ĐẠI HỌC QUỐC GIA TP. HCM
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

Chủ đề:

Giới thiệu về Word2Vec: mô hình ánh xạ từ trong văn bản về không gian vector (word embedding)

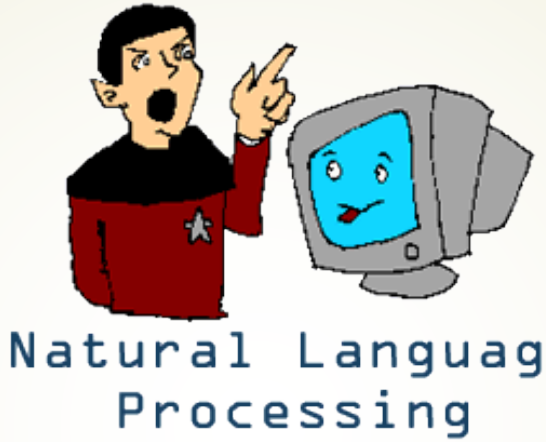
Trình bày: **Th.S. PHẠM THẾ ANH PHÚ**

(phamtheanhphu@gmail.com)

Các nội dung trình bày

1. Tổng quan về xử lý ngôn ngữ tự nhiên (NLP) và tầm quan trọng của Word Embedding (WE).
2. Mô hình Word2Vec.
3. Xây dựng mô hình Word2Vec với ngôn ngữ Python.
4. Tài liệu tham khảo.

UIT-HCM



1. Tổng quan về xử lý ngôn ngữ tự nhiên (NLP) và tầm quan trọng của Word Embedding (WE).

Xử lý ngôn ngữ tự nhiên (Natural Language Processing) (NLP)

UIT-HCM

- Xử lý ngôn ngữ tự nhiên là một nhánh quan trọng của lĩnh vực trí tuệ nhận tạo (AI).
- Tập trung vào giải quyết bài toán giao tiếp giữa người và máy tính (người-máy).
- Đòi hỏi sự kết hợp tri thức của nhiều ngành khoa học khác nhau: ngôn ngữ học, văn hóa, khoa học máy tính, v.v.

Tầm quan trọng của NLP trong các lĩnh vực khác nhau

UIT-HCM

- NLP là một lĩnh vực rất quan trọng vì nó được áp dụng để giải quyết rất nhiều bài toán khác nhau liên quan đến:
  - Xử lý dữ liệu văn bản (text mining).
  - Nhận diện & xử lý giọng nói (voice recognition & processing).
  - Truy hồi thông tin (information retrieval).
  - Phân tích cảm xúc (sentiment analysis)
  - …

