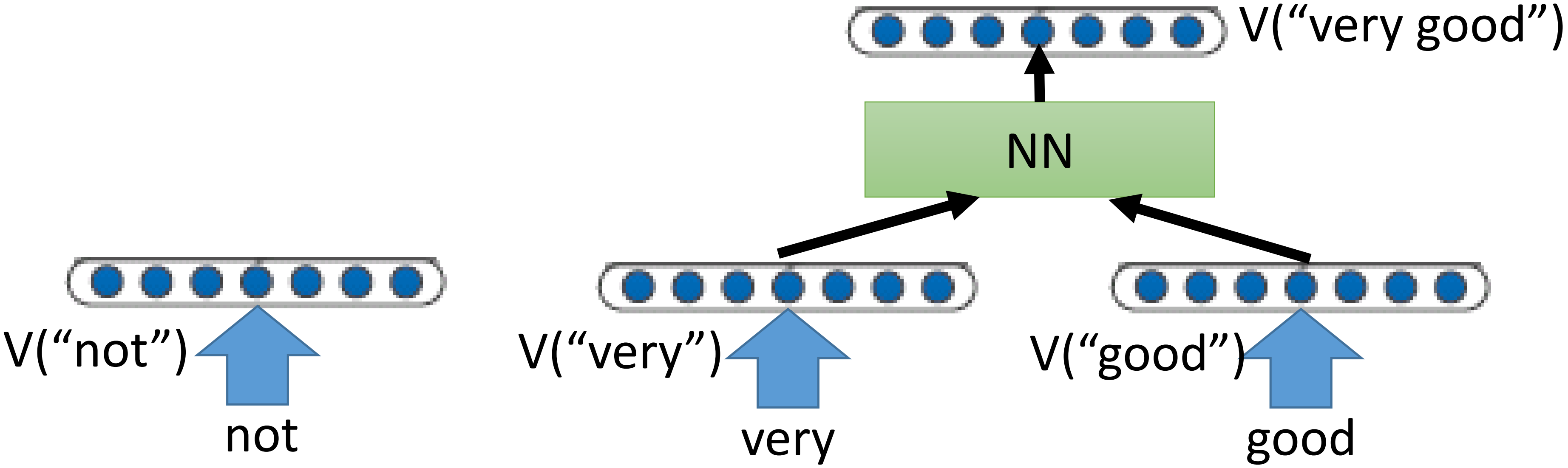
“good”: positive

“not good”: negative

syntactic structure



Meaning of “very good”



