



Natural Language Processing with Deep Learning

BÀI 4: VECTOR SPACE & DIMENSIONALITY REDUCTION



<https://csc.edu.vn/data-science-machine-learning/natural-language-processing-with-deep-learning> 293





VECTOR SPACE & DIMENSIONALITY REDUCTION

I. Ngôn ngữ trong NLP & Word Vector

II. Mô hình Word2vec

III. Mô hình GloVe

IV. Neural Network Classifier

Ngôn ngữ trong NLP & Word Vector



Làm thế nào để biểu diễn các từ?

N-gram language model

Ví dụ:

It is 76 C and .

$P(w \mid \text{it is } 76 \text{ F and})$

[0.0001, 0.1, 0, 0, 0.002, ..., 0.3, ..., 0]

Text classification

I like this book. 

I don't like this book at all.

$$P(y = 1 | x) = \sigma(\theta^T w + b)$$

$$w(1) = [0, 1, 0, 0, 0, \dots, 1, \dots, 1]$$

$$w(2) = [0, 1, 0, 1, 0, \dots, 1, \dots, 1]$$

don't





Ngôn ngữ trong NLP & Word Vector

Danh sách các bộ từ đồng nghĩa (*synonym*) và trường từ vựng (*hypernym*) có “is a” relationships. VD: WordNet.

Bộ từ đồng nghĩa với từ “good”:

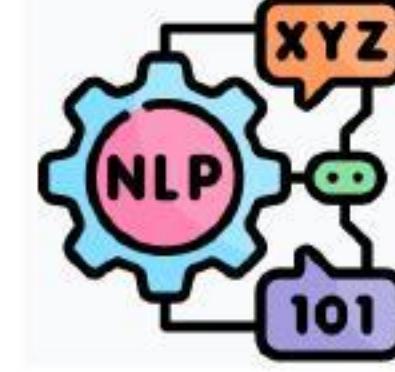
```
from nltk.corpus import wordnet as wn
poses = { 'n' : 'noun', 'v' : 'verb', 's' : 'adj (s)', 'a' : 'adj', 'r' : 'adv'}
for synset in wn.synsets("good"):
    print("{}: {}".format(poses[synset.pos()],
                          ", ".join([l.name() for l in synset.lemmas()])))
```

```
noun: good
noun: good, goodness
noun: good, goodness
noun: commodity, trade_good, good
adj: good
adj (sat): full, good
adj: good
adj (sat): estimable, good, honorable, respectable
adj (sat): beneficial, good
adj (sat): good
adj (sat): good, just, upright
...
adverb: well, good
adverb: thoroughly, soundly, good
```

Trường từ vựng của từ “panda”:

```
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')]
```



Ngôn ngữ trong NLP & Word Vector

Vấn đề của WordNet

Hữu dụng nhưng thiếu sắc thái nghĩa.

- “proficient” đồng nghĩa với “good” chỉ đúng trong 1 số ngữ cảnh.
- Không có sự bao quát về ý nghĩa hoặc sự thích hợp của từng từ (ví dụ: từ đồng nghĩa có tính xúc phạm).

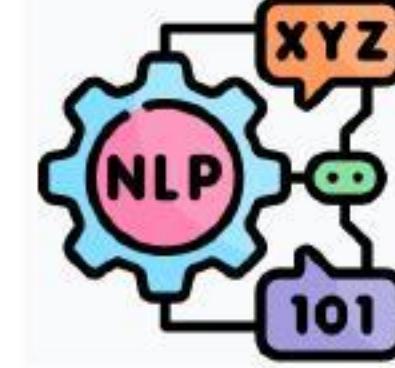
Thiếu ý nghĩa mới cho các từ, khó cập nhật.

- Các từ mới như wicked, badass, genius

Chỉ mang tính tương đối

Cần con người thao tác

Không thể dùng để tính toán chính xác word similarity.



Ngôn ngữ trong NLP & Word Vector

Biểu diễn từ bằng discrete symbol

Trong NLP truyền thống, chúng ta còn sử dụng discrete symbol để biểu diễn từ.

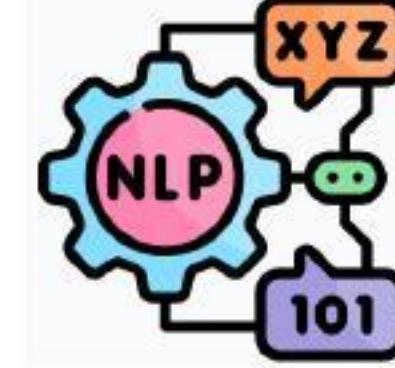
hotel, conference, motel – a localist representation

Các discrete symbol là các one-hot vector (1 giá trị là 1, còn lại là 0):

motel = [0 0 0 0 0 0 0 0 0 1 0 0 0]

hotel = [0 0 0 0 0 0 1 0 0 0 0 0]

Vector dimension = số từ trong từ điển (VD: Tiếng Anh có 500,000+ từ) → **Encode word similarity cho những vector này là bất khả thi.**



Ngôn ngữ trong NLP & Word Vector

Vấn đề của Discrete symbol

Sử dụng **vector một chiều** để biểu diễn các từ.

- không có tính tương đồng.
- Mã hóa độ tương đồng giữa các vector (**word similarity**).

VD: Nếu tìm kiếm “*Saigon motel*” trên internet, chúng ta cũng muốn tìm kiếm các tài liệu có chứa “*Saigon hotel*”.

motel = [0 0 0 0 0 0 0 0 0 1 0 0 0]

hotel = [0 0 0 0 0 0 1 0 0 0 0 0 0]

Hai vector *motel* và *hotel* là trực giao, có nghĩa là chúng không có bất kỳ điểm chung nào.



Ngôn ngữ trong NLP & Word Vector

Biểu diễn từ bằng ngũ cảnh

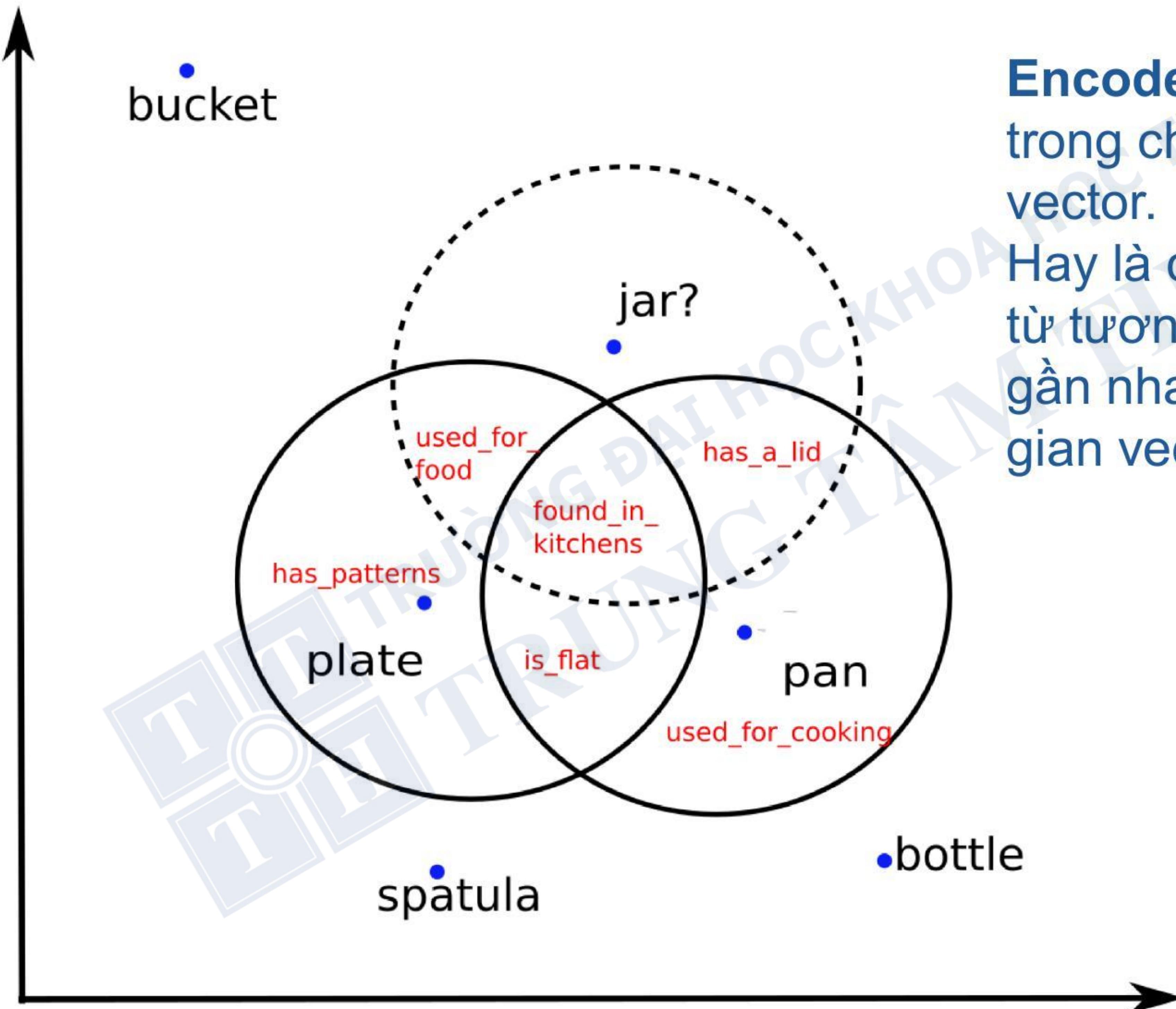
Distributional hypothesis - Giả thiết phân phối: các từ xuất hiện cùng các ngũ cảnh có xu hướng sở hữu ý nghĩa tương tự nhau.

“Ta có thể hiểu được nghĩa một từ bằng cách nhìn vào các từ đi theo nó.”

- J.R.Firth 1957



Ngôn ngữ trong NLP & Word Vector



Encode word similarity

trong chính các word vector.

Hay là các vector của các từ tương đồng nhau sẽ ở gần nhau trong không gian vector.



Word Vector

- C1: A bottle of ____ is on the table.
- C2: Everybody likes ____.
- C3: Don't have ____ before you drive.
- C4: We make ____ out of corn.

	C1	C2	C3	C4
tejuino	1	1	1	1
loud	0	0	0	0
motor-oil	1	0	0	0
tortillas	0	1	0	1
choices	0	1	0	0
wine	1	1	1	0



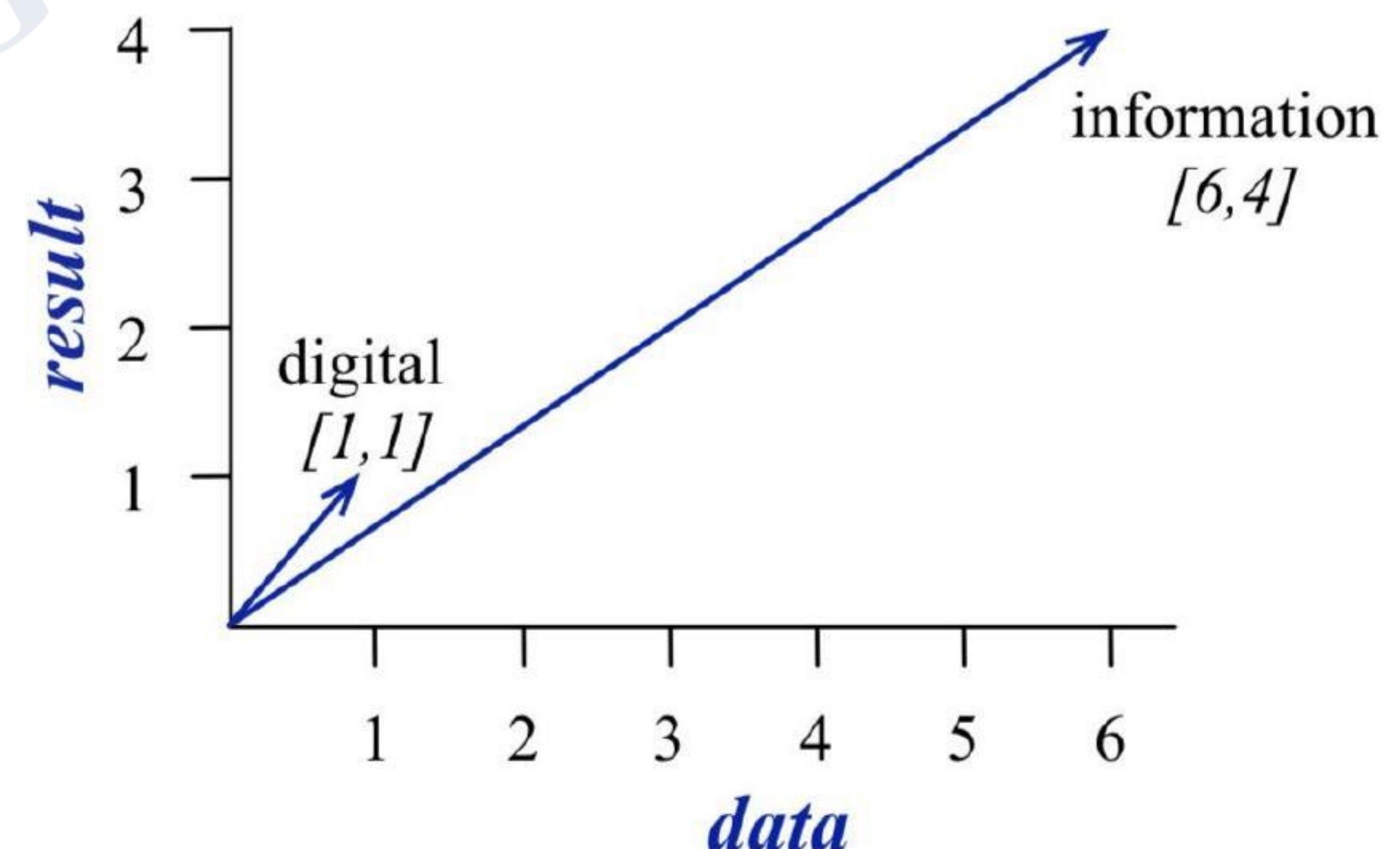
Word Vector

Giải pháp: Sử dụng context vector biểu diễn ý nghĩa từ.