Ứng dụng của NLP trong các lĩnh vực khác nhau

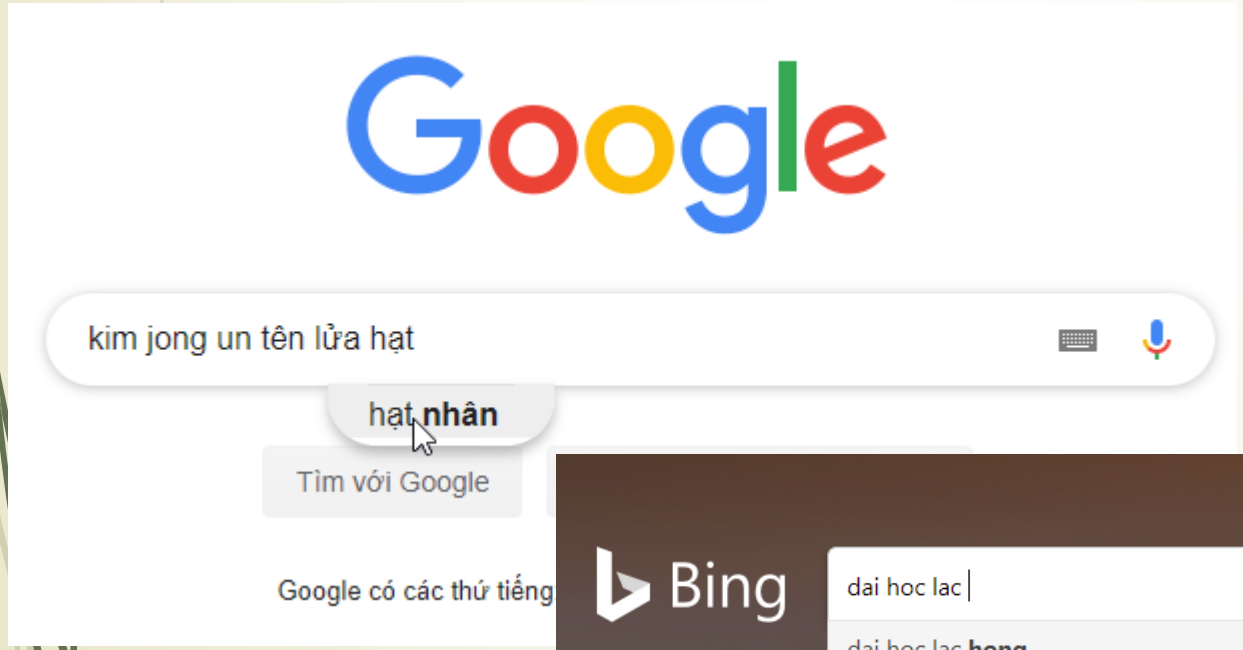
UIT-HCM

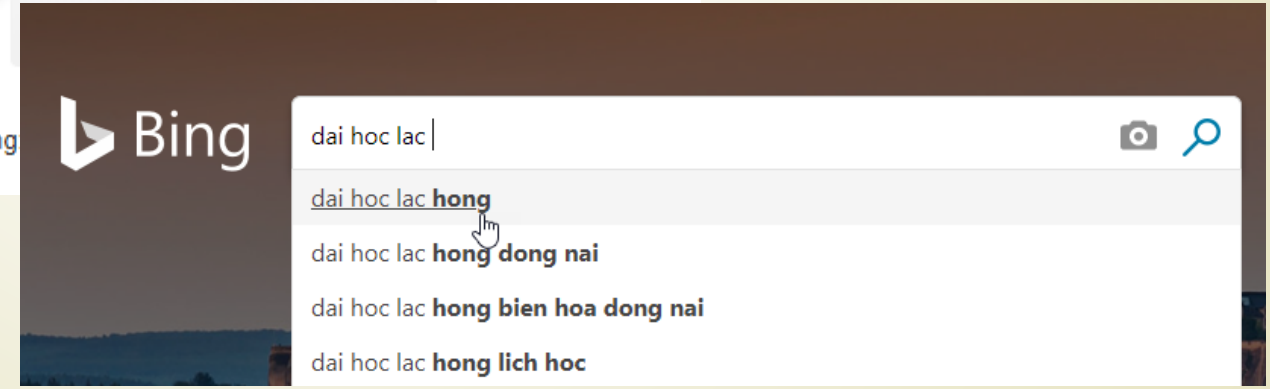
- Hỗ trợ cho việc giao tiếp giữa người và máy (Google Assistant, Siri, v.v.).

Ứng dụng của NLP trong các lĩnh vực khác nhau (tt)

UIT-HCM

- Truy hồi thông tin, mở rộng, gợi ý truy vấn cho người dùng, ví dụ: Google, Bing, v.v.





Các khó khăn hiện tại của NLP trong việc xây dựng ứng dụng

UIT-HCM

- Sự phức tạp và đa dạng trong cấu trúc của từng ngôn ngữ khác nhau (Xin chào, Hello, 你好 (Ni hảo), こんにちは (Kon'nichiwa), v.v.) → **không thể có mô hình tổng quát**.
- Sự phức tạp trong **ngữ cảnh (context)** và **cách thức dùng từ** của văn bản tự nhiên, ví dụ:
  - Cái **giá(1)** này **giá(2)** bao nhiêu tiền ? → giá(1): giá treo quần áo, giá (2): giá cả
  - Ăn **giá(3)** rất tốt cho sức khỏe ! → giá: một loại rau/thực vật.

→ Quá phức tạp để có thể **tổng hợp** được hết các **quy luật** của một loại ngôn ngữ.

Word embedding (WE) một cách nghĩ mới trong NLP

UIT-HCM

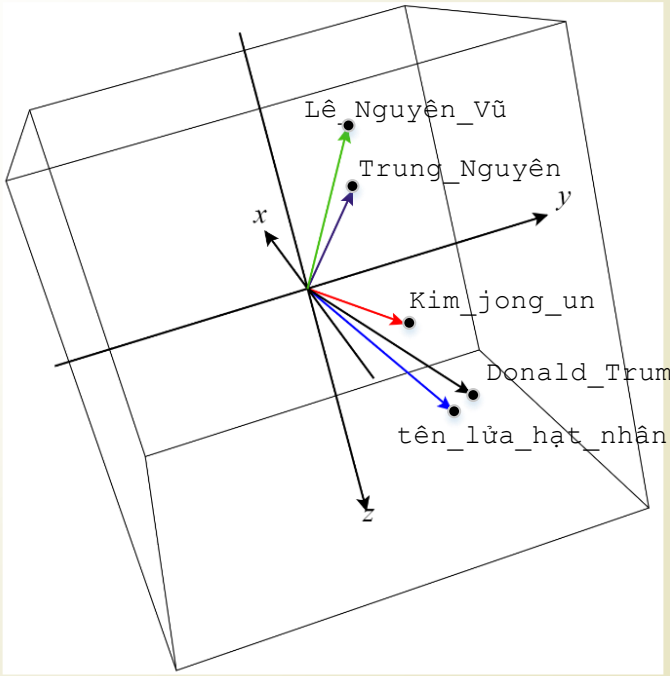
- **Độc lập về ngôn ngữ** (language independency) – có thể áp dụng cho nhiều loại ngôn ngữ khác nhau.
- Tận dụng tập dữ liệu văn bản rất lớn từ nhiều nguồn khác trên Internet để **tự học**.
- Học một cách **chủ động** từ tập dữ liệu được nạp, và hầu như **không cần** sự can thiệp của **tri thức chuyên gia**.
- Có khả năng **tự học tăng cường** với tập dữ liệu mới để nâng cao chất lượng của mô hình.

→**Làm thế nào ? (HOW ?)**

Mô hình ánh xạ/nhúng từ về không gian vector Word2Vec

UIT-HCM

- Đề xuất bởi **T. Mikolov (2013)** - MIKOLOV, Tomas, et al. "***Efficient estimation of word representations in vector space***". arXiv preprint arXiv:1301.3781, 2013.
- Là một mô hình **biểu diễn từ** (word representation learning).
- Mỗi từ sẽ được biểu diễn dựa trên tập **các từ ngữ cảnh** (context) xuất hiện xung quanh từ đó.
- Quá trình ánh xạ/nhúng (embedding) sẽ được thực hiện/học thông qua việc **huấn luyện mạng nơ-ron** (neural network).