Recursive Deep Model

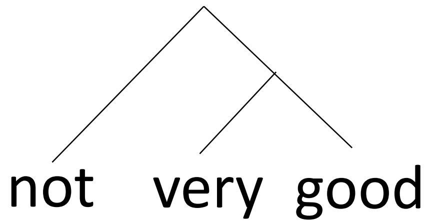
$V(w_A\, w_B) \neq V(w_A) + V(w_B)$

“棒”: positive

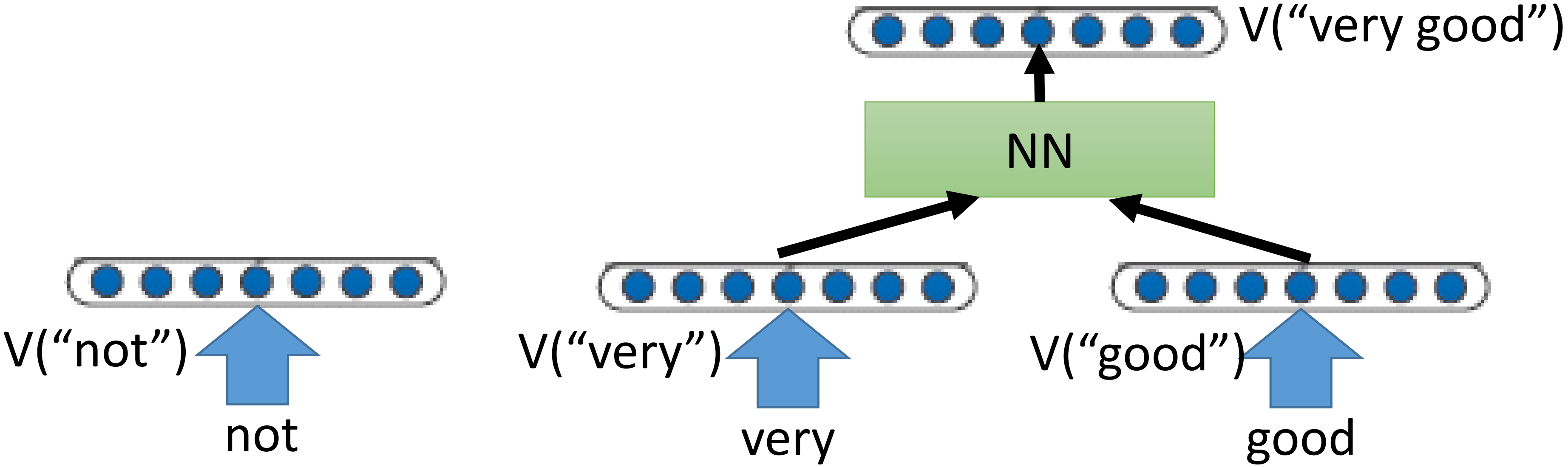
“好棒”: positive

“好棒棒”: negative

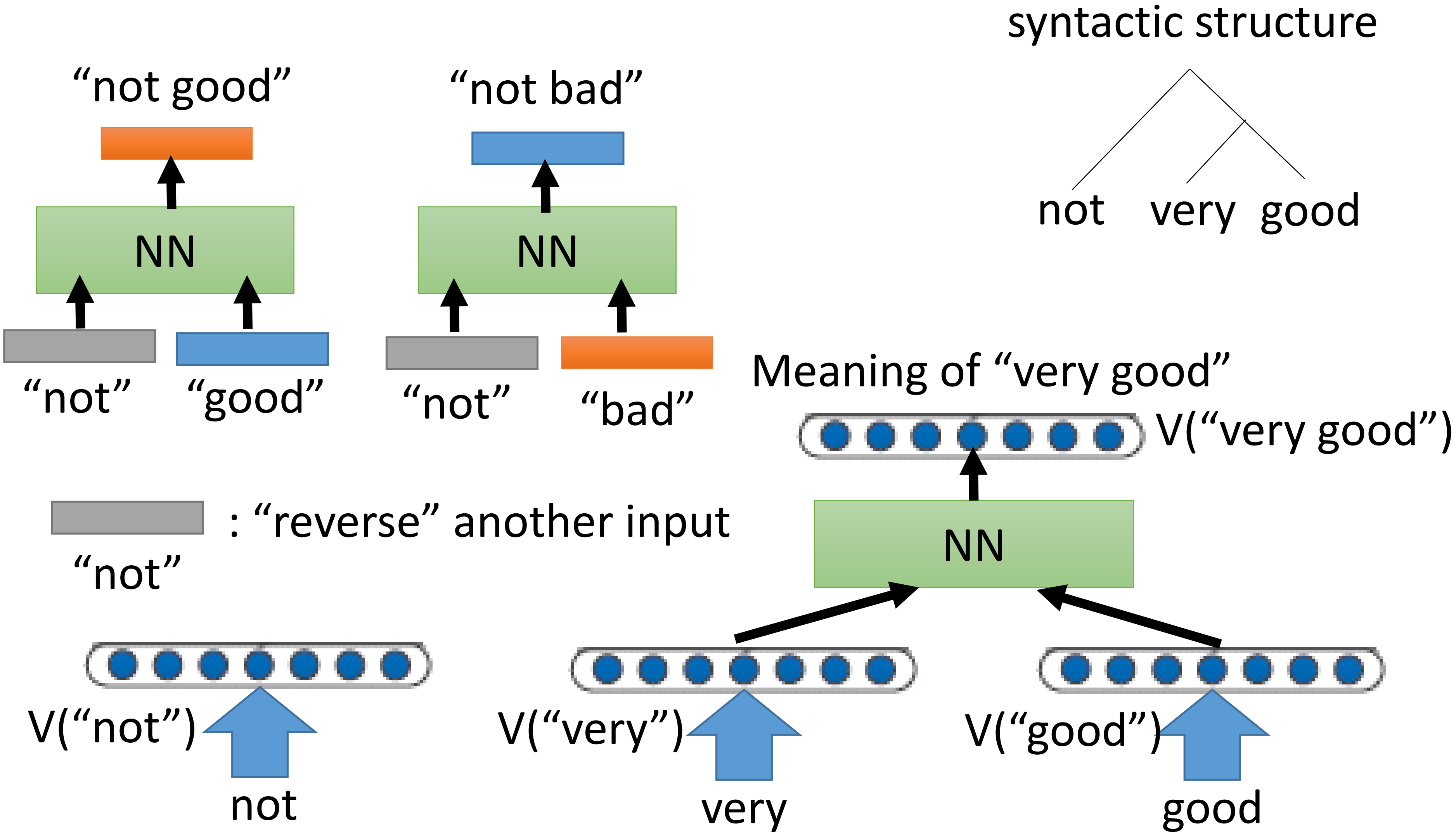
syntactic structure



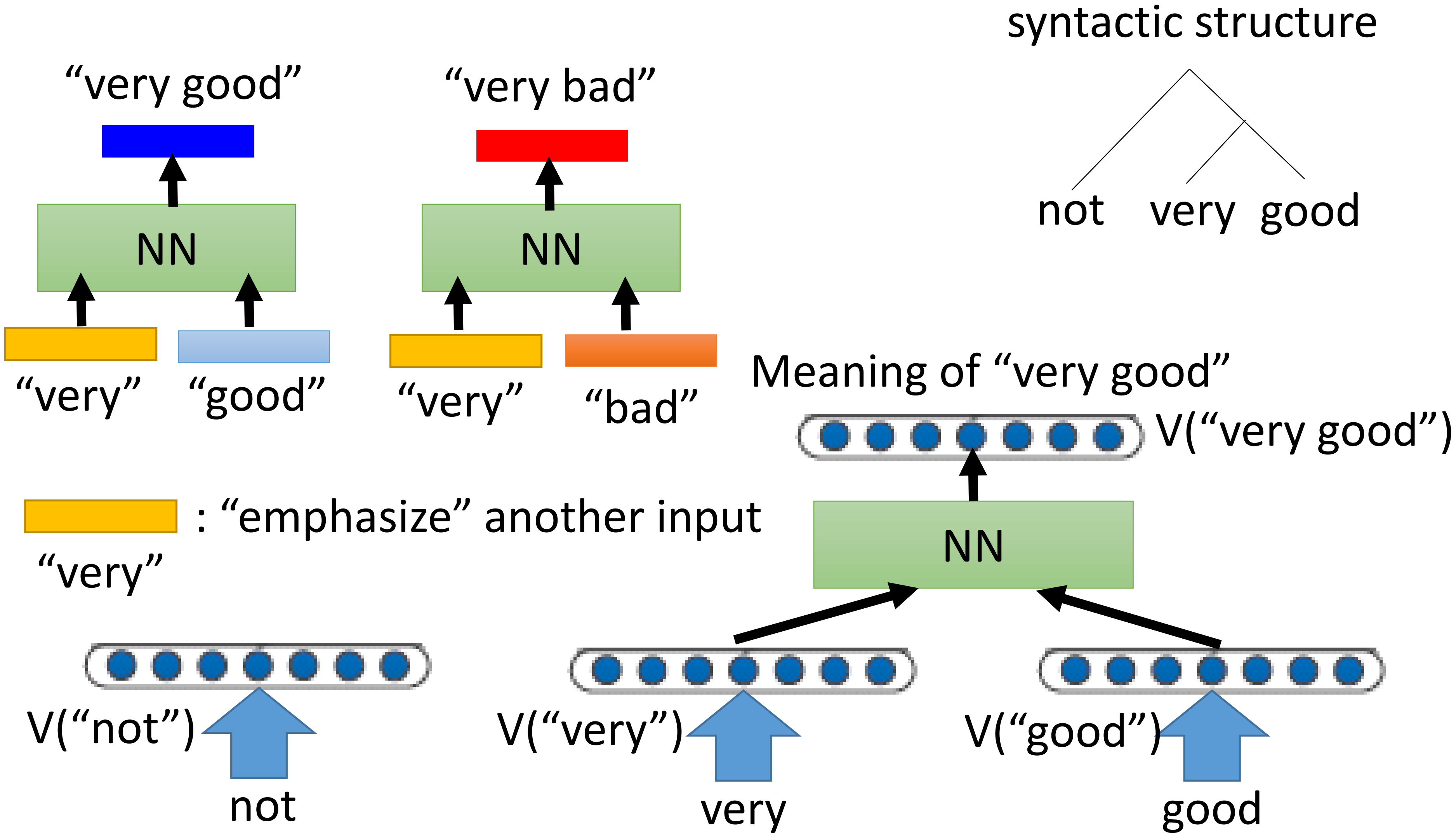
Meaning of “very good”