Word-word Co-occurrence Matrix

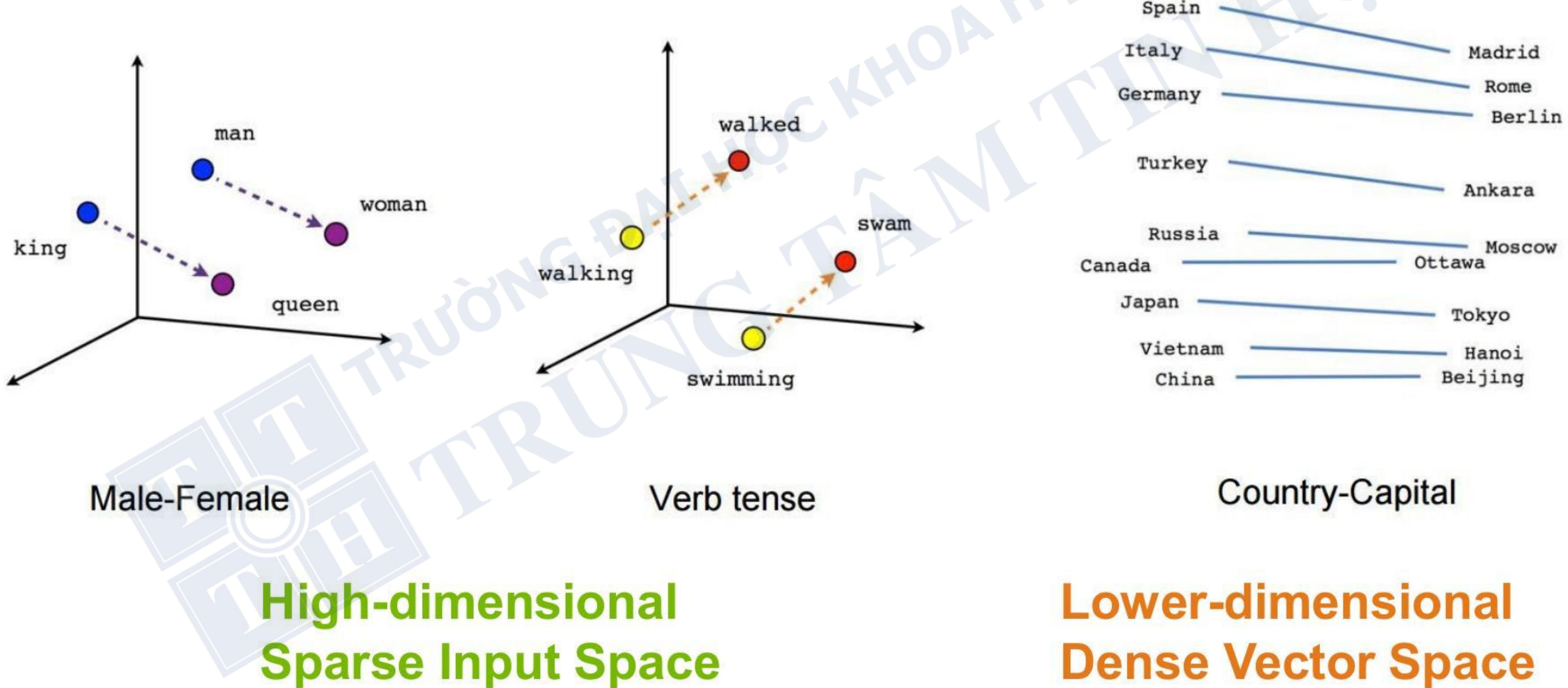
	Computer	Data	Pinch	Result	Sugar
apricot	0	0	1	0	1
apple	0	0	1	0	1
digital	2	1	0	1	0
information	1	6	0	4	0

Mỗi từ là một vector.
Các từ tương đồng có
vị trí gần nhau trong
không gian vector.



Tại sao dùng Word Embeddings?

Word embeddings có thể nắm bắt được cấu trúc quan hệ phong phú của từ vựng.





Word Embeddings

Tạo **dense vector** cho mỗi từ, sao cho tương ứng với các vector của các từ xuất hiện trong bối cảnh tương tự, tính độ tương đồng bằng tích vô hướng (scalar product) của các vector.

banking =

0.286
0.792
-0.177
-0.107
0.109
-0.542
0.349
0.271

monetary =

0.413
0.582
-0.007
0.247
0.216
-0.718
0.147
0.051

Word vector - word embedding neural word representation, được biểu diễn theo dạng phân phối.



Word Embeddings

expect =

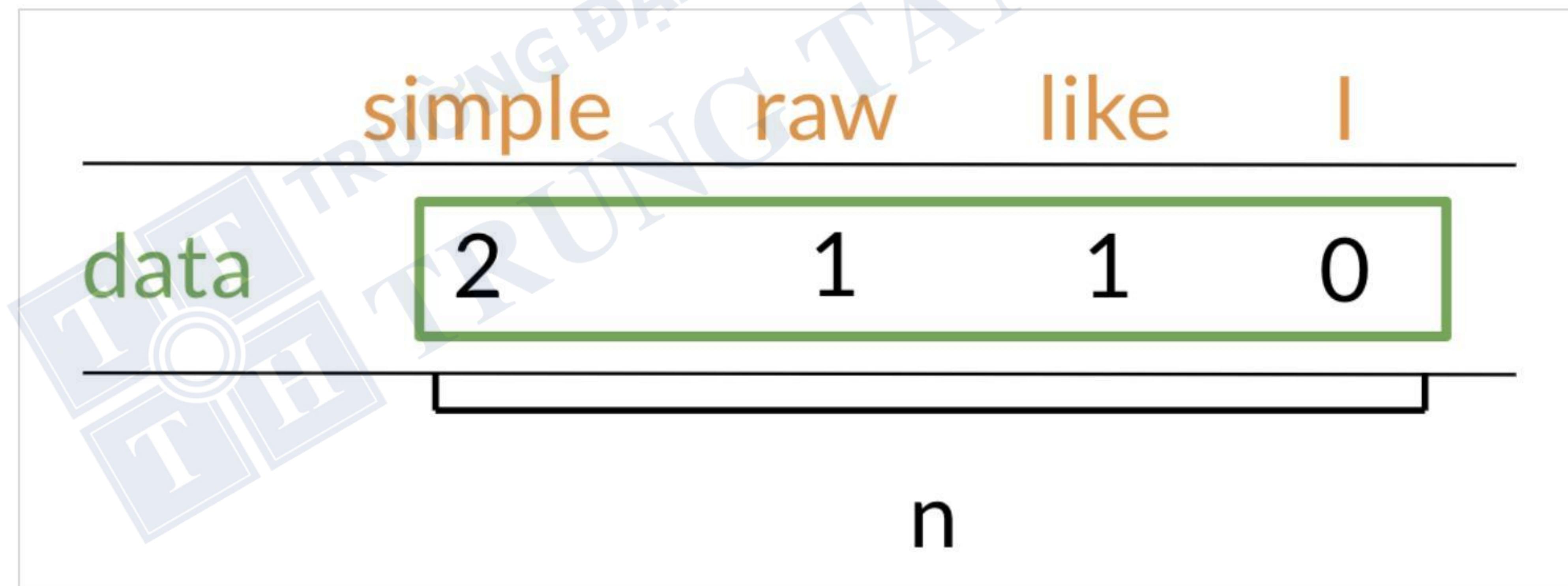
$$\begin{bmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 0.109 \\ -0.542 \\ 0.349 \\ 0.271 \\ 0.487 \end{bmatrix}$$



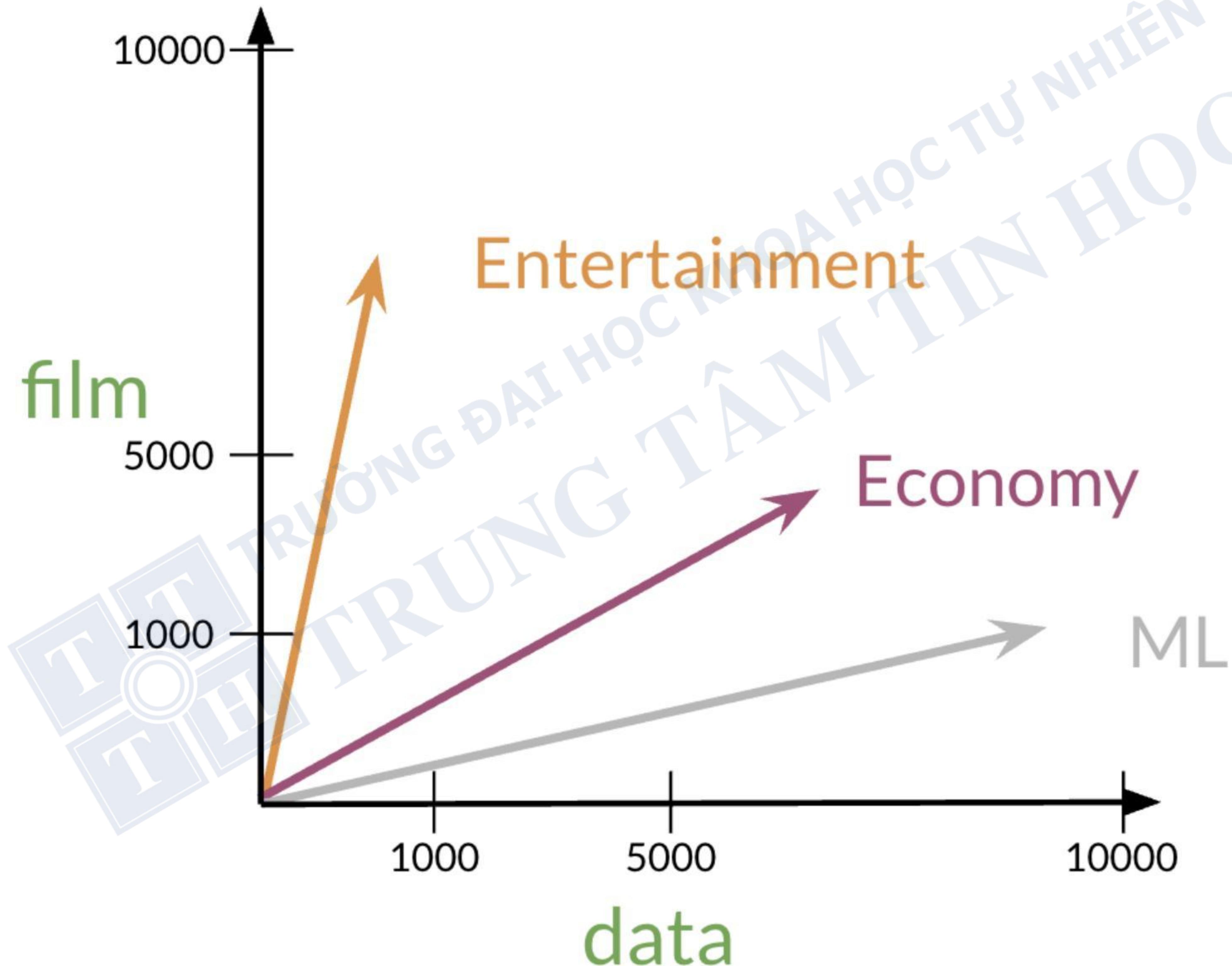

Word by Word Design

I like simple data

I prefer simple raw data



Word by Document Design





Word by Document Design

	Entertainment	Economy	ML
data	500	6620	9320
film	7000	4000	1000

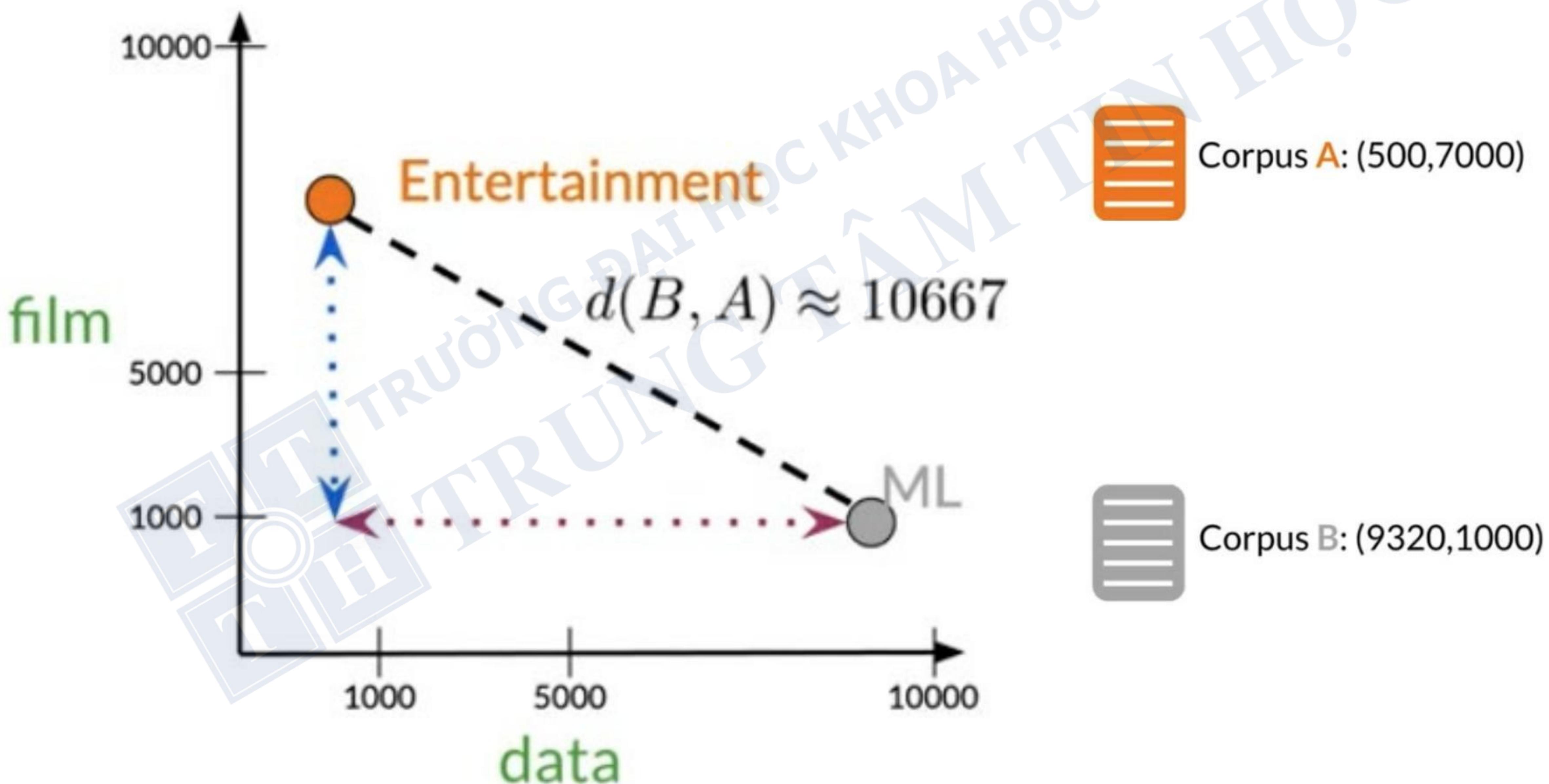
Phân loại *Entertainment* có $v = [500, 7000]$