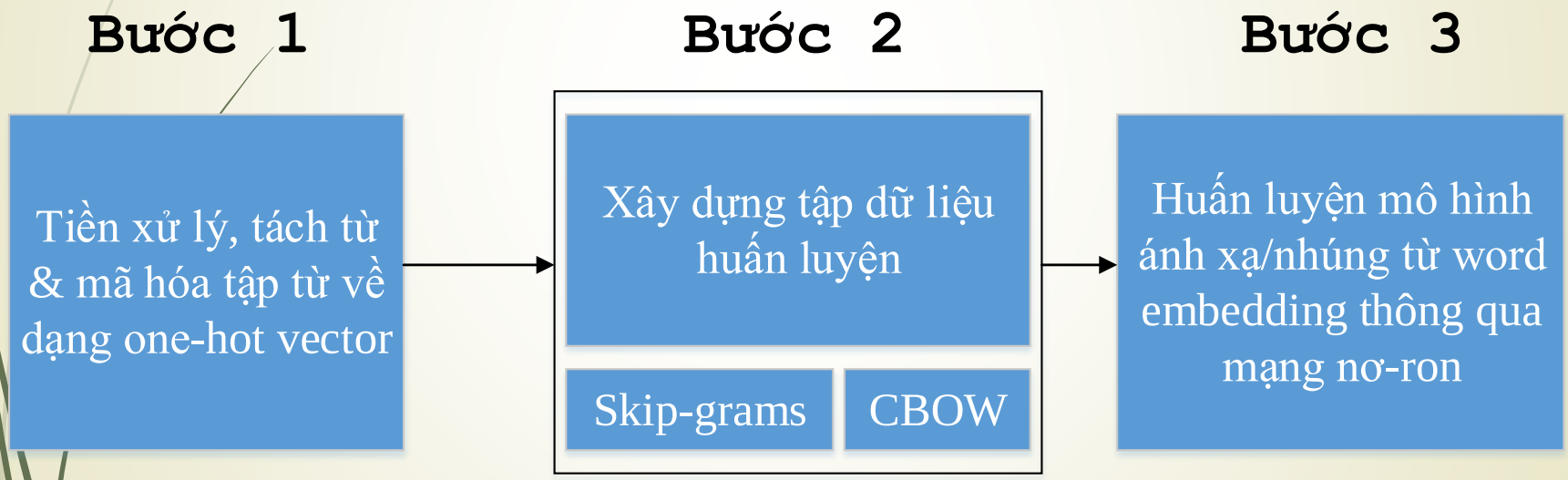
2. Giới thiệu về cơ chế & nguyên lý mô hình Word2Vec (*)

(*) MIKOLOV, Tomas, et al. "***Efficient estimation of word representations in vector space***". arXiv preprint arXiv:1301.3781, 2013.

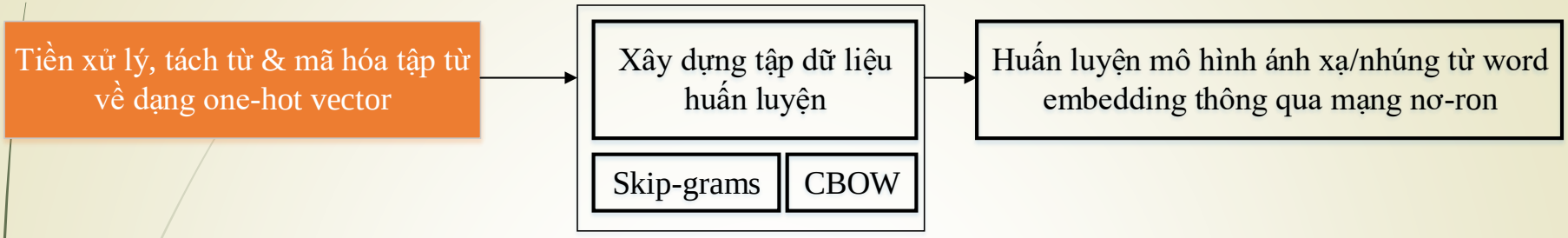
Quá trình huấn luyện mô hình Word2Vec

UIT-HCM

- Quá trình huấn luyện mô hình Word2Vec sẽ có ba giai đoạn chính, bao gồm:



Word2Vec - Bước 1: tiền xử lý, tách từ và mã hóa từ về dạng vector



- Từ một tập văn bản cho trước, qua các bước xử lý các từ sẽ được tách biệt, ví dụ cho một câu như sau:

"***Trăm năm trong cõi người ta***" → ["**Trăm**", "**năm**", "**trong**", "**cõi**", "**người**", "**ta**"].

- Mỗi từ sau đó sẽ được lập chỉ mục, sau đó chuyển đổi về dạng one-hot vector, ví dụ với cùng câu trên ta sẽ có 6 từ được chuyển về dạng one-hot vector như sau:
  - **Trăm**: [1, 0, 0, 0, 0, 0]
  - **Năm**: [0, 1, 0, 0, 0, 0]
  - …

Word2Vec - Bước 1: tiền xử lý, tách từ và mã hóa từ về dạng vector (tt)

UIT-HCM

Tiền xử lý, tách từ & mã hóa tập từ về dạng one-hot vector → Xây dựng tập dữ liệu huấn luyện (Skip-grams | CBOW) → Huấn luyện mô hình ánh xạ/nhúng từ word embedding thông qua mạng nơ-ron

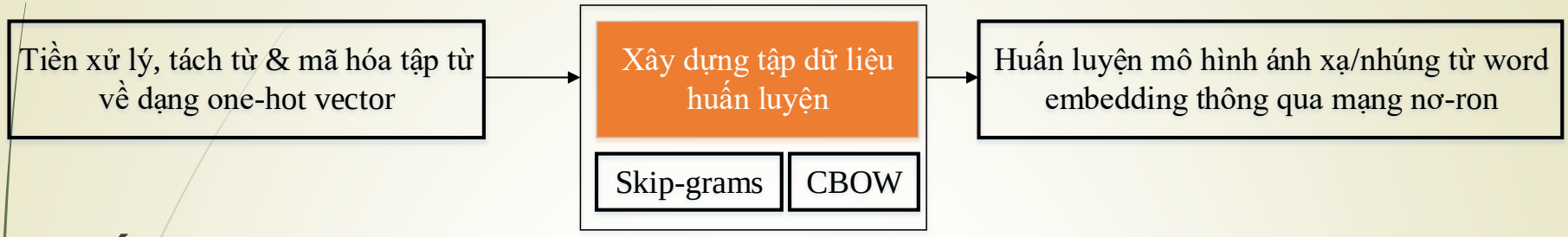
- Như vậy từ một tập văn bản cho trước, với số các từ đặc trưng/hay còn gọi là từ vựng (V) ta sẽ có một ma trận one-hot vector, với kích thước (|V|x|V|) được tạo ra như sau:

"Trăm năm trong cõi người ta" →

	trăm	năm	trong	cõi	người	ta
trăm	1	0	0	0	0	0
năm	0	1	0	0	0	0
trong	0	0	1	0	0	0
cõi	0	0	0	1	0	0
người	0	0	0	0	1	0
ta	0	0	0	0	0	1

|V| (rows) |V| (columns)

Word2Vec - Bước 2: tầm quan trọng của từ ngữ cảnh



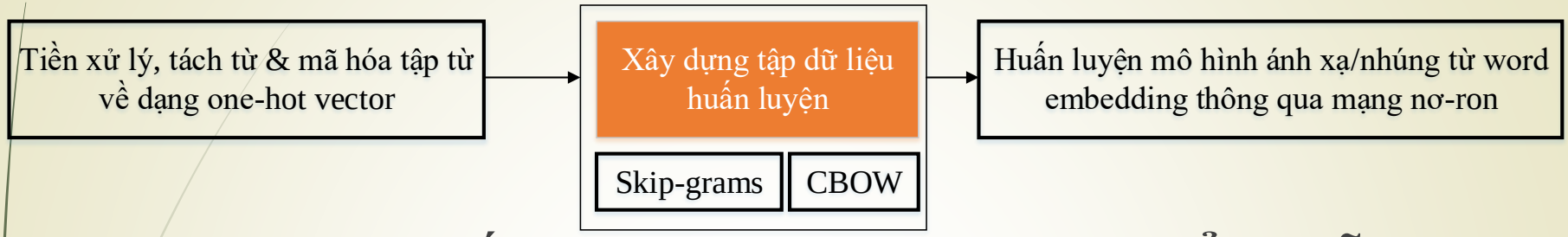
- Ý tưởng → "một từ được **định nghĩa** bởi tập các **từ ngữ cảnh** xung quanh nó", ví dụ: từ "giá" (giá cả) sẽ khác ngữ cảnh với giá (thực vật/ăn), v.v.

"Trăm năm trong **cõi** người ta"
tập từ ngữ cảnh | tập từ ngữ cảnh

"...cái giá này **giá** bao nhiêu..."
tập từ ngữ cảnh | tập từ ngữ cảnh

"...ăn **giá** rất tốt cho sức khỏe..."
tập từ ngữ cảnh | tập từ ngữ cảnh

Word2Vec - Bước 2: tầm quan trọng của từ ngữ cảnh (tt)



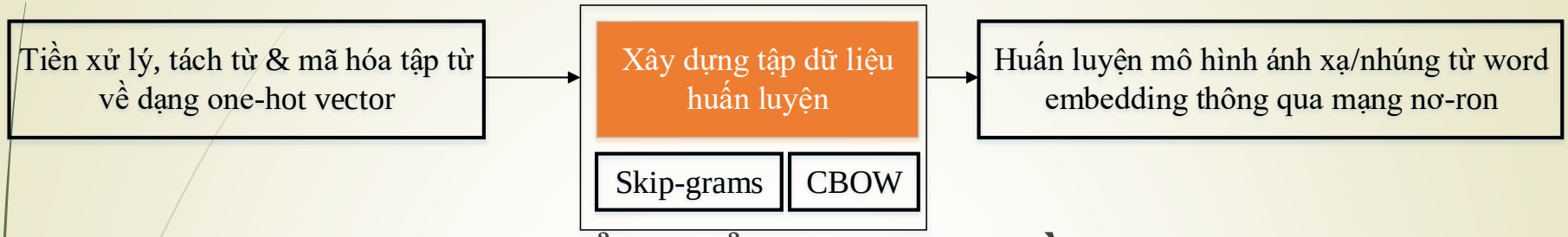
- Ngữ cảnh từ rất quan trọng trong việc **biểu diễn** sự **đặc trưng** của mỗi từ cũng như **phân biệt** giữa nó và các từ khác ra sao, vì hầu như mỗi từ khác nhau sẽ có các tập từ ngữ cảnh khác nhau. Ví dụ: "**con** sông…", "**con** kênh…", "**cái** hồ…", "**cái** ao…"

"...con **sông** này..." "...cái **hồ** này..."

"...con **kênh** này..." "...cái **ao** này..."

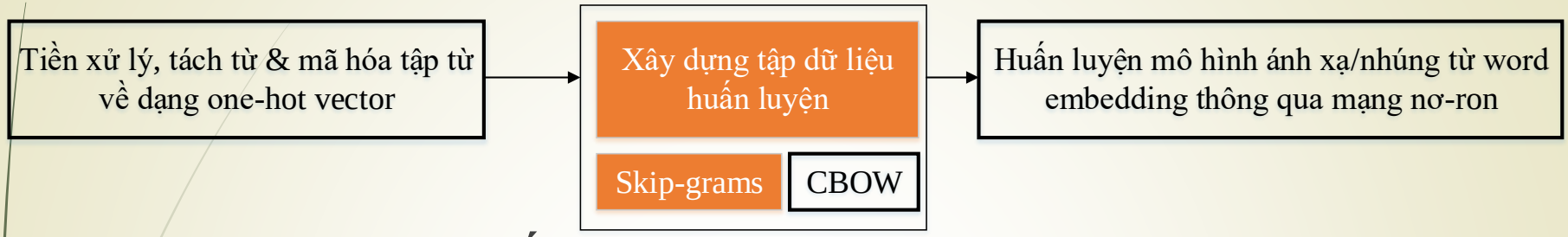
Word2Vec - Bước 2: sinh tập dữ liệu huấn luyện (tt)