Recursive Deep Model



Recursive Deep Model



🟨 : “emphasize” another input

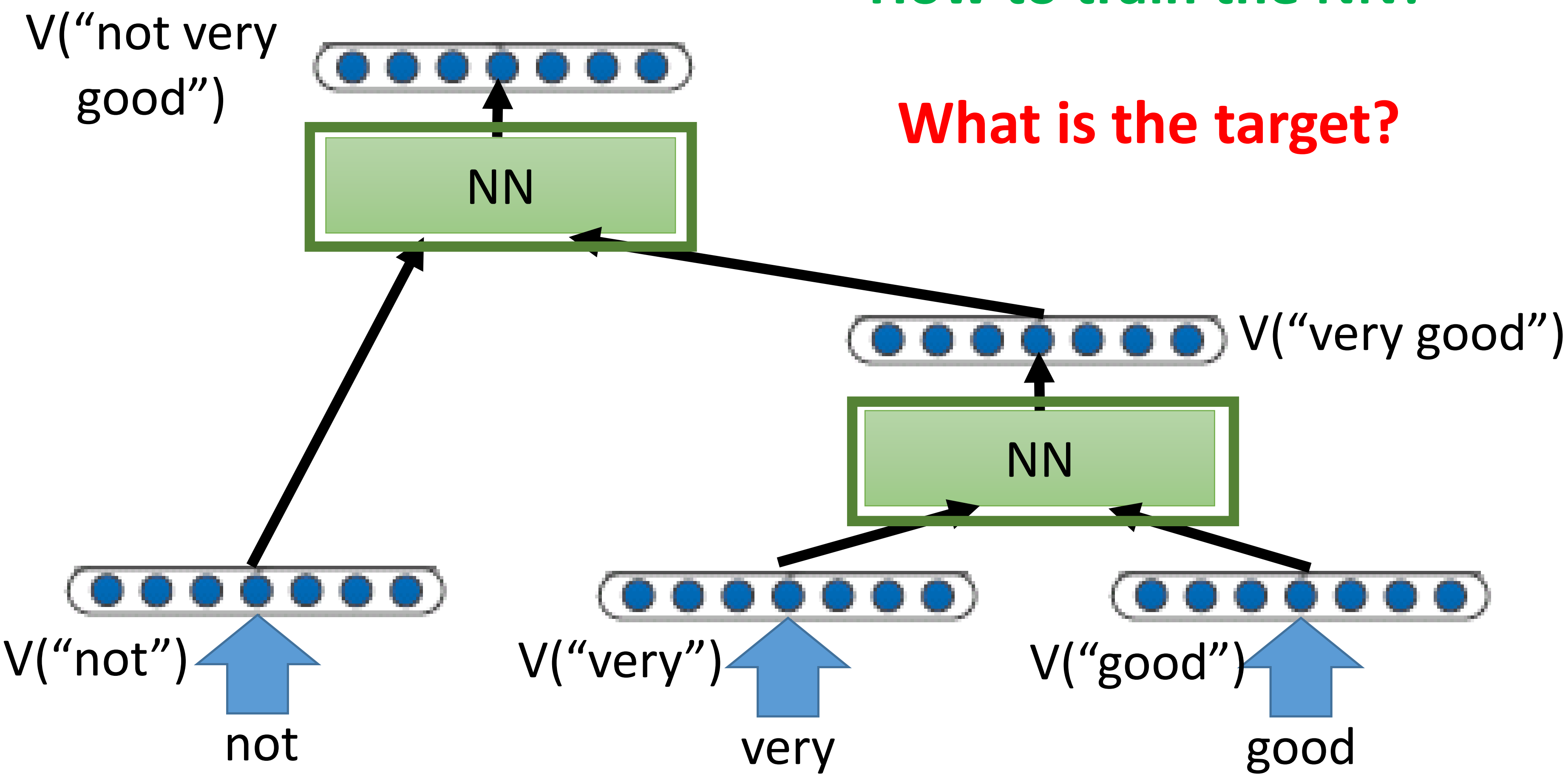
“very”

The word order is considered.

The representation of the sequence will change if the order of the words are changed