Measures of “similarity”:

Angle
Distance

Euclidian Distance

$$d(B, A) = \sqrt{((B_1 - A_1)^2 + (B_2 - A_2)^2)}$$





Euclidian Distance

Công thức tính khoảng cách với vector có n chiều

$$d(\vec{v}, \vec{w}) = \sqrt{\sum_{i=1}^n (v_i - w_i)^2}$$

$$= \sqrt{(1 - 0)^2 + (6 - 4)^2 + (8 - 6)^2}$$

$$= \sqrt{1 + 4 + 4} = \sqrt{9} = 3$$

	data	\vec{w}	\vec{v}
AI	6	0	1
drinks	0	4	6
food	0	6	8

Principal Component Analysis - PCA

Visualization of word vectors

	$d > 2$		
	0.20	...	0.10
oil	0.20	...	0.10
gas	2.10	...	3.40
city	9.30	...	52.1
town	6.20	...	34.3



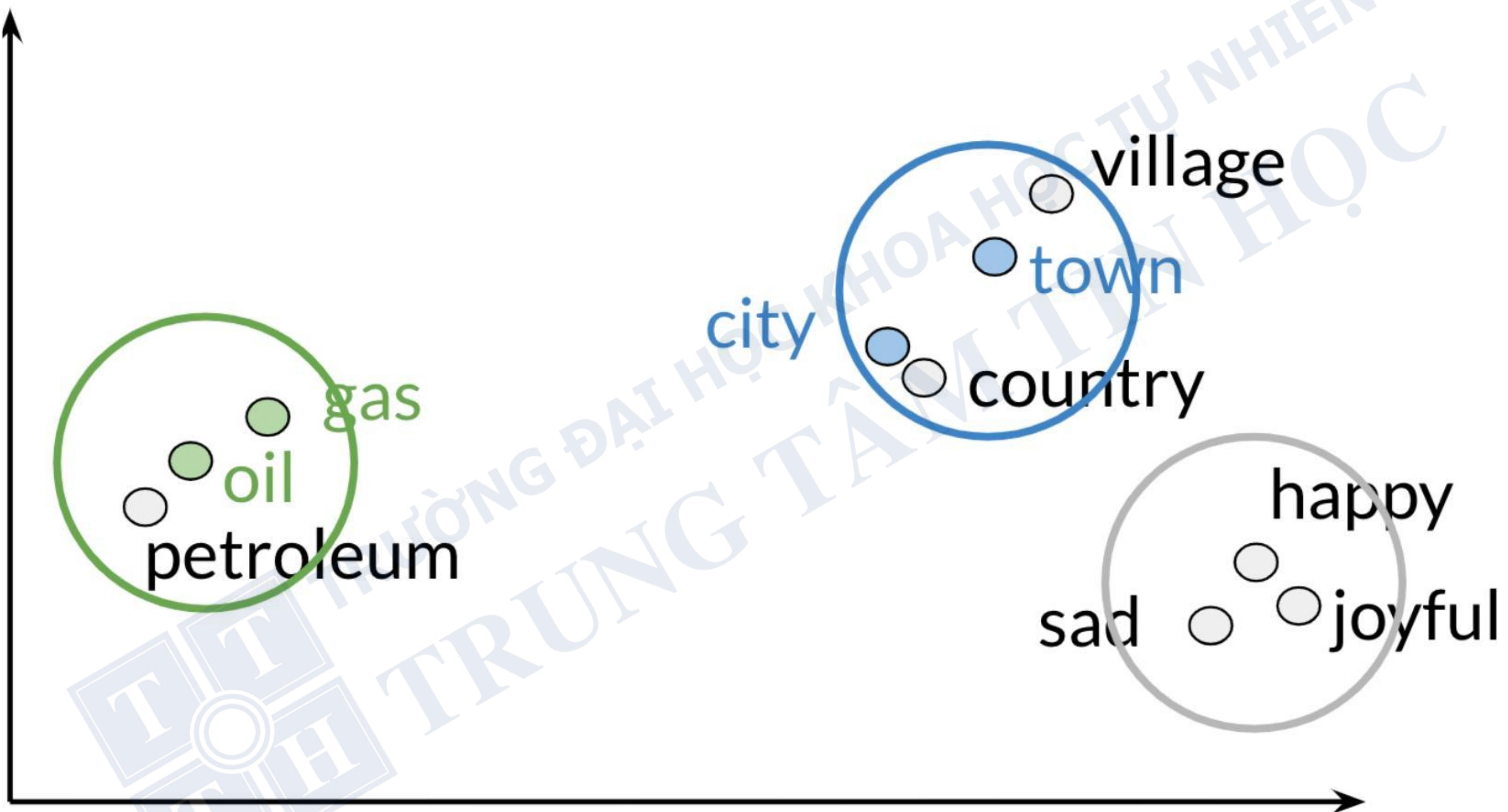
oil & gas



town & city

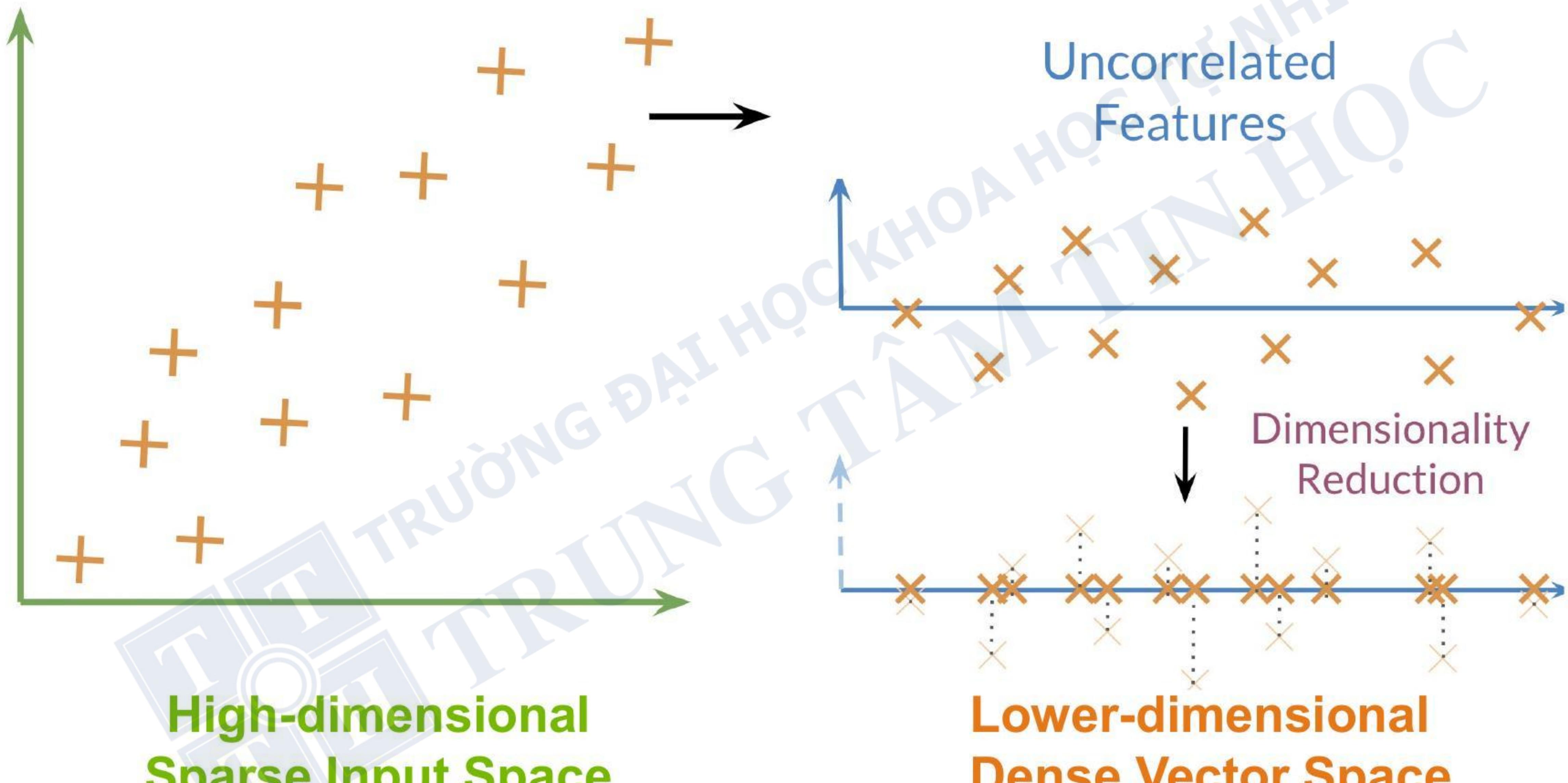
How can you visualize if your representation captures these relationships?

Principal Component Analysis - PCA



Biểu diễn các word vector

Principal Component Analysis - PCA

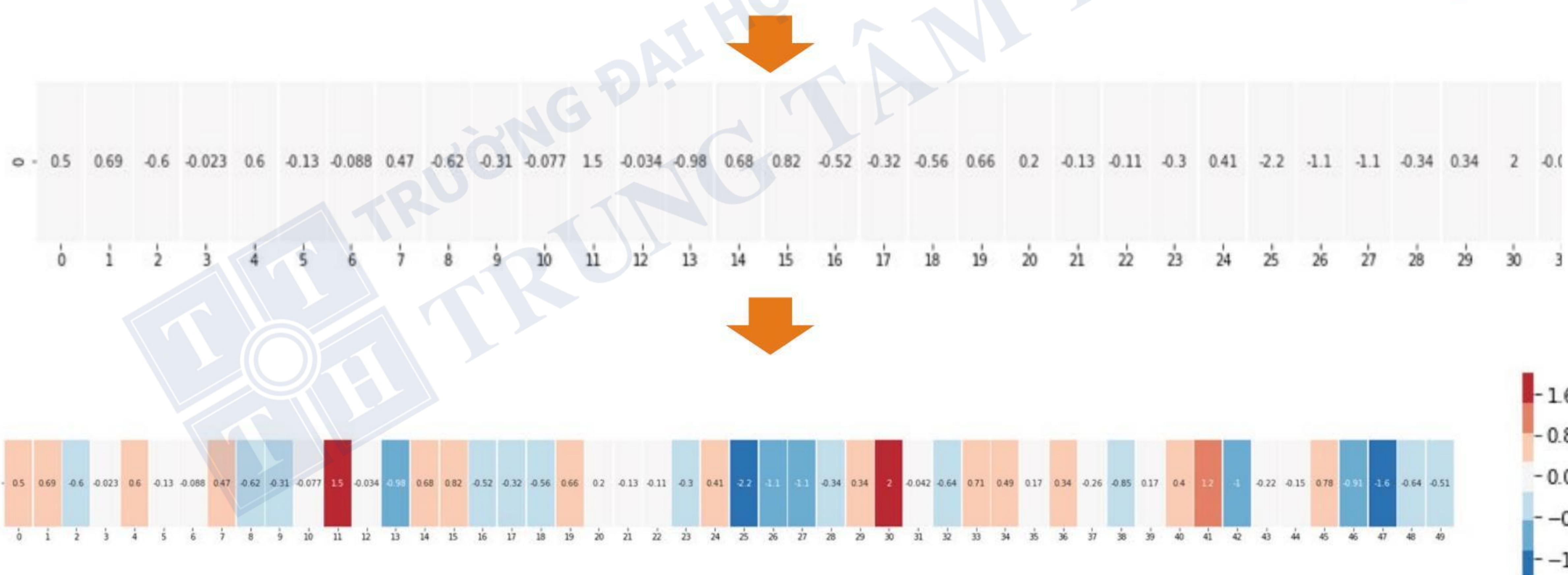




Ví dụ Word Embedding

Word embedding cho từ “King” (GloVe vector/ Wikipedia)

```
[ 0.50451, 0.68607, -0.59517, -0.022801, 0.60046, -0.13498, -0.08813, 0.47377, -0.61798, -0.31012, -0.076666, 1.493, -0.034189, -0.98173, 0.68229, 0.81722, -0.51874, -0.31503, -0.55809, 0.66421, 0.1961, -0.13495, -0.11476, -0.30344, 0.41177, -2.223, -1.0756, -1.0783, -0.34354, 0.33505, 1.9927, -0.04234, -0.64319, 0.71125, 0.49159, 0.16754, 0.34344, -0.25663, -0.8523, 0.1661, 0.40102, 1.1685, -1.0137, -0.21585, -0.15155, 0.78321, -0.91241, -1.6106, -0.64426, -0.51042 ]
```



Ví dụ Word Embedding



Tương tự, word embedding cho từ “king”, “man” và “woman”.