UIT-HCM



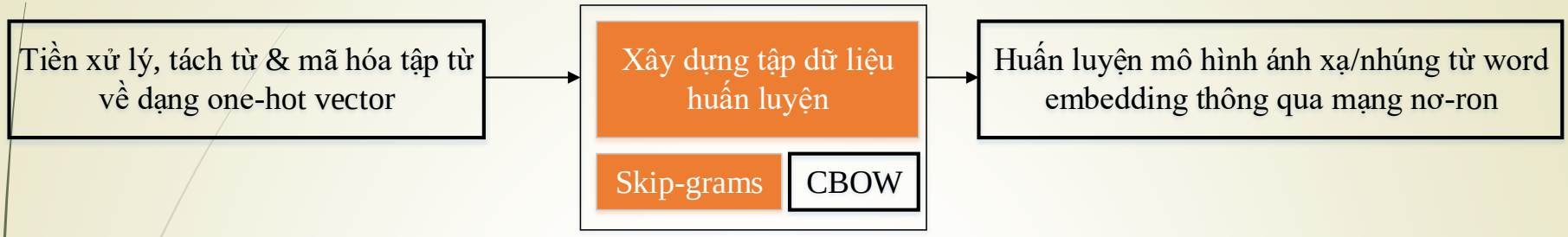
- Sau khi đã chuyển đổi các từ về dạng one-hot vector, bước tiếp theo là ta sẽ sinh tập dữ liệu huấn luyện cho từng từ dựa trên sự "**đồng hiện**" (**co-occurrence**) của chúng trong tập văn bản đã cho.
- Có **hai phương pháp** chính để xây dựng tập dữ liệu huấn luyện, đó là:
  - **Skip-gram**
  - **CBOW** (continuous bag-of-words).

Word2Vec - Bước 2: phương pháp Skip-grams



- Với một tham số quét (window size, ký hiệu **n**) cho trước, mô hình Skip-grams sẽ thực hiện việc sinh tập các từ ngữ cảnh cho 1 từ dựa trên sự đồng hiện của chúng trong văn bản như sau:
  - Từ vị trí từ được xét, gọi là **x** ta lấy về trước và về sau mỗi bên **n** phần tử, sau đó tạo thành 2n các cặp có dạng $< \mathbf{x}, \mathbf{x_n} >$.
  - Tổng hợp các cặp này sẽ tạo thành tập các từ ngữ cảnh (cũng ở dạng one-hot vectors) cho từ được xét **x**, tập dữ liệu huấn luyện sẽ có dạng: x:[$< \mathbf{x}, \mathbf{x_n} >$]

Word2Vec Bước 1: phương pháp Skip-grams



- Trở lại với ví dụ cho câu: "*Trăm năm trong cõi người ta*" – với tham số quét (windows size, n = 2) ta sẽ có tập các từ ngữ cảnh được sinh ra cho mỗi từ như sau:

"**Trăm năm** trong cõi người ta"

Năm:<**năm**,trăm>, <**năm**,trong>, <**năm**, cõi>

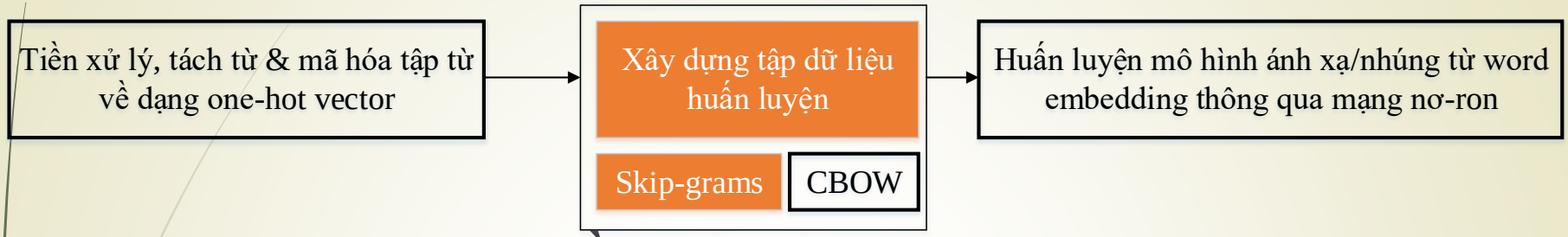
[0,1,0,0,0,0]:[[1,0,0,0,0,0],[0,0,1,0,0,0],[0,0,0,1,0,0]**]**

Trăm:<**trăm**,năm>, <**trăm**,trong>

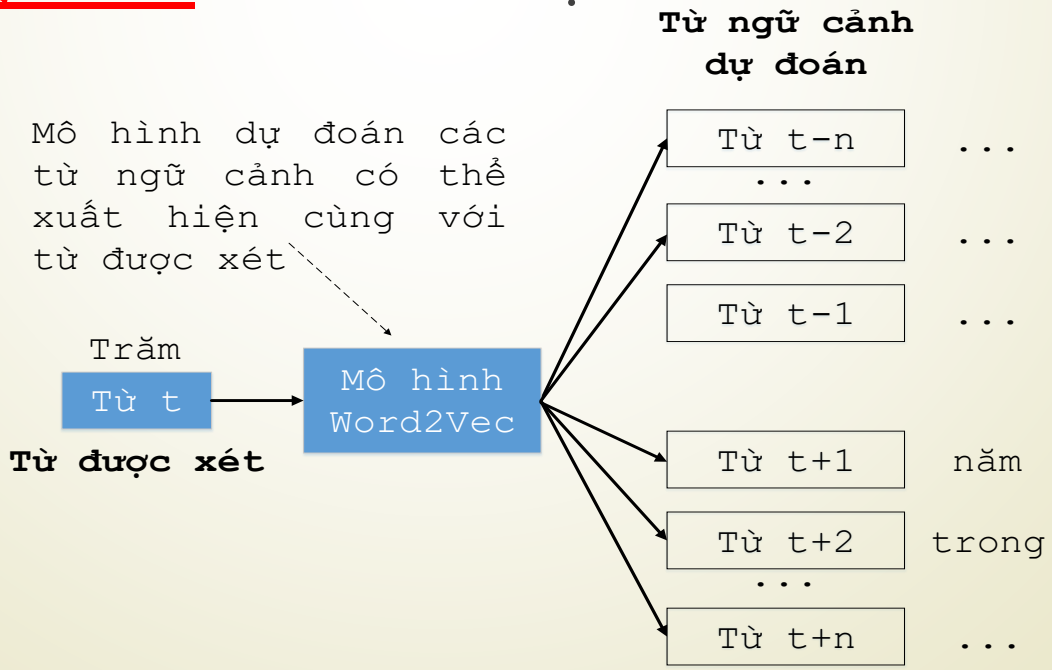
[1,0,0,0,0,0]:[[0,1,0,0,0,0],[0,0,1,0,0,0]**]**