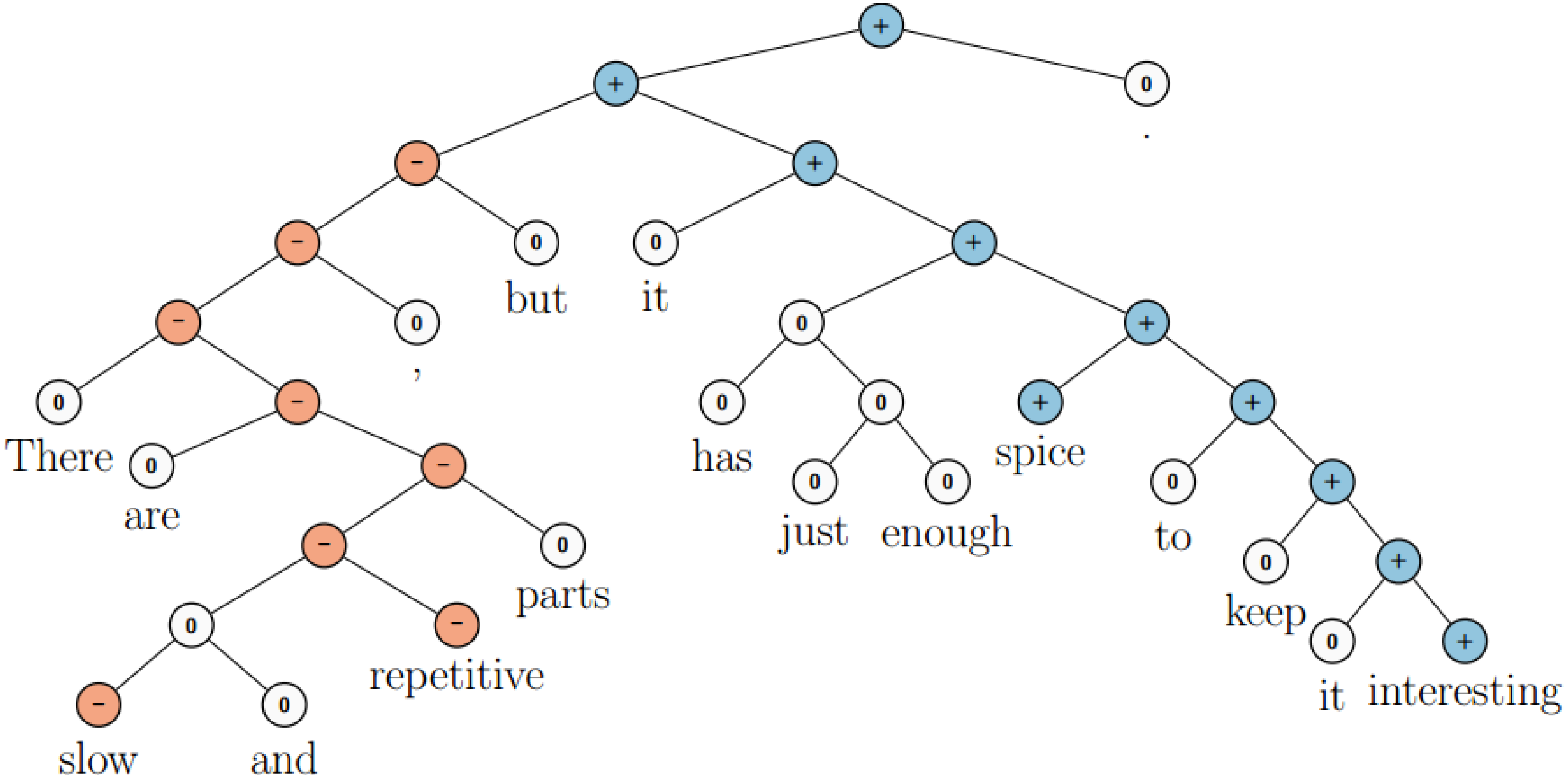
How to train the NN?

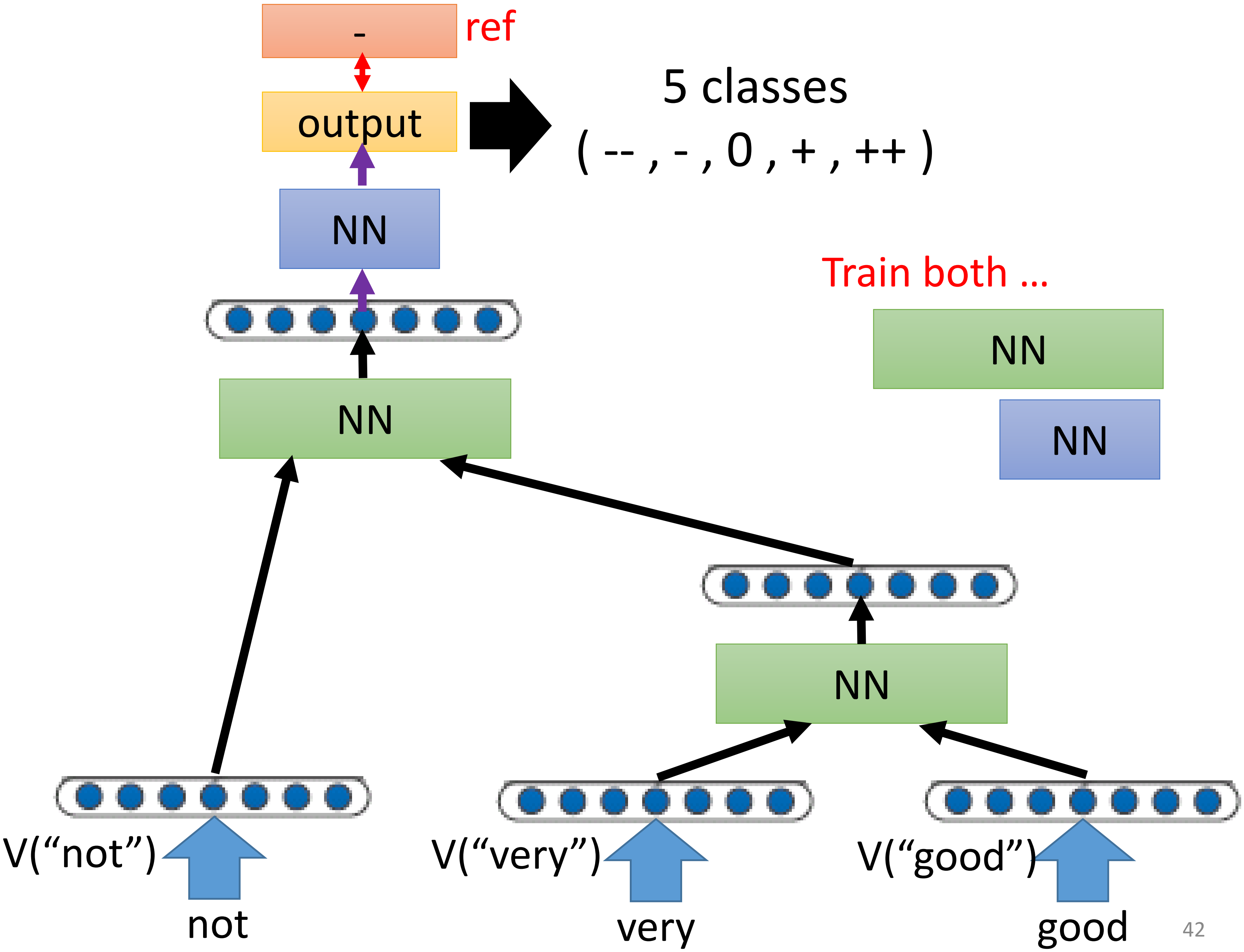
What is the target?