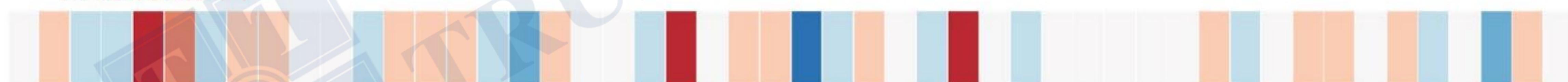
“king”



“Man”



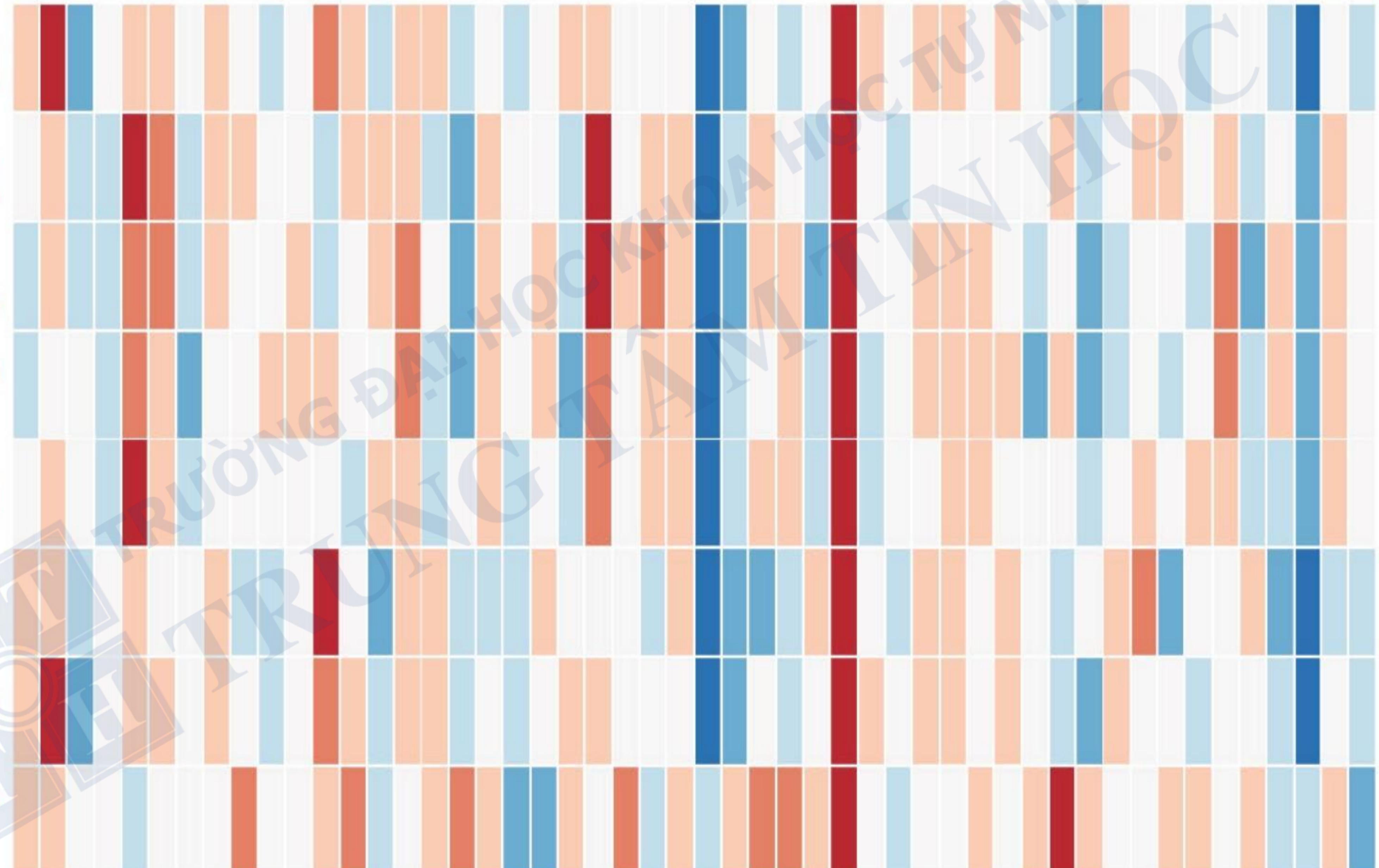
“Woman”





Ví dụ Word Embedding

queen
woman
girl
boy
man
king
queen
water

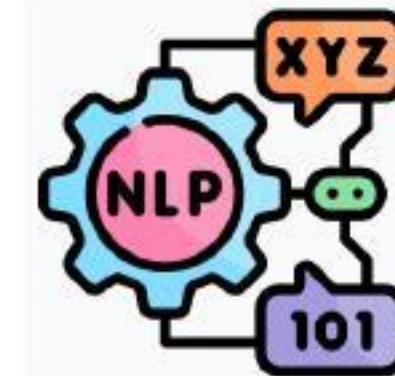




Ví dụ Word Embedding

king - man + woman \approx queen





VECTOR SPACE & DIMENSIONALITY REDUCTION

I. Ngôn ngữ trong NLP & Word Vector

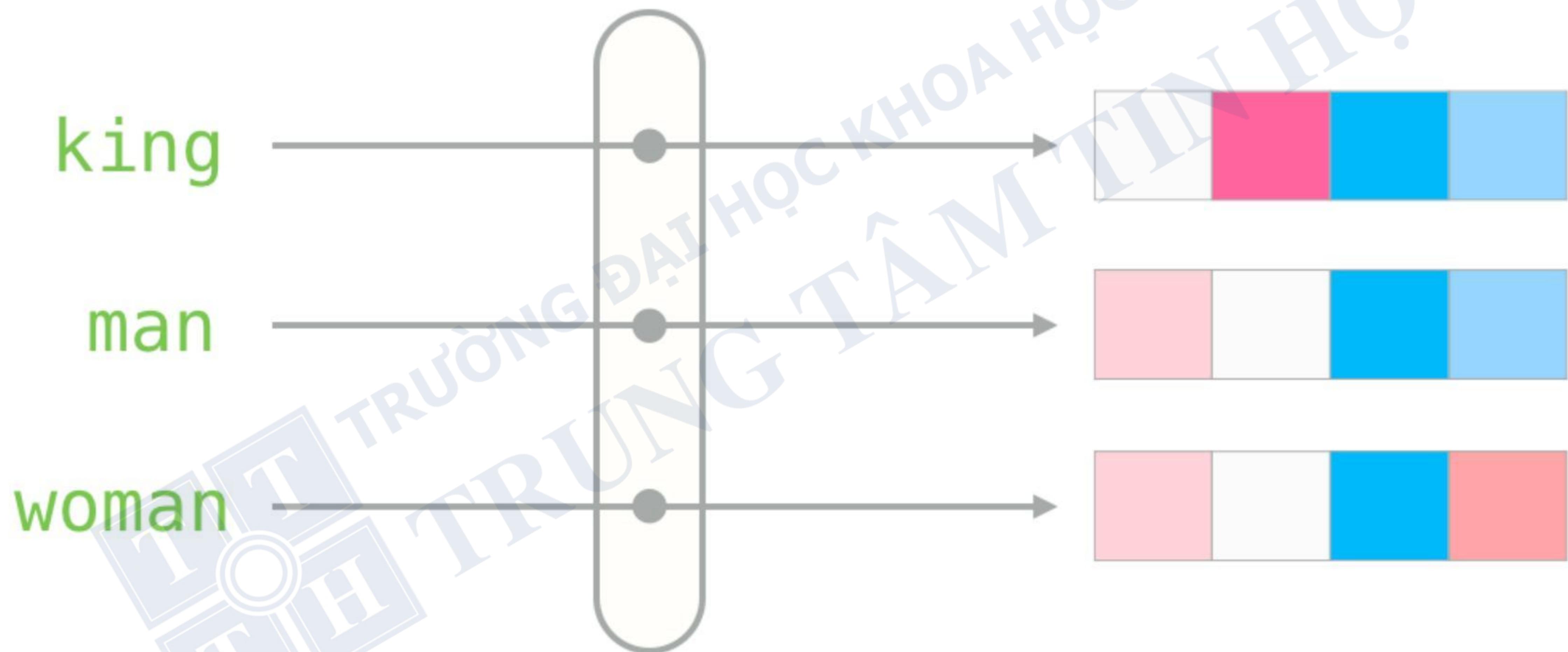
II. Mô hình Word2vec

III. Mô hình GloVe

IV. Neural Network Classifier

Mô hình Word2vec

Là một mô hình trong NLP dùng để biểu diễn từ dưới dạng các **vector**.



→ Hiểu và biểu diễn ngữ nghĩa của từ thông qua **không gian vector**.



Ưu điểm Word2vec

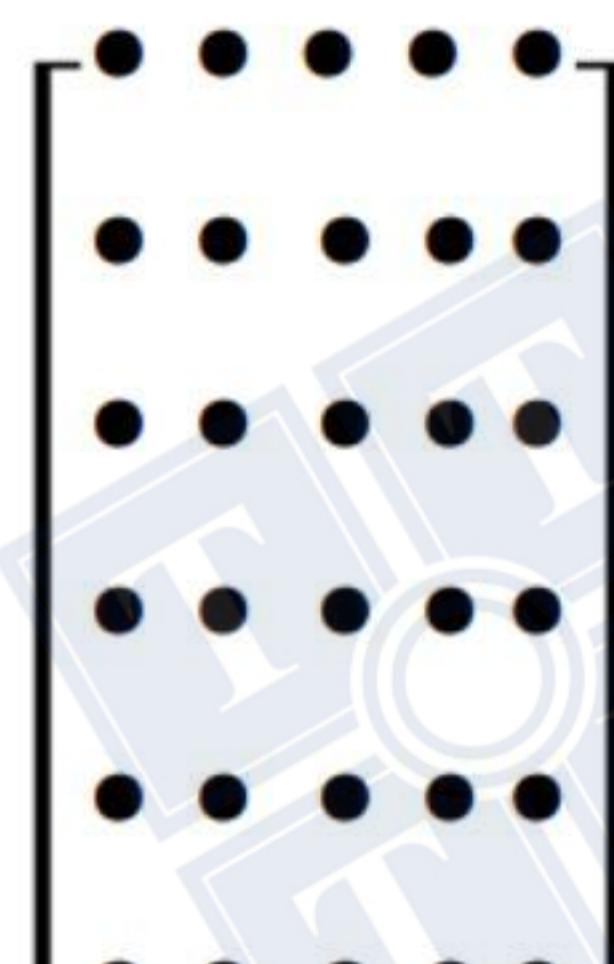
- Word2Vec dùng trong phân loại văn bản, trích xuất thông tin, dịch và tìm kiếm thông tin.
- Còn dùng để cải thiện kết quả các mô hình ML khác trong NLP.
 - **Biểu diễn từ dưới dạng vector số**: giúp mô hình hiểu và xử lý lý nghĩa của từ.
 - **Giảm kích thước từ điển** và cải thiện hiệu suất trong các tác vụ xử lý ngôn ngữ tự nhiên.
 - Word representative có tính **tương đồng ngữ nghĩa**, tăng khả năng biểu diễn và tìm kiếm thông tin.