Word2Vec - Bước 2: phương pháp Skip-grams (tt)

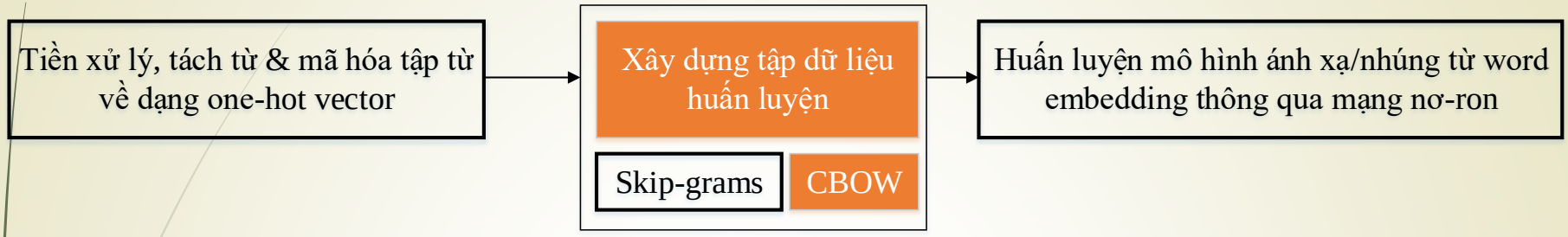
UIT-HCM



- Mục đích kỳ vọng **đầu ra (output)** của việc dùng phương pháp **Skip-grams** là → từ **1 từ biết trước** ta có thể **dự đoán** tập các **từ ngữ cảnh** có thể xuất hiện của nó.



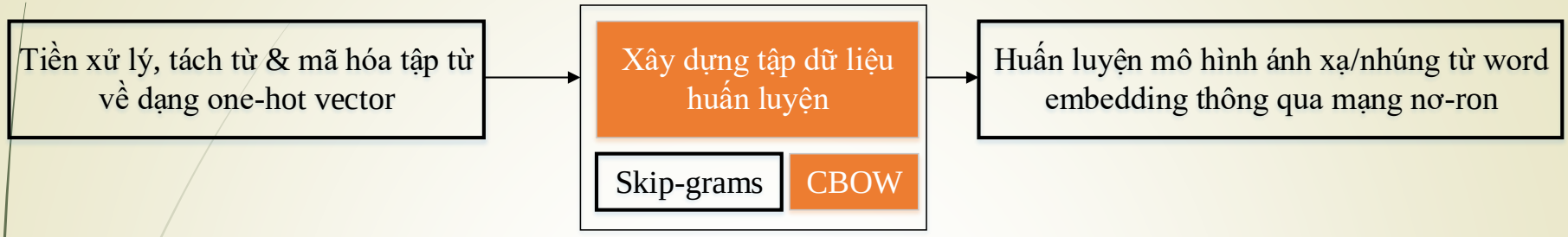
Word2Vec - Bước 2: phương pháp CBOW



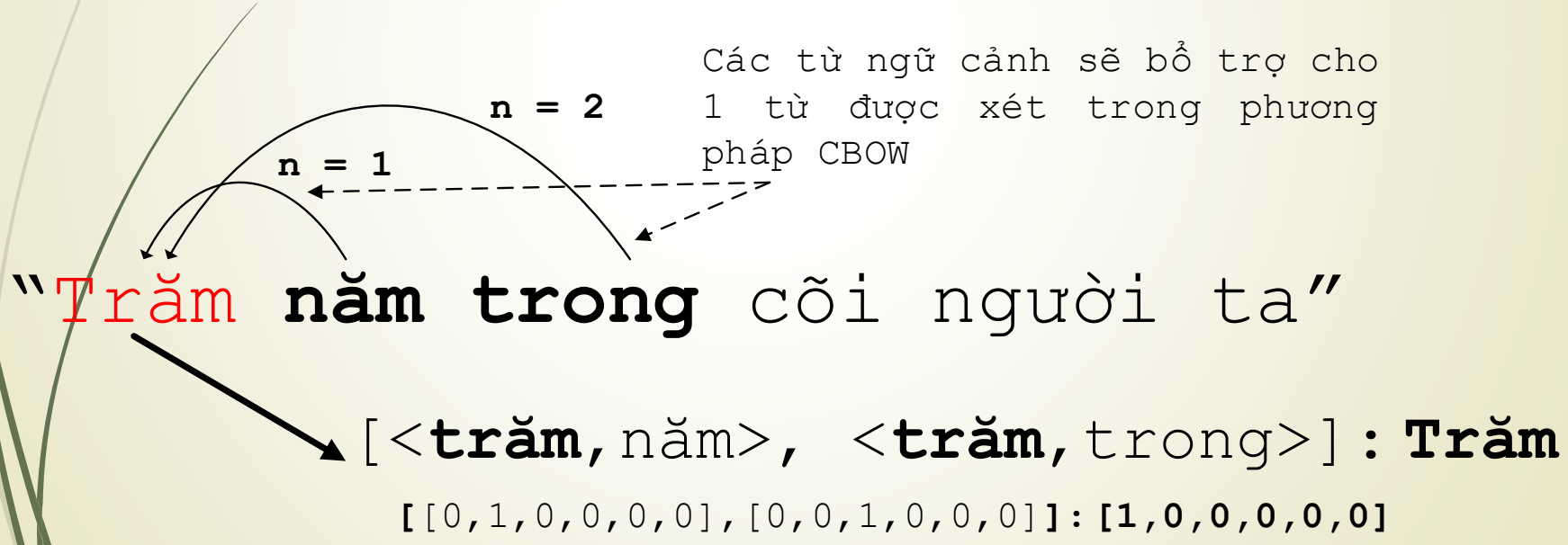
- **CBOW (continuous bag-of-words):** là một phương pháp sinh tập dữ liệu huấn luyện ngược với Skip-grams.
- Mục đích sử dụng CBOW trong quá trình huấn luyện Word2Vec là kỳ vọng đầu ra của mô hình ta là từ **1 tập từ ngữ cảnh biết trước** ta sẽ **dự đoán 1 từ** có khả năng **xuất hiện**.
- Phương pháp sinh tập dữ liệu huấn luyện cũng tương tự với Skip-grams, với tham số quét (window size, ký hiệu **n**) cho trước, tuy nhiên tập dữ liệu huấn luyện của CBOW sẽ có dạng: $[< \mathbf{x}, \mathbf{x_n} >]$:x, (ngược với Skip-grams x:$[< \mathbf{x}, \mathbf{x_n} >]$).