Need a Training Target ……

5-class sentiment classification (-- , - , 0 , + , ++)





ref

-

output

5 classes

(-- , - , 0 , + , ++)

NN

Train both …

NN

NN

NN

NN

V("not")

not

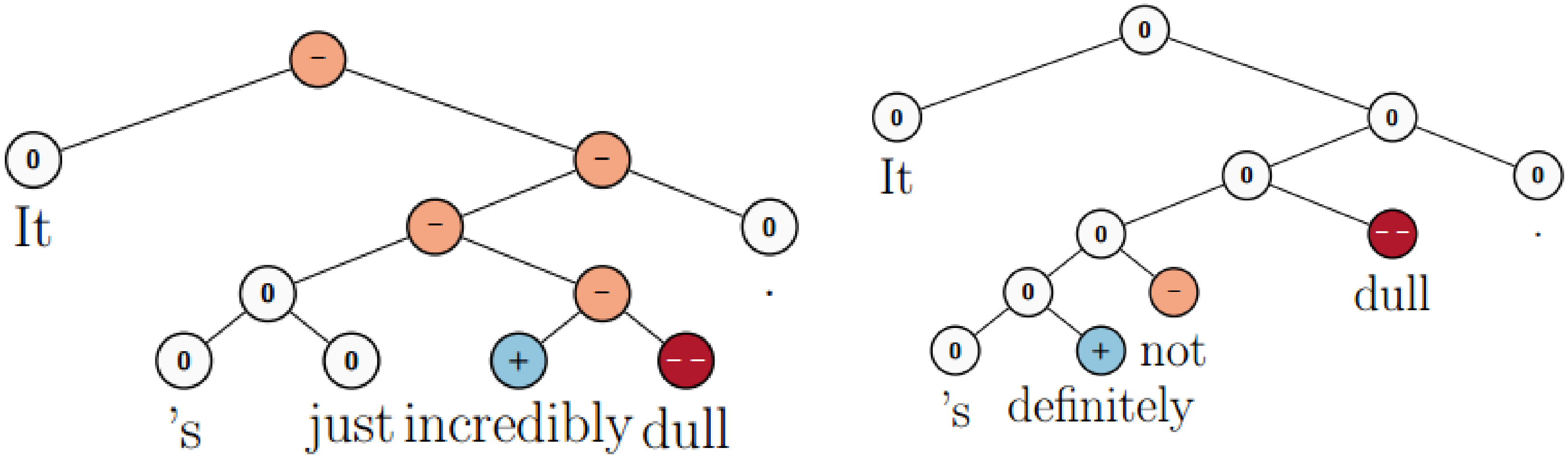
V("very")

very

V("good")

good

Sentiment Analysis

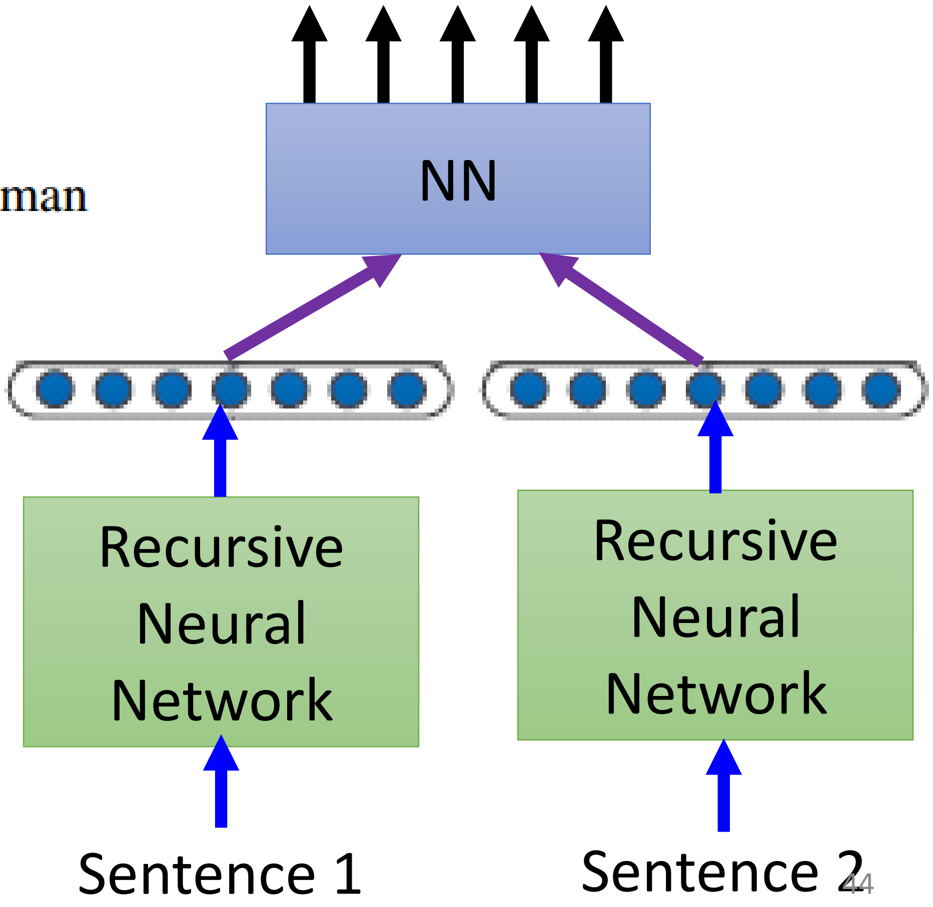


Socher, Richard, et al. "Recursive deep models for semantic compositionality over a sentiment treebank." *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*. Vol. 1631. 2013.