Mô hình Word2vec

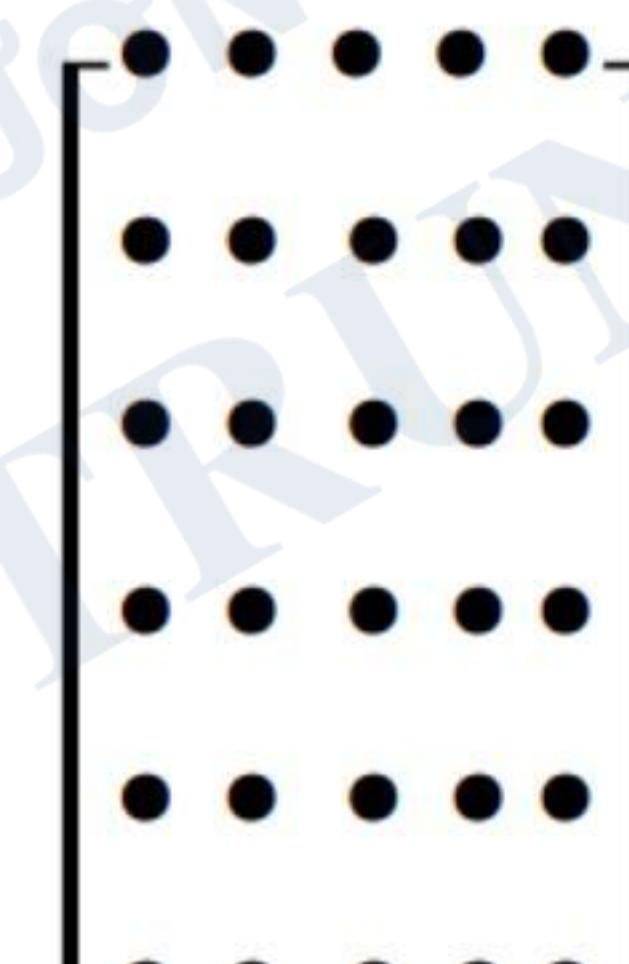
1. Bắt đầu với 1 word vector ngẫu nhiên.
2. Lặp lại từ trong scopus để lấy word position.
3. Dự đoán các từ xung quanh bằng word vector:

$$P(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w \in V} \exp(u_w^T v_c)}$$



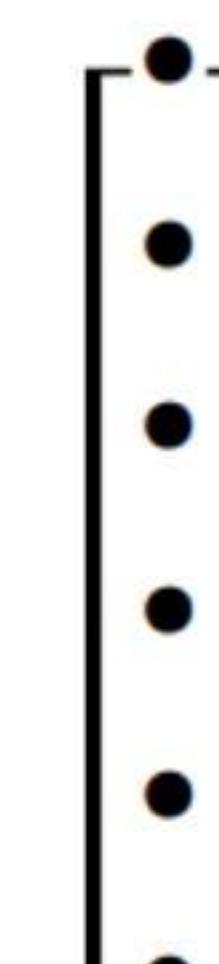
U

outside



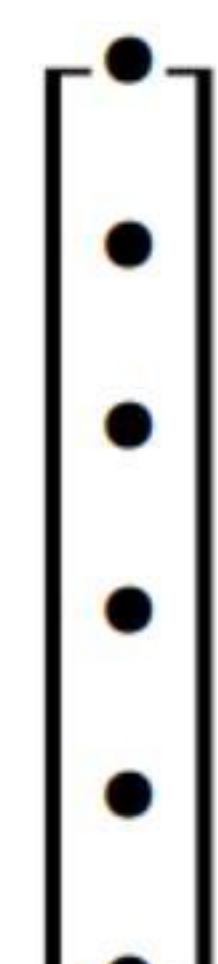
V

center



$U \cdot v_4^T$

dot product

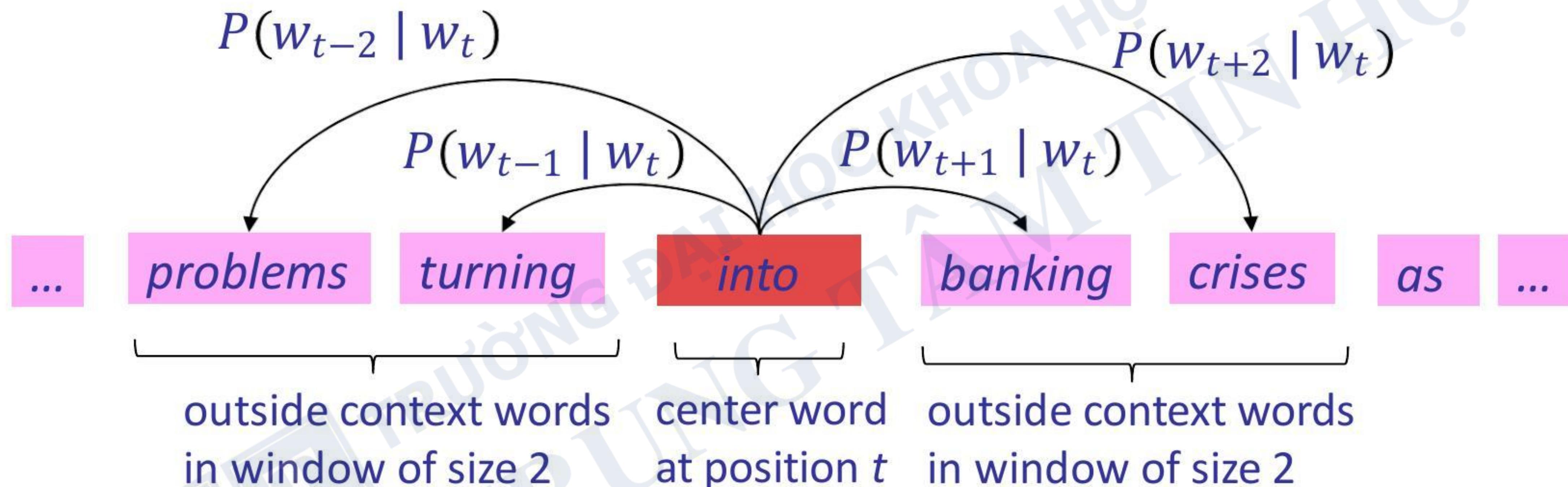


$\text{softmax}(U \cdot v_4^T)$

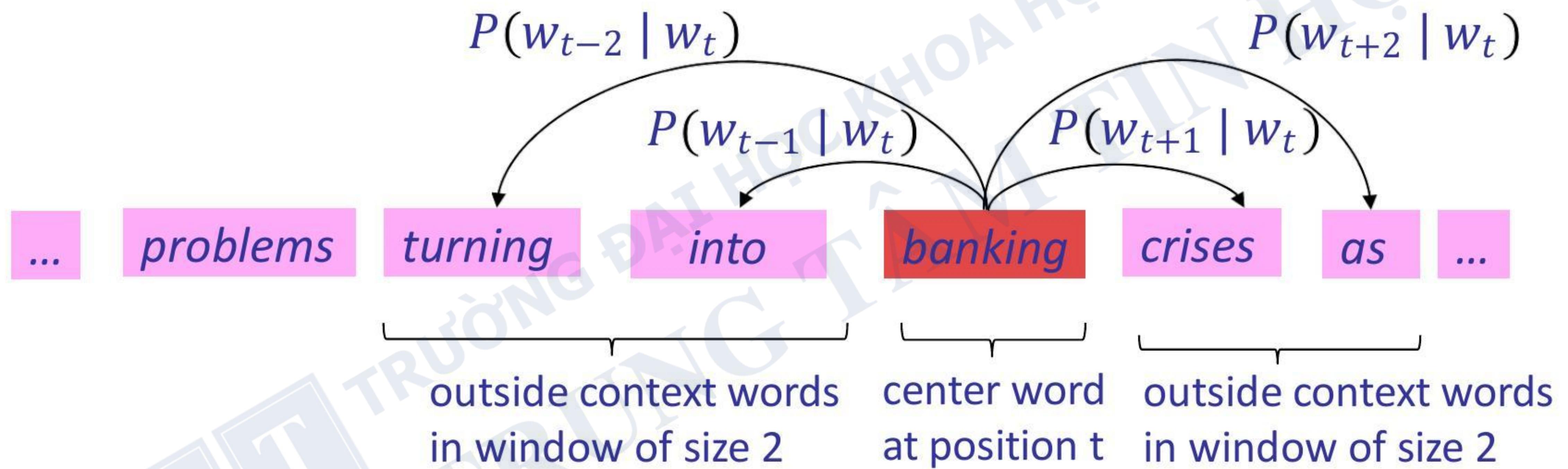
probabilities



Mô hình Word2vec



Mô hình Word2vec

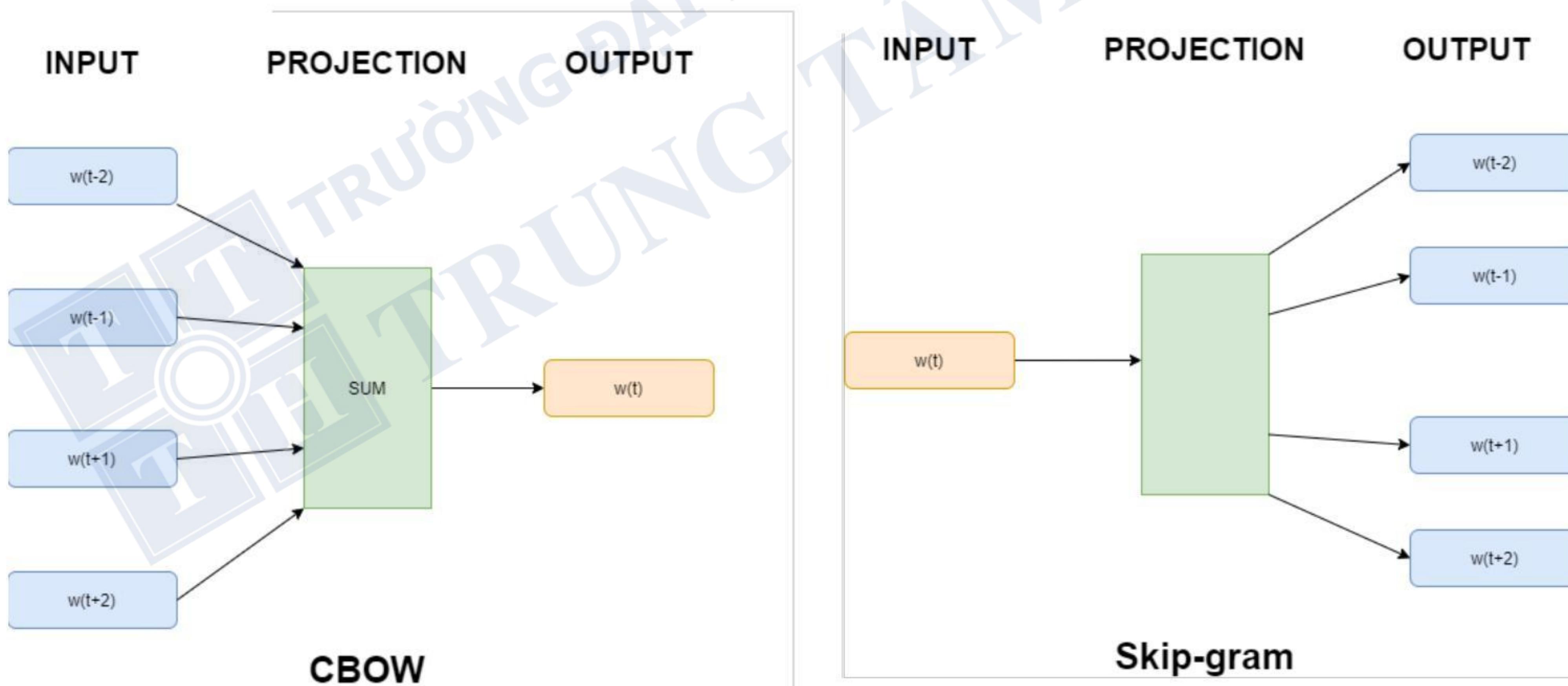




Mô hình Word2vec

Word2Vec có hai cấu trúc chính:

1. **Skip-gram** dự đoán các từ lân cận dựa trên từ input.
2. **Continuous Bag of Words (CBOW)** dự đoán từ input dựa trên các từ lân cận.





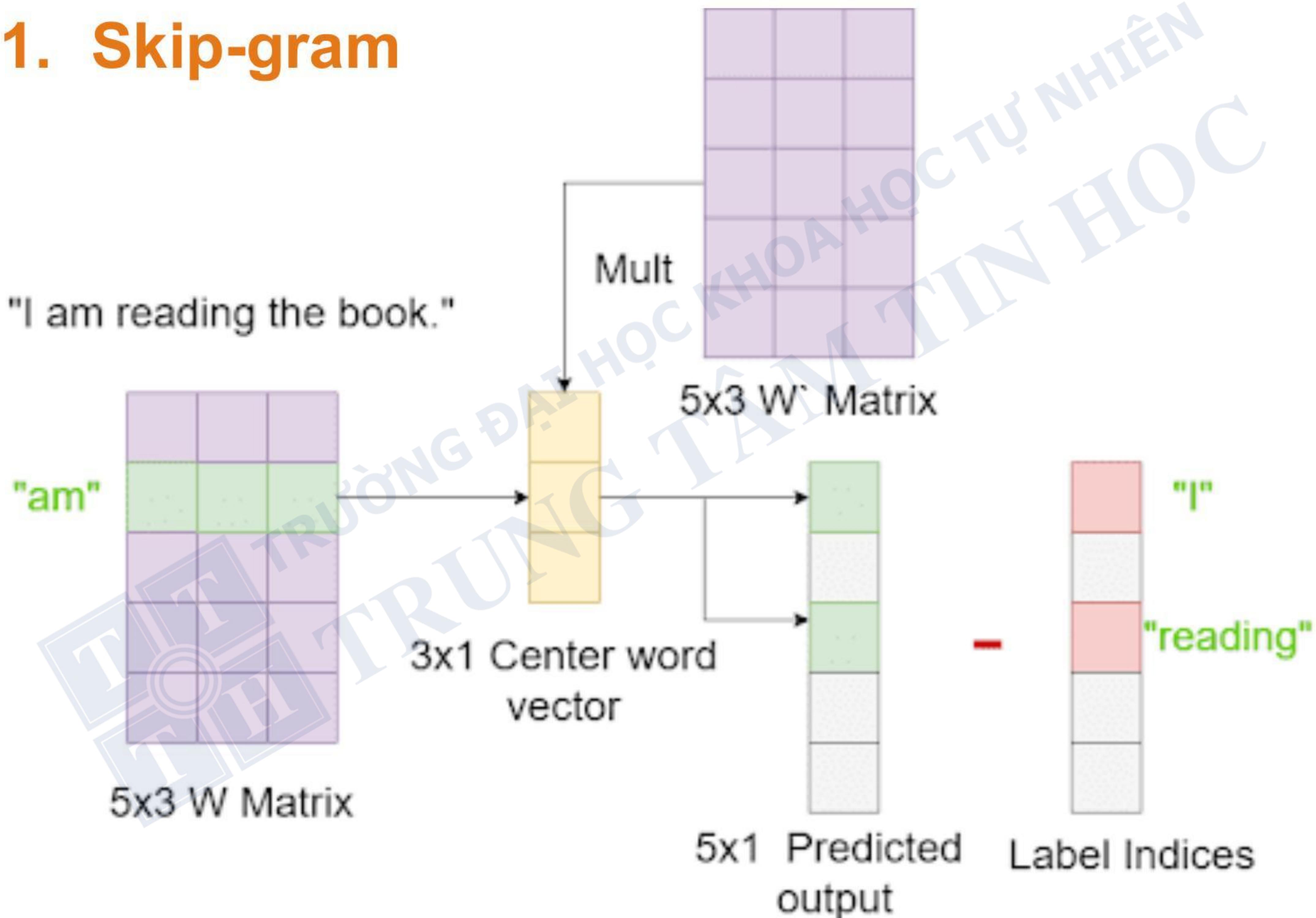
Loss function của Word2vec:

1. Naïve softmax → Đơn giản, hiệu quả khi có nhiều output class.
2. Hierarchical softmax
3. Negative sampling

Mô hình Word2vec

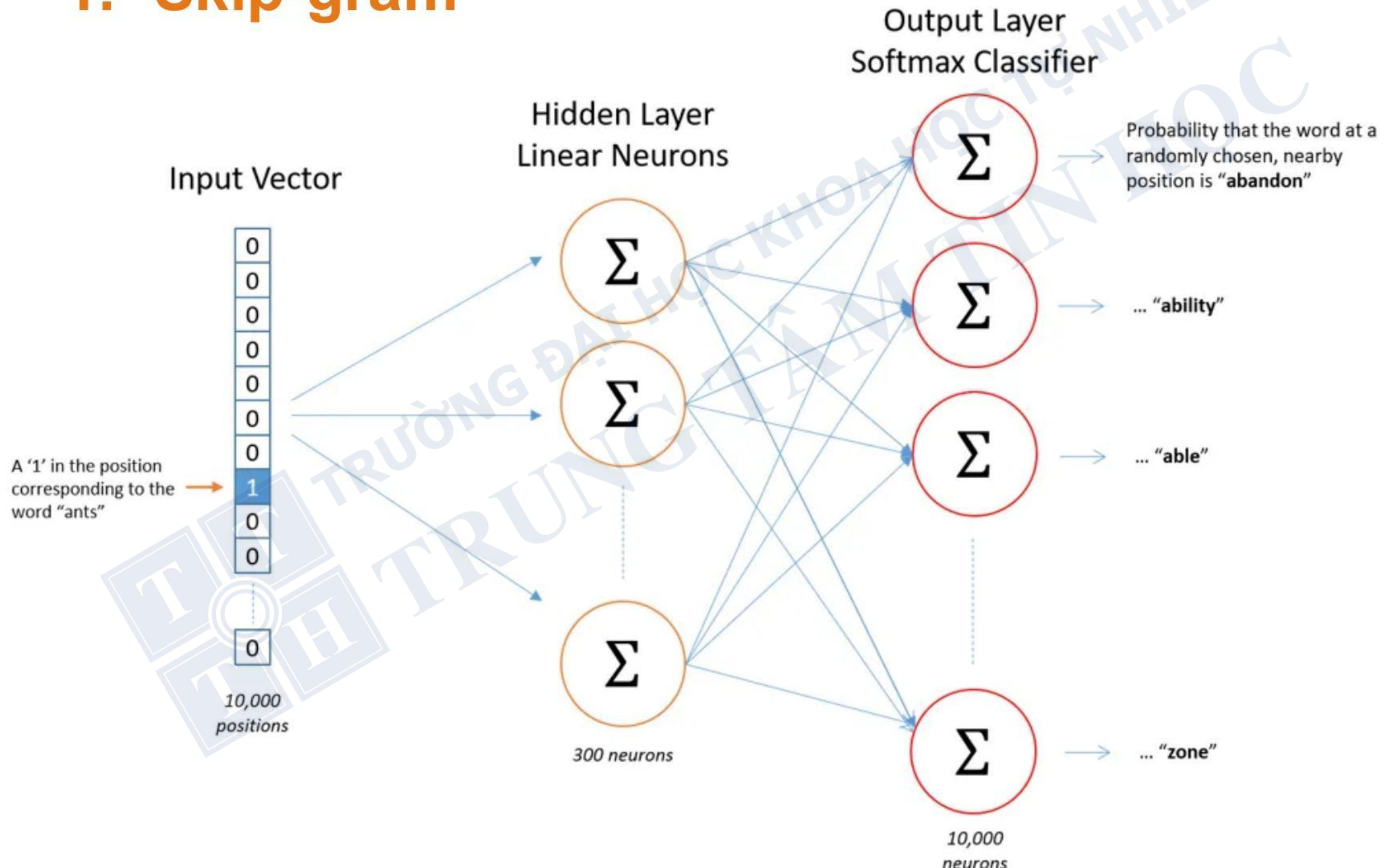
1. Skip-gram

"I am reading the book."