Word2Vec - Bước 2: phương pháp CBOW (tt)

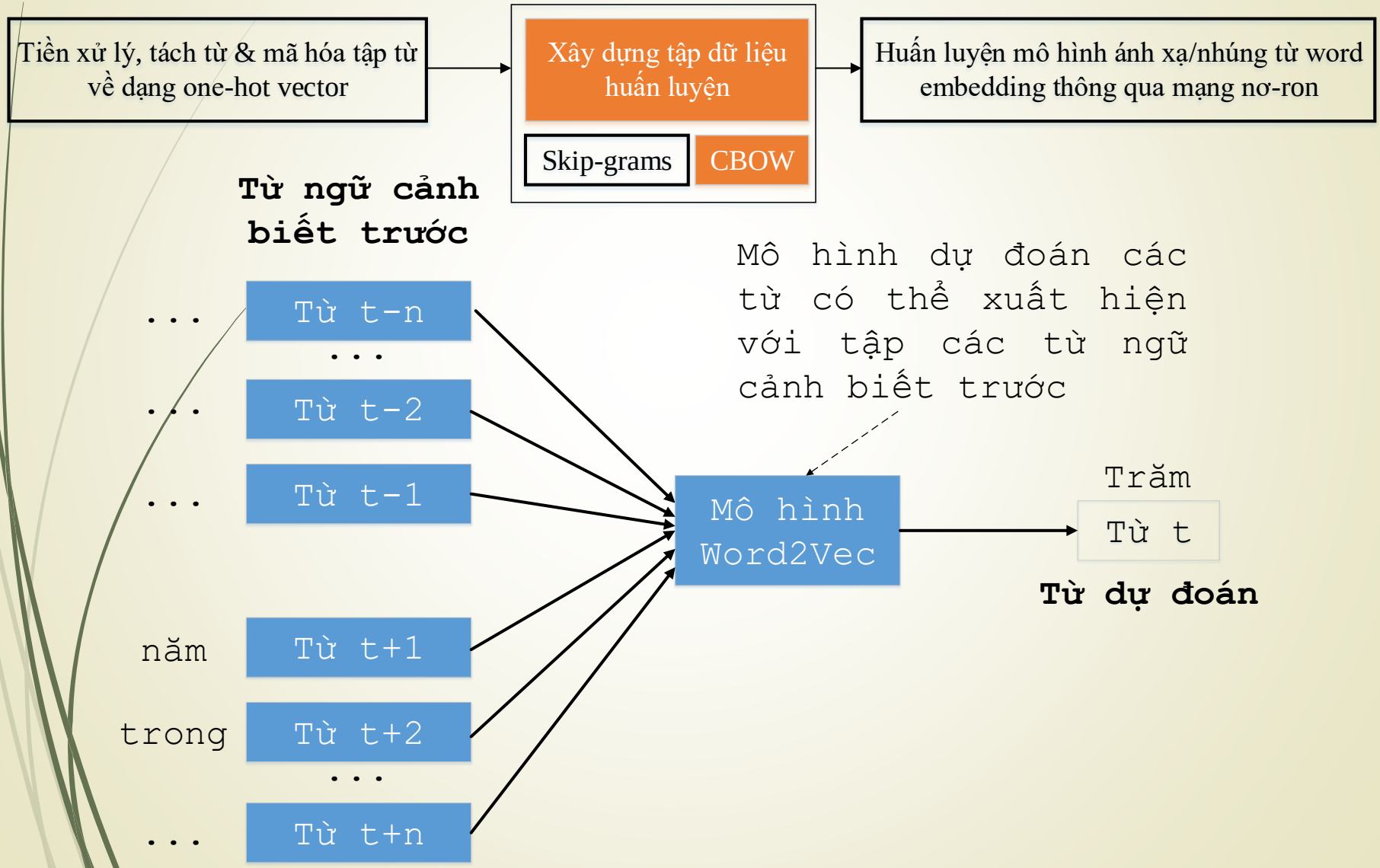
UIT-HCM



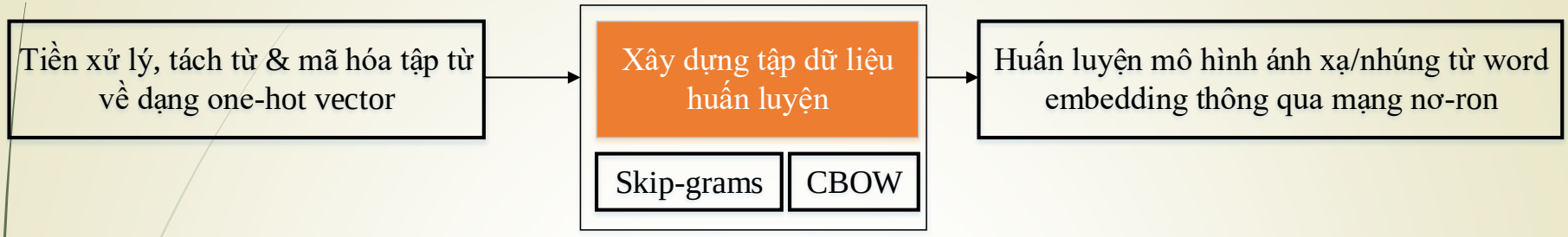
- Ví dụ với CBOW cho câu "*Trăm năm trong cõi người ta*":