Need a Training Target ……

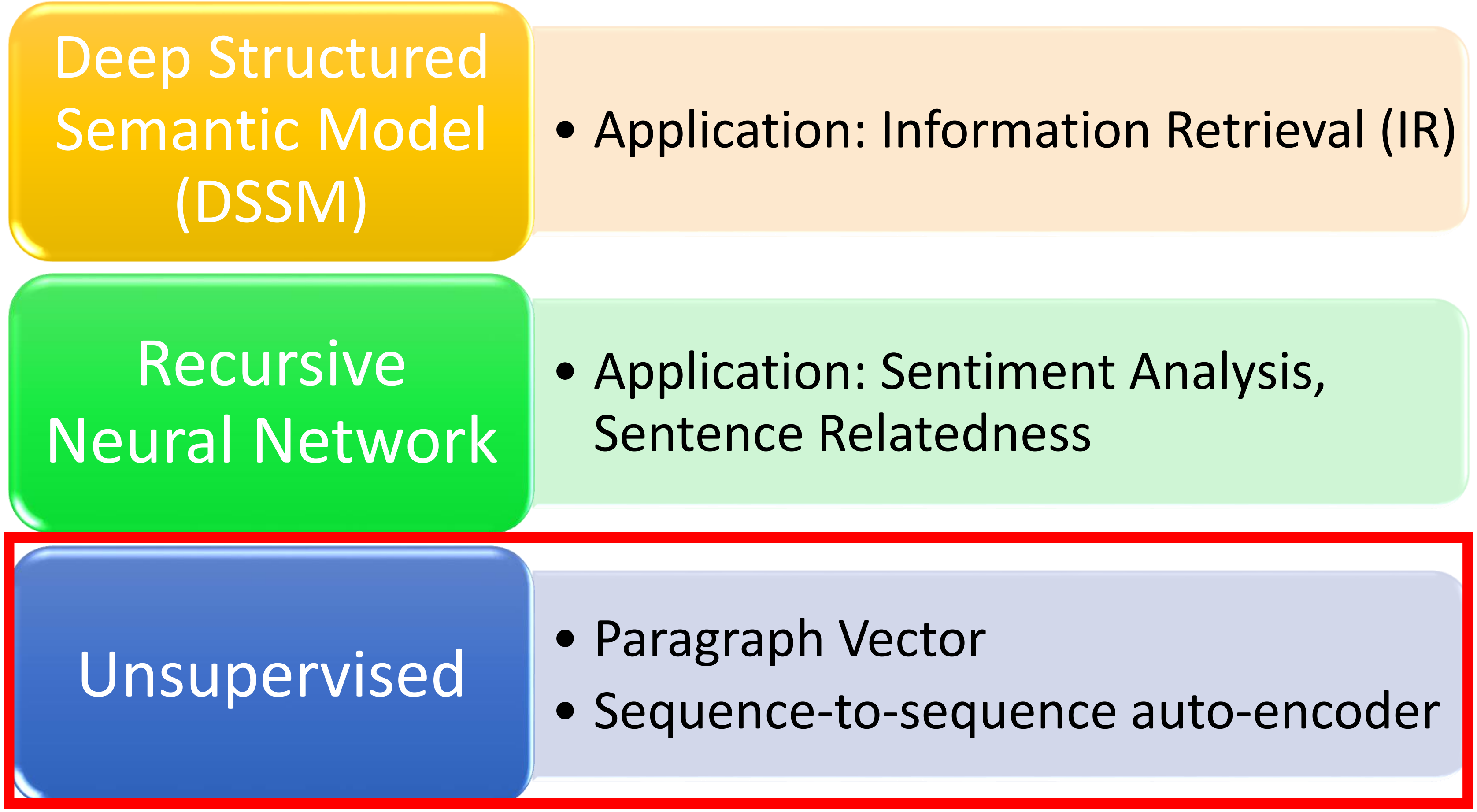
- Sentence relatedness

	a woman is slicing potatoes
4.82	a woman is cutting potatoes
4.70	potatoes are being sliced by a woman
4.39	tofu is being sliced by a woman



Tai, Kai Sheng, Richard Socher, and Christopher D. Manning. "Improved semantic representations from tree-structured long short-term memory networks." *arXiv preprint arXiv:1503.00075* (2015).

Outline



Deep Structured Semantic Model (DSSM)

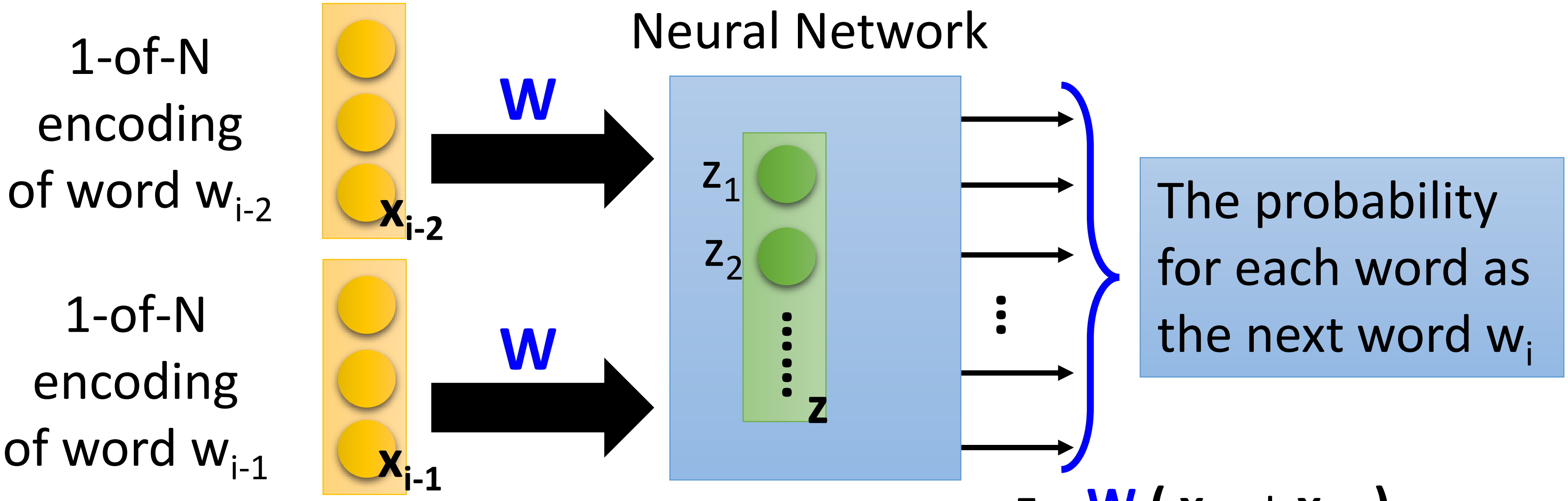
- Application: Information Retrieval (IR)

Recursive Neural Network

- Application: Sentiment Analysis, Sentence Relatedness

Unsupervised

- Paragraph Vector
- Sequence-to-sequence auto-encoder


1-of-N encoding of word w_{i-2}
x_{i-2}
1-of-N encoding of word w_{i-1}
x_{i-1}
W
W
Neural Network
z_1
z_2
z
The probability for each word as the next word w_i

$$z = W\,(\,x_{i-2} + x_{i-1}\,)$$

Paragraph d_1: (The paragraph is from "The lord of the ring")

…… 魔君 名叫 索倫 (Sauron) ……

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

$$z = W\,(\,x_{i-2} + x_{i-1}\,)$$

the same → Same output

Paragraph d_2: (The paragraph is from "仙五")

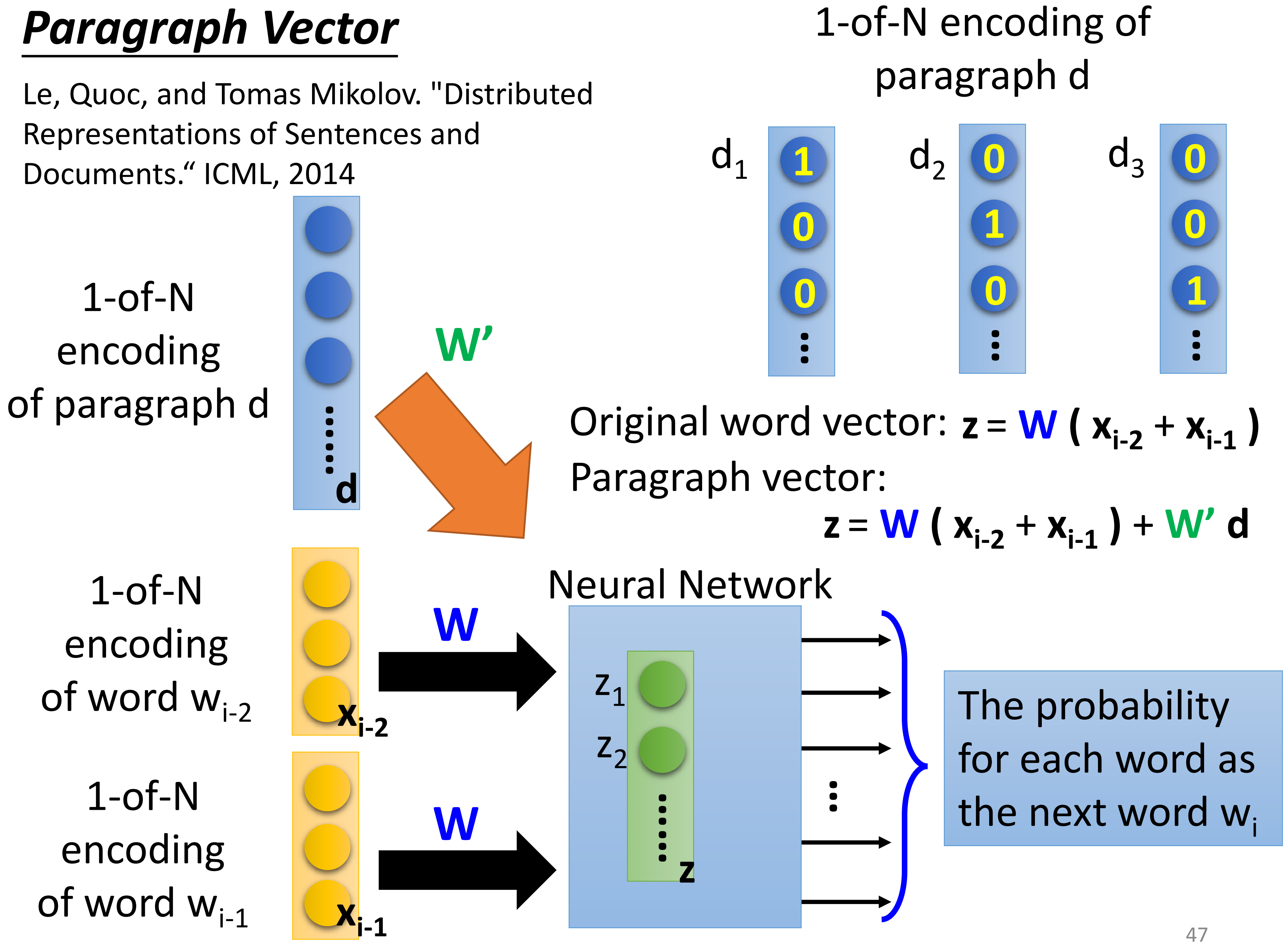
…… 魔君 名叫 姜世離 ……

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

$$z = W\,(\,x_{i-2} + x_{i-1}\,)$$

Paragraph Vector

Le, Quoc, and Tomas Mikolov. "Distributed Representations of Sentences and Documents.“ ICML, 2014


1-of-N encoding of paragraph d
d_1
1
0
0
d_2
0
1
0
d_3
0
0
1
1-of-N encoding of paragraph d
d
W’
1-of-N encoding of word w_{i-2}
x_{i-2}
1-of-N encoding of word w_{i-1}
x_{i-1}
W
W
Neural Network
z_1
z_2
z
The probability for each word as the next word w_i

Original word vector: $z = W (x_{i-2} + x_{i-1})$

Paragraph vector:

$$z = W (x_{i-2} + x_{i-1}) + W' d$$

Paragraph Vector

Le, Quoc, and Tomas Mikolov. "Distributed Representations of Sentences and Documents." ICML, 2014

Original word vector:

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

Paragraph vector:

$$z = W (x_{i-2} + x_{i-1}) + W' d$$

Then error of the prediction can be explained by the meaning of the paragraphs.

Paragraph d_1: (The paragraph is related to "The lord of the ring")

…… 魔君 名叫 索倫 (Sauron) ……

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

$$z = W (x_{i-2} + x_{i-1}) + W' d_1$$

different

Paragraph d_2: (The document is related to "仙五")