Mô hình Word2vec

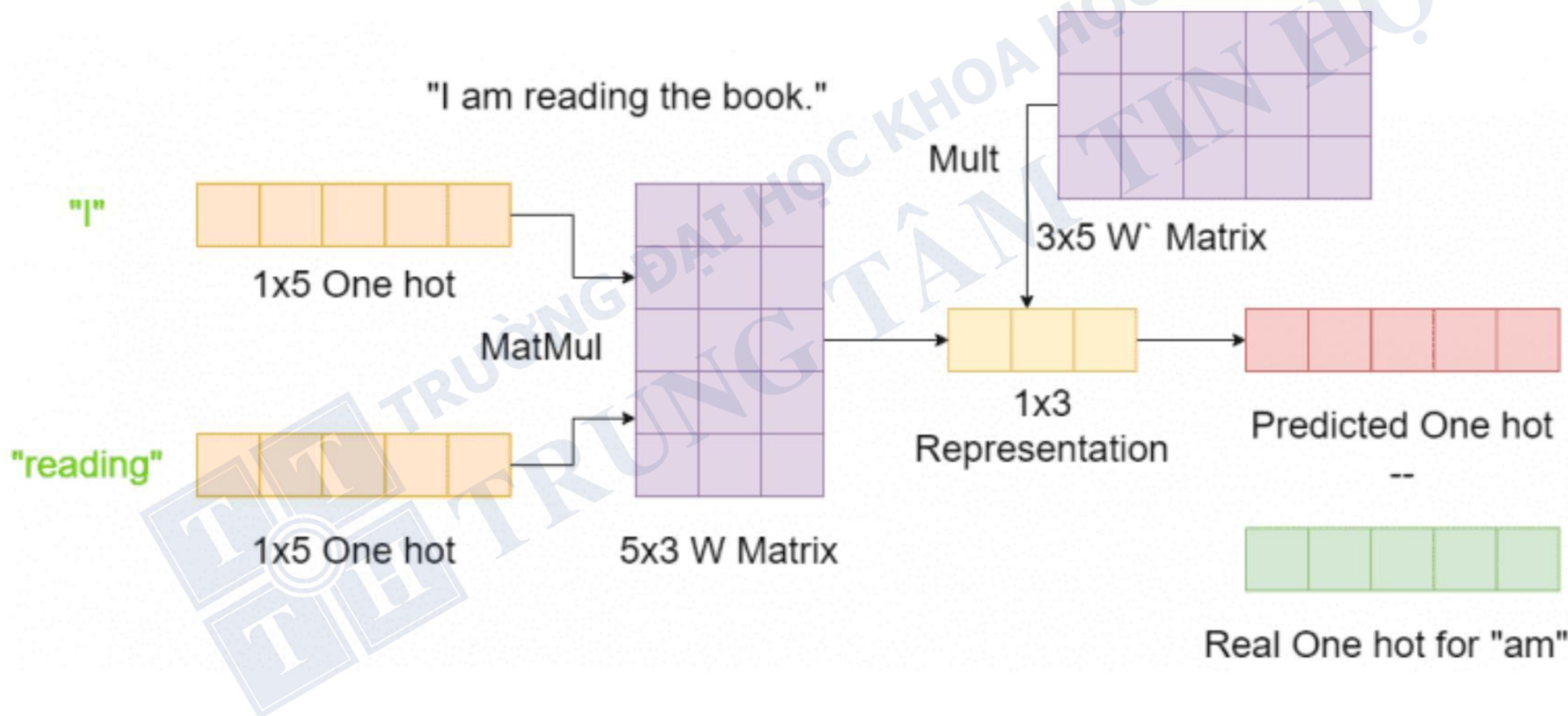
1. Skip-gram



Mô hình Word2vec

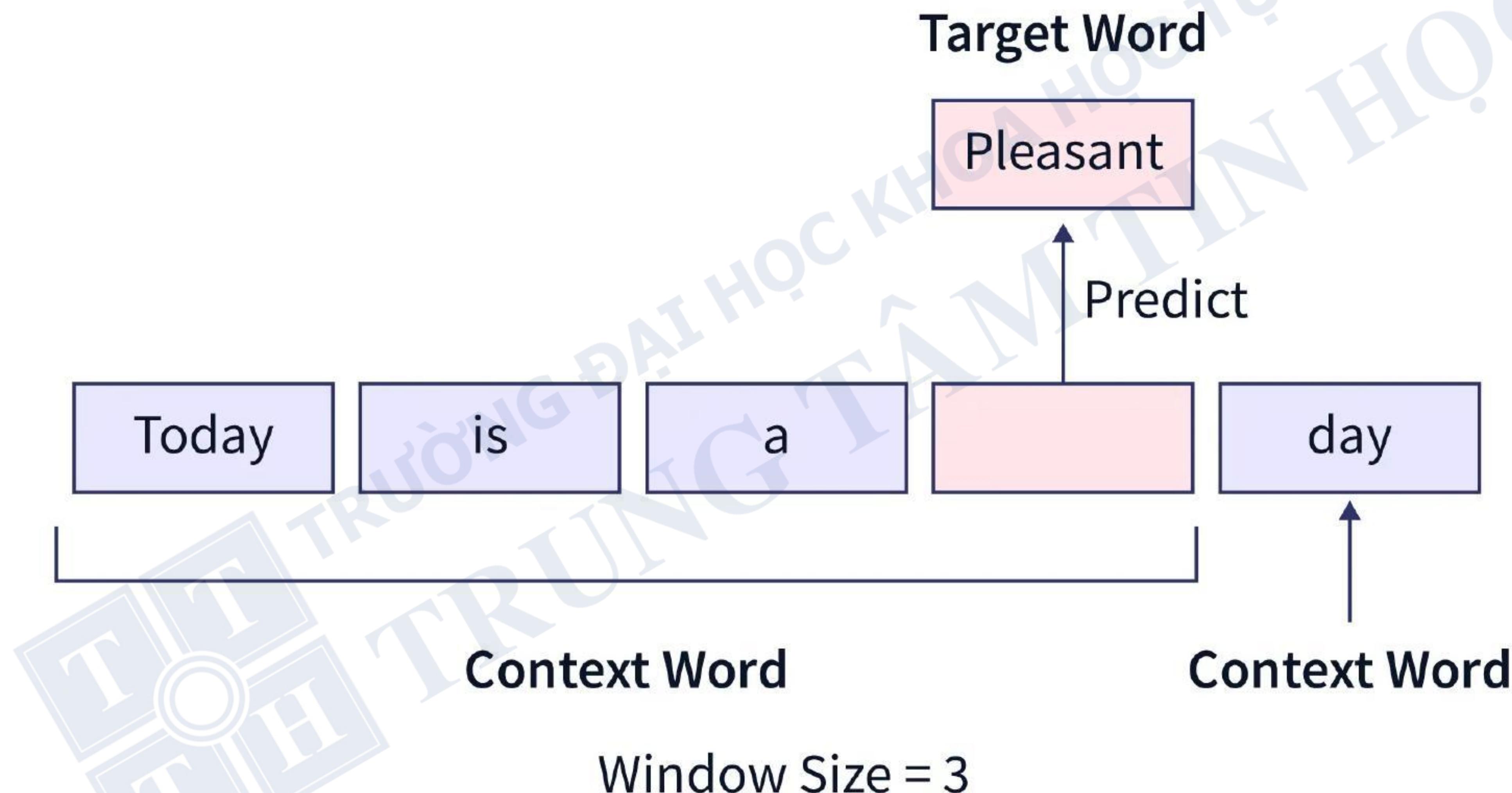


2. Continuous Bag of Words (CBOW)



Mô hình Word2vec

2. Continuous Bag of Words (CBOW)

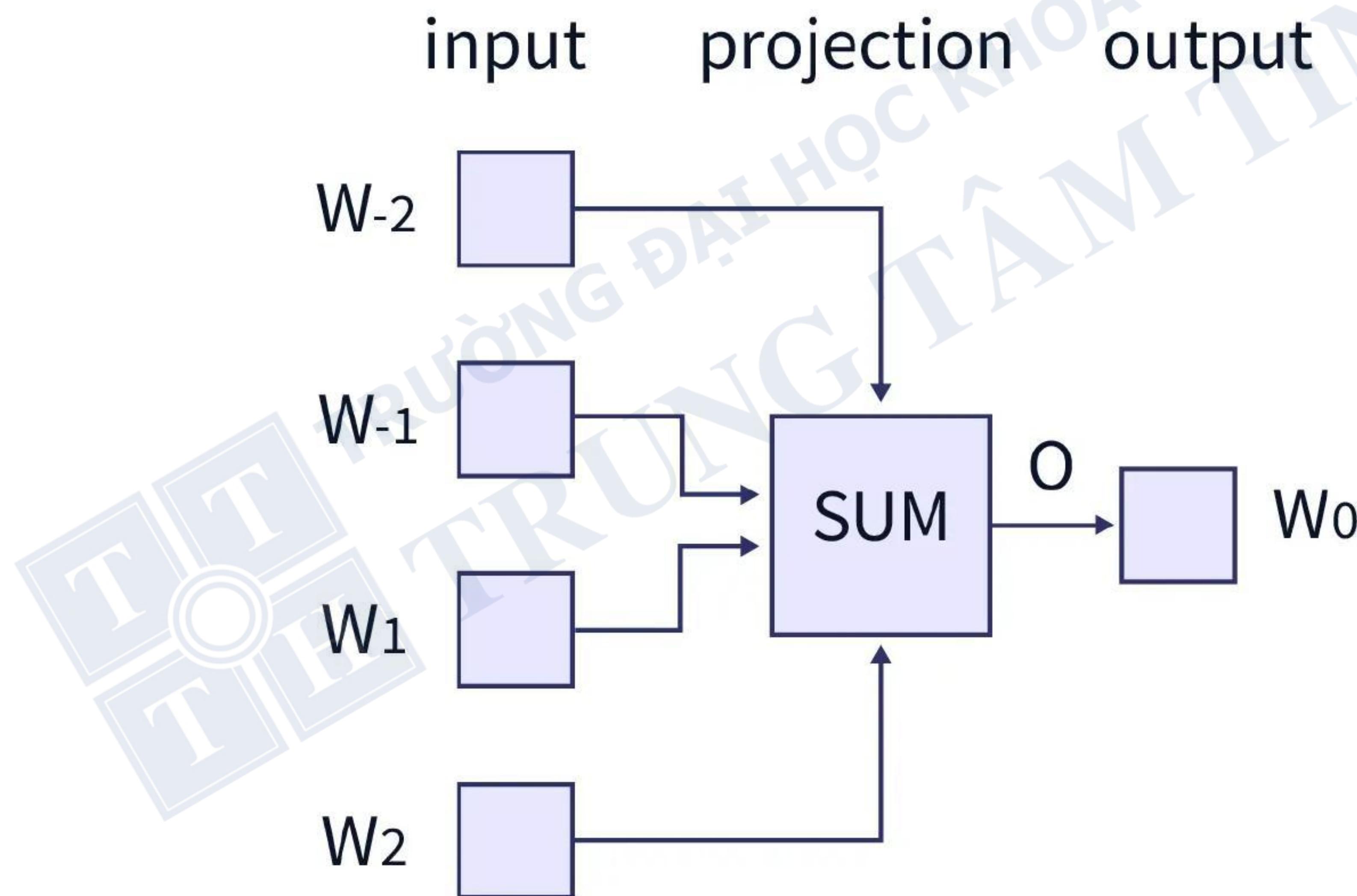


Mô hình Word2vec



2. Continuous Bag of Words (CBOW)

CBOW





Mô hình Word2vec

1. Skip-gram

A quick brown fox jumps over the lazy dog

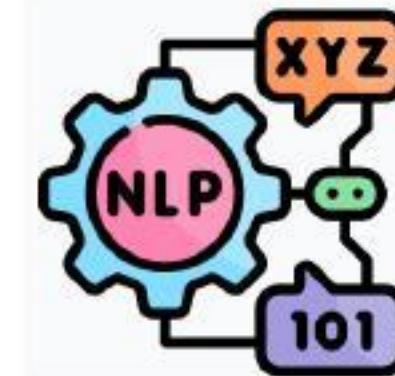
2. Continuous Bag of Words (CBOW)

A quick brown fox jumps over the lazy dog

King

Man

Woman

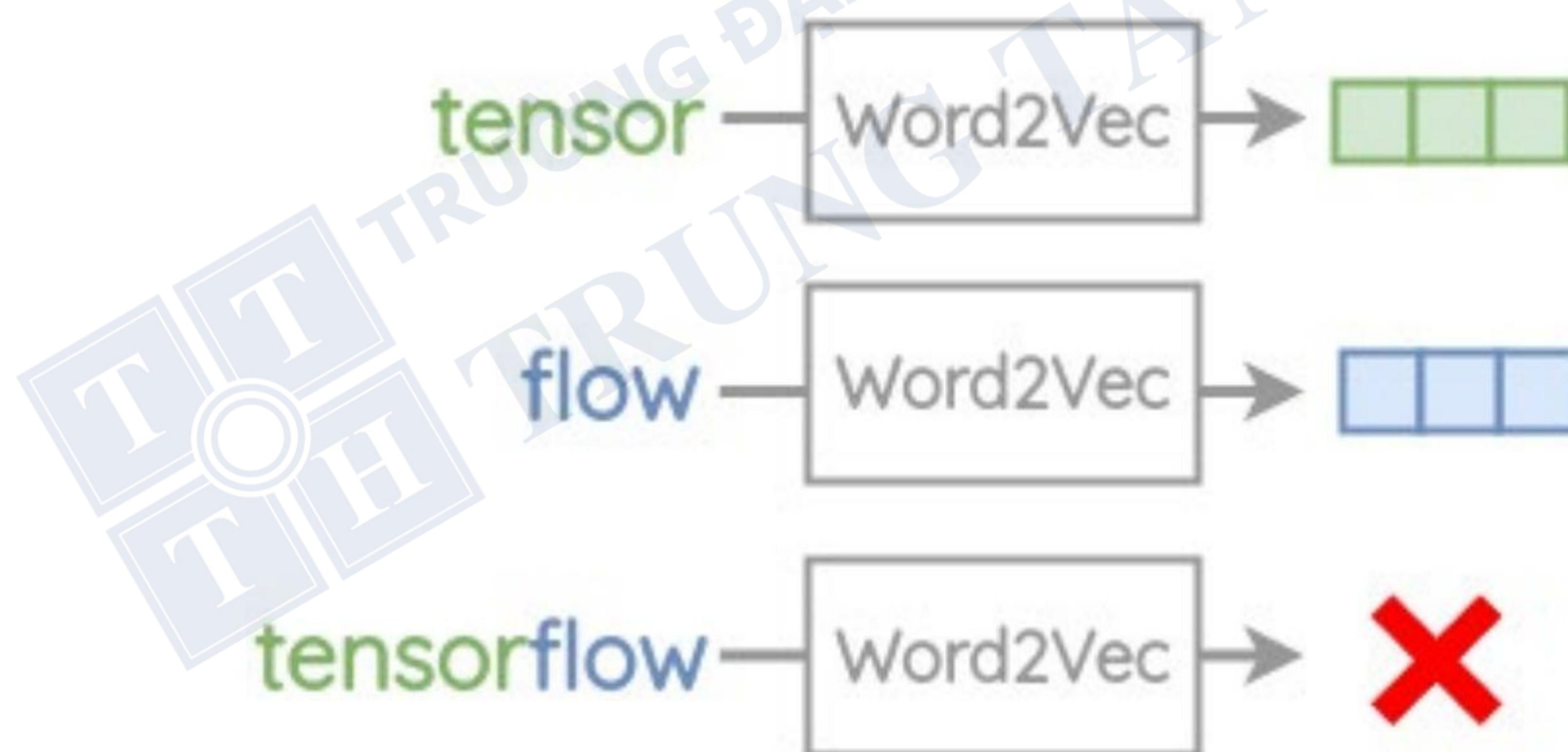


Hạn chế của Word2vec

Word2Vec biểu diễn từ trong không gian vector.



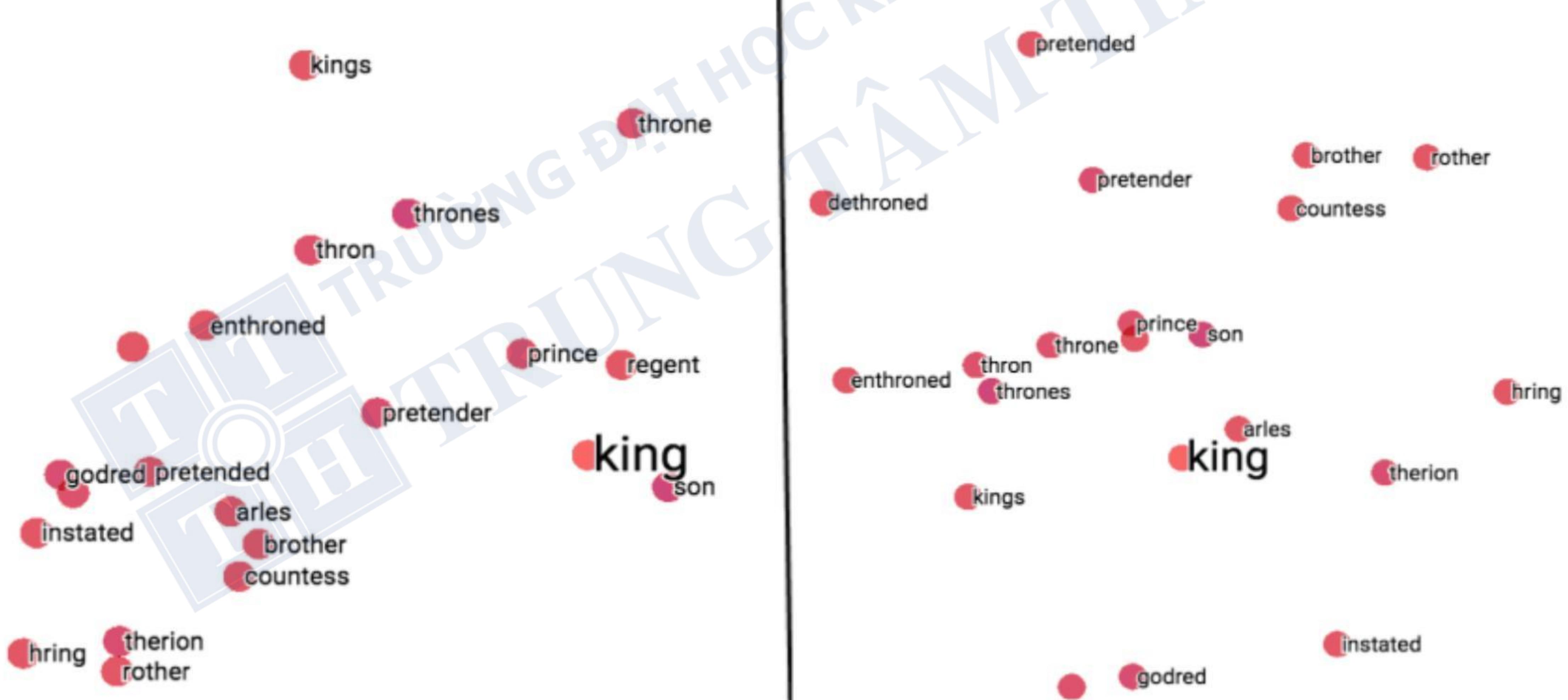
→ Không biểu diễn được từ ghép



fastText vs. Word2vec

fastText

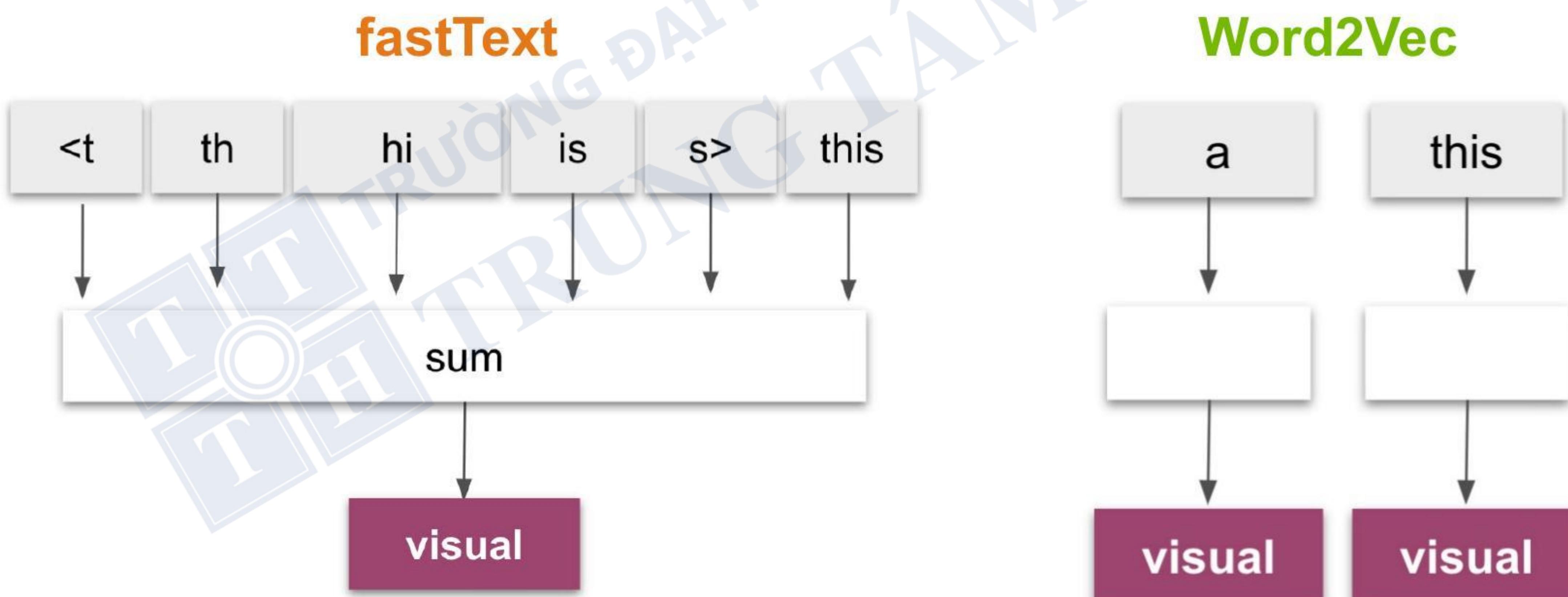
Library for efficient text classification and representation learning

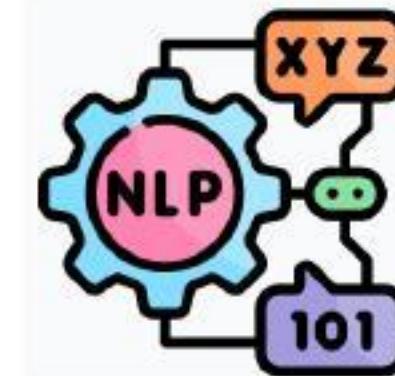


fastText vs. Word2vec

fastText là mô hình mở rộng của Word2Vec biểu diễn từ/ văn bản dưới dạng vector.