Word2Vec - Bước 2: phương pháp CBOW (tt)

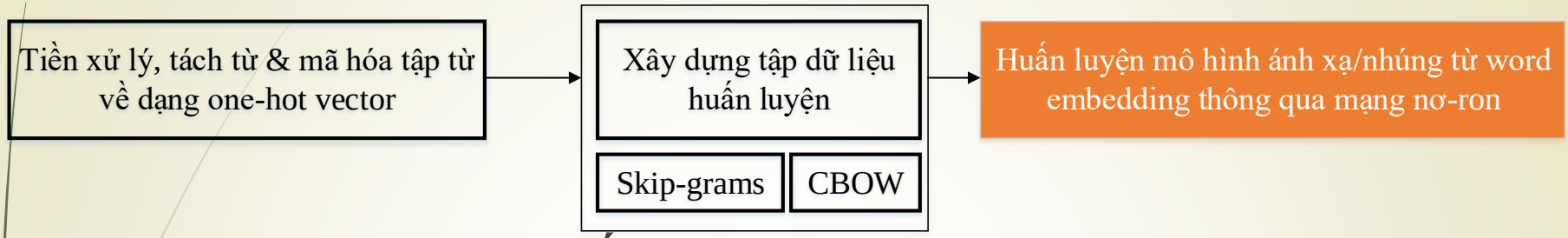


Word2Vec Bước 2: sinh tập dữ liệu huấn luyện (tt)



- Theo thực nghiệm của T. Mikolov và cộng sự (2013) thì cả hai phương pháp đều cho kết quả chính xác như nhau. Tuy nhiên tùy vào từng trường hợp mà ta sẽ áp dụng các phương pháp khác nhau, cụ thể là:
  - **Skip-grams**: cho 1 từ dự đoán các từ xung quanh có thể xuất hiện.
  - **CBOW**: với tập các từ ngữ cảnh biết trước, dự đoán từ có thể xuất hiện.

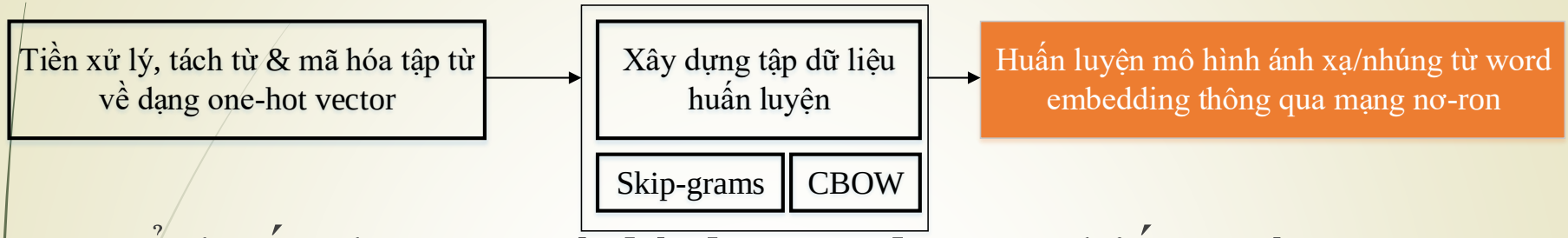
Word2Vec Bước 3: huấn luyện mô hình Word2Vec



- Với tập dữ liệu huấn luyện đã được xây dựng trong **bước 2**, ta sẽ **phân chia ra 2 phần**, như sau:
  - $\mathbf{W_t}$: là tập từ mục tiêu (target) được đưa vào ở đầu vào (input) của mạng nơ-ron. Với mỗi từ trong tập dữ liệu: $\overrightarrow{\mathbf{w_t}}, \overrightarrow{\mathbf{w_t}} \in W_t$ ở dạng one-hot vector.
  - $\mathbf{W_c}$: là tập từ ngữ cảnh (context) được dùng làm tập dữ liệu kỳ vọng cho kết quả đầu ra của mạng nơ-rơn. Mỗi tập ngữ cảnh, ký hiệu: $\mathbf{w_c} = \{\overrightarrow{w_{c_1}}, \overrightarrow{w_{c_2}}, \dots \overrightarrow{w_{c_n}}\}$, sẽ gồm nhiều từ: $\overrightarrow{\mathbf{w_{c_n}}}$ cũng ở dạng one-hot vector.

Word2Vec Bước 3: kiến trúc mạng nơ-ron của Word2Vec

UIT-HCM



- Để huấn luyện mô hình Word2Vec, kiến trúc mạng nơ-ron đơn giản với 1 tầng ẩn sẽ được áp dụng, trong đó ta sẽ có các cấu hình & tham số, như sau:
  - **Số chiều của vector đầu ra, ký hiệu d**: cũng chính là số lượng nơ-ron sẽ được dùng ở tầng ẩn của mạng nơ-ron.
  - **Hàm kích hoạt (activation function):**
    - Giữa đầu vào (input) và tầng ẩn (hidden), ký hiệu (**h**): sẽ là hàm tuyến tính có dạng: $\mathbf{f}(\overrightarrow{\mathbf{w_t}}) = \mathbf{h} = \mathbf{W1}.\ \overrightarrow{\mathbf