…… 魔君 名叫 姜世離 ……

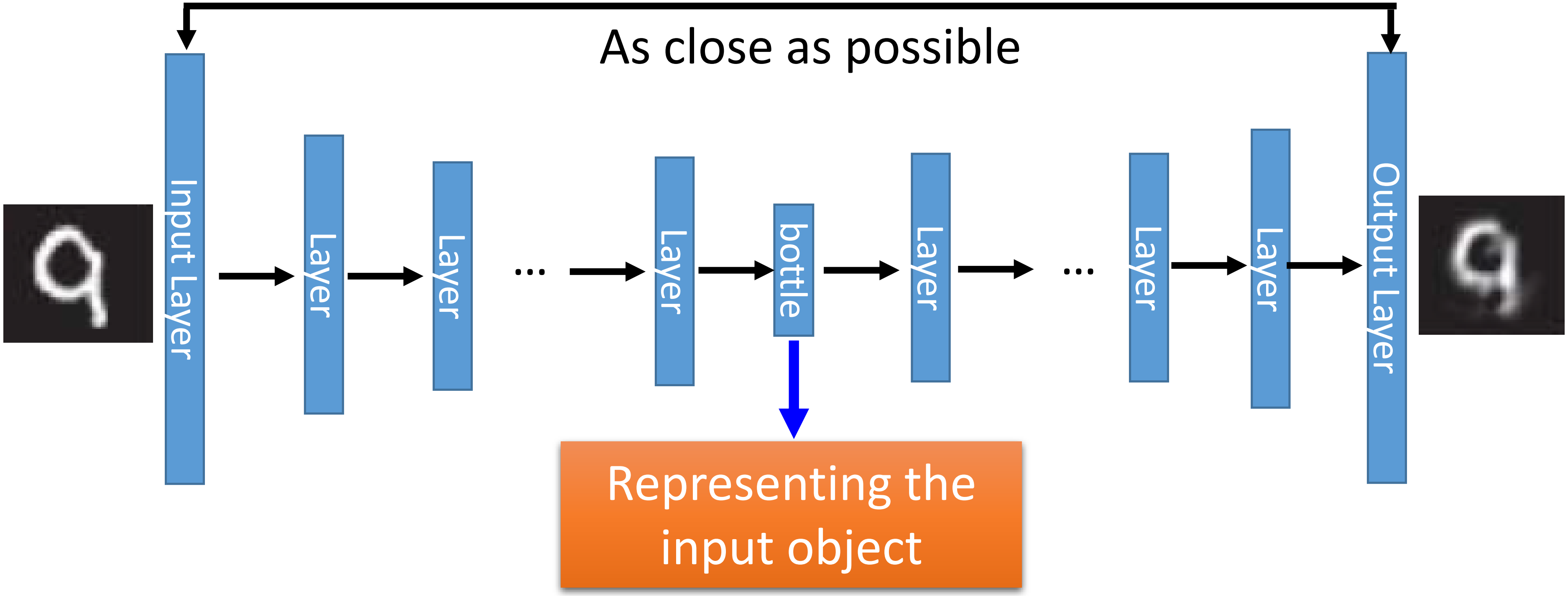
w_{i-2} w_{i-1} w_i

$$z = W (x_{i-2} + x_{i-1}) + W' d_2$$

Paragraph vector of d:V(d) = $W' d$ ➡ Meaning of the paragraph

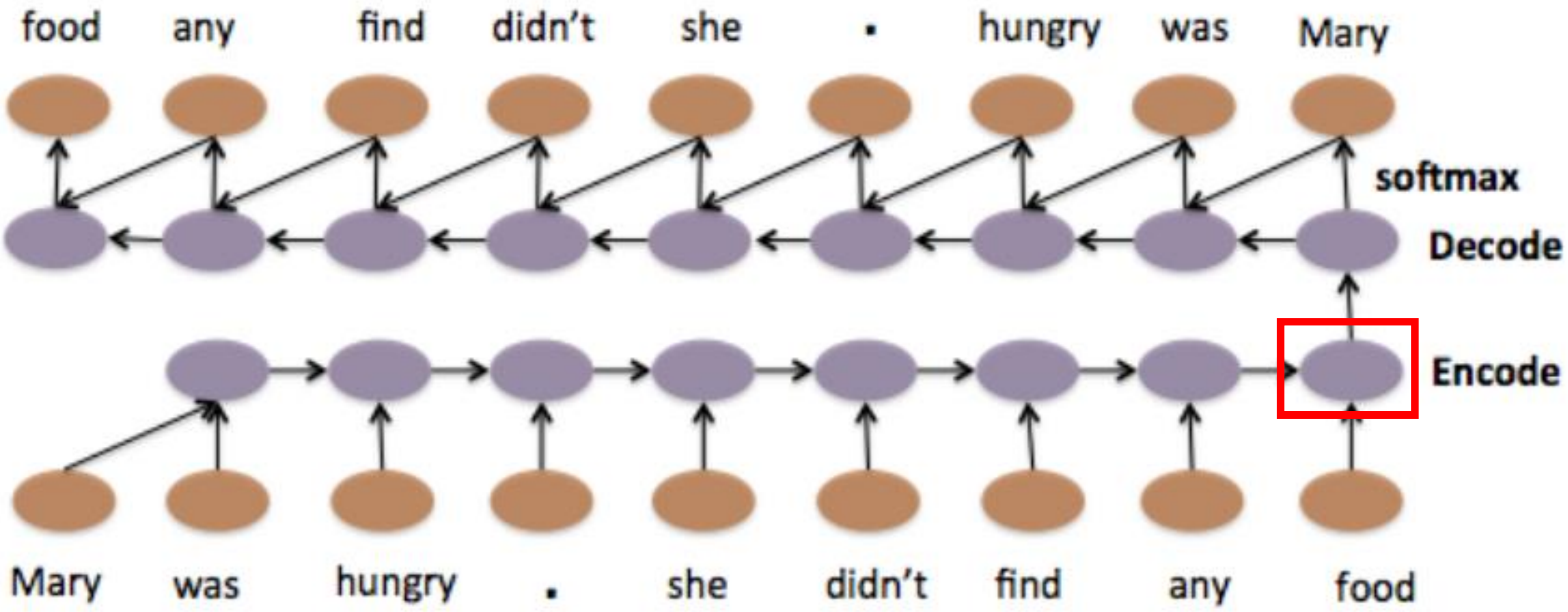
Sequence-to-sequence Auto-encoder

- Original Auto-encoder



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

Sequence-to-sequence Auto-encoder



Li, Jiwei, Minh-Thang Luong, and Dan Jurafsky. "A hierarchical neural autoencoder for paragraphs and documents." *arXiv preprint arXiv:1506.01057*(2015).

Summary

Deep Structured Semantic Model (DSSM)	• Application: Information Retrieval (IR)
Recursive Neural Network	• Application: Sentiment Analysis, Sentence Relatedness
Unsupervised	• Paragraph Vector • Sequence-to-sequence auto-encoder