Sử dụng **bag of n-grams**, xem xét các n-gram (từ hoặc subword) của từ để tạo ra word vector.





fastText vs. Word2vec



Word	Length(n)	Character n-grams
eating	3	<ea, eat, ati, tin, ing, ng>
eating	4	<eat, eati, atin, ting, ing>
eating	5	<eati, eatin, ating, ting>
eating	6	<eatin, eating, ating>



VECTOR SPACE & DIMENSIONALITY REDUCTION

I. Ngôn ngữ trong NLP & Word Vector

II. Mô hình Word2vec

III. Mô hình GloVe

IV. Neural Network Classifier



Mô hình GloVe

Global Vectors for Word Representation là mô hình biểu diễn từ dưới dạng các vector số.

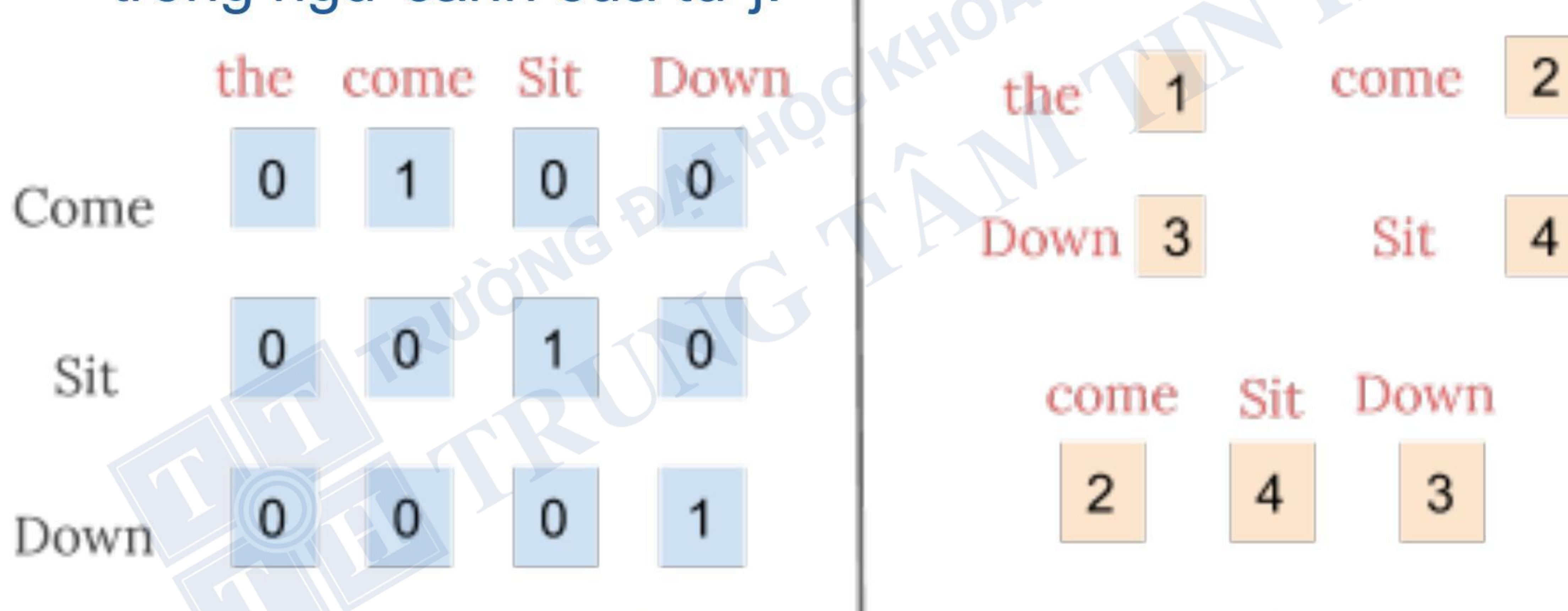
- Phát triển bởi Stanford University.

GloVe kết hợp thông tin từ cả ngũ cảnh và **thống kê** toàn bộ văn bản để tạo ra *word representative*.
→ Tính toán xác suất xuất hiện của từ i trong ngũ cảnh của từ j.

→ Hiểu và biểu diễn ngũ nghĩa của từ toàn diện hơn.

Mô hình GloVe

1. **Ma trận đếm từ** đếm số lần xuất hiện của từ i trong ngũ cảnh của từ j.
2. **Ma trận trọng số** tính toán xác suất xuất hiện của từ i trong ngũ cảnh của từ j.



→ Tạo word vector thể hiện ngũ nghĩa và mối quan hệ giữa các từ từ xác suất xuất hiện trong ngũ cảnh.



Mô hình GloVe

1. **Ma trận đếm từ** đếm số lần xuất hiện của từ i trong ngũ cảnh của từ j.
2. **Ma trận trọng số** tính toán xác suất xuất hiện của từ i trong ngũ cảnh của từ j.

$$p_{\text{co}}(w_k|w_i) = \frac{C(w_i, w_k)}{C(w_i)} \quad F(w_i, w_j, \tilde{w}_k) = \frac{p_{\text{co}}(\tilde{w}_k|w_i)}{p_{\text{co}}(\tilde{w}_k|w_j)}$$

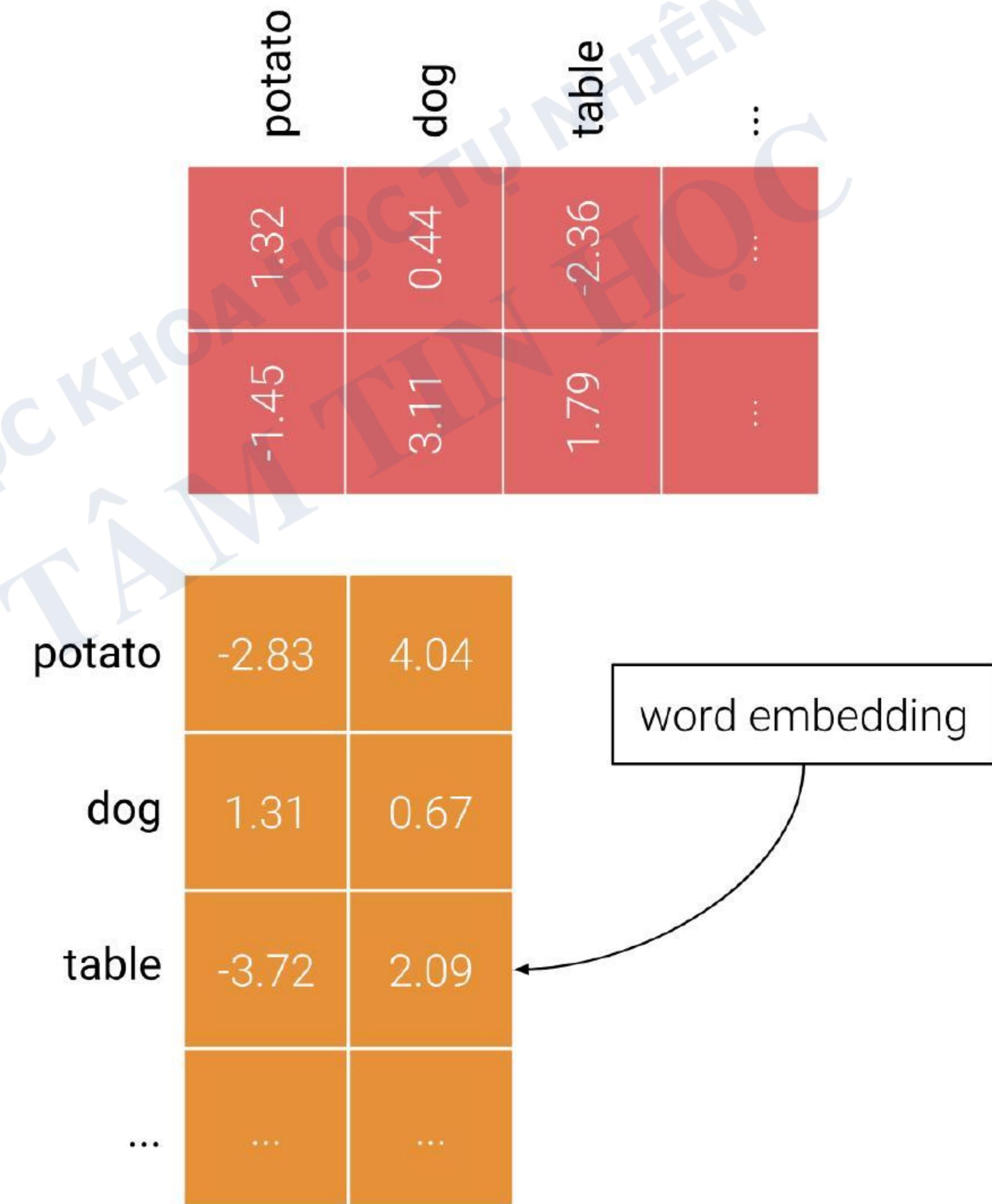
$$F((w_i - w_j)^\top \tilde{w}_k) = \frac{p_{\text{co}}(\tilde{w}_k|w_i)}{p_{\text{co}}(\tilde{w}_k|w_j)}$$

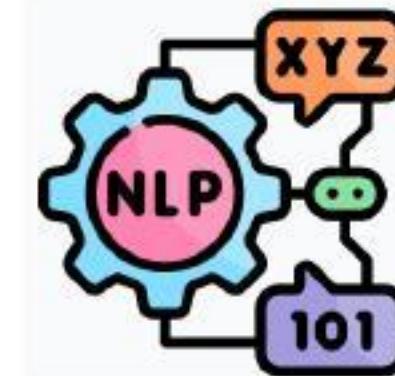
Mô hình GloVe



	potato	dog	table	...
potato	0	1.14	2.72	...
dog	1.14	0	1.69	...
table	2.72	1.69	0	...
...

logarithm of
co-occurrence matrix





VECTOR SPACE & DIMENSIONALITY REDUCTION

I. Ngôn ngữ trong NLP & Word Vector

II. Mô hình Word2vec

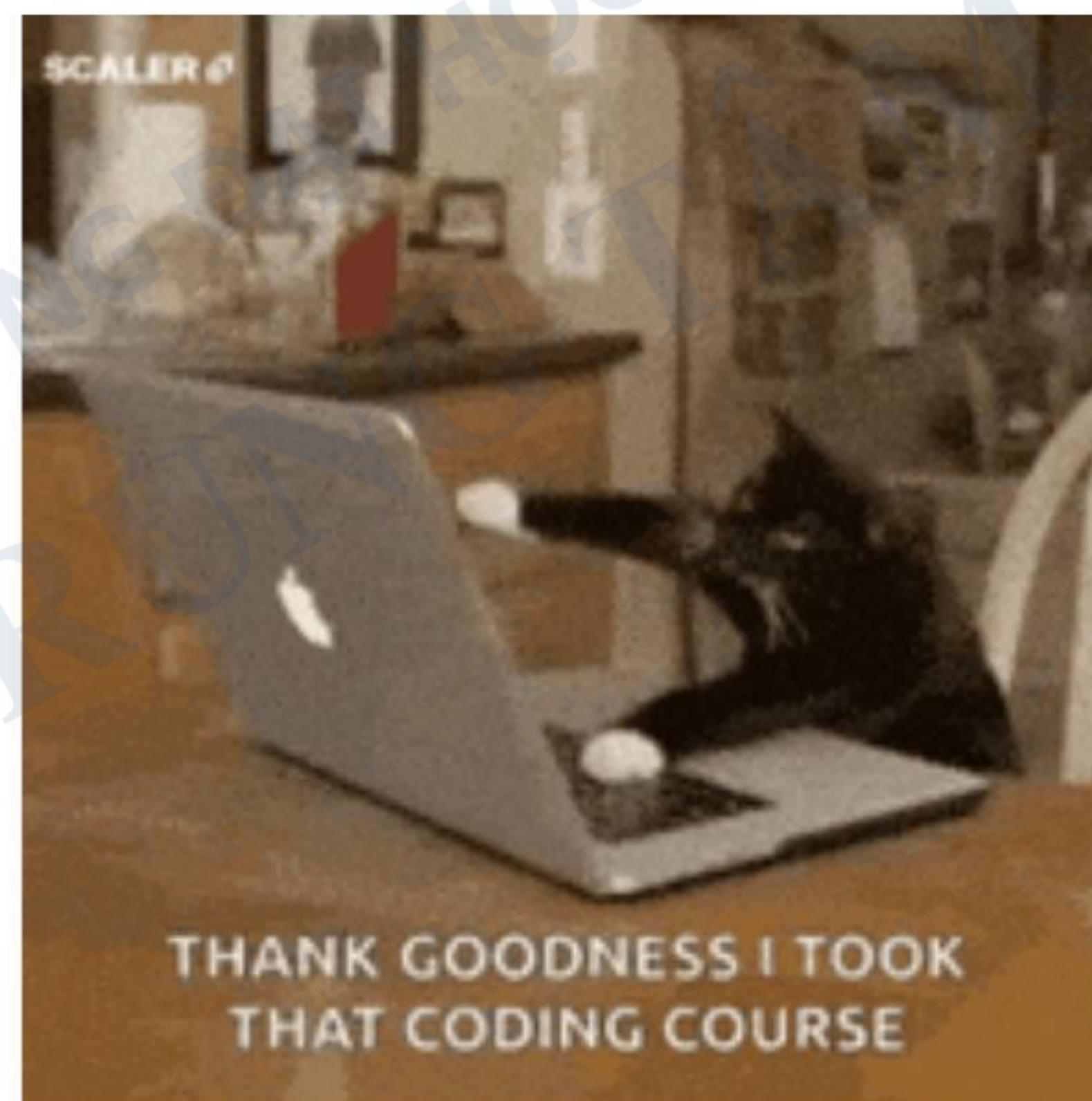
III. Mô hình GloVe

IV. Neural Network Classifier

Code Demo



THỰC HÀNH



Q&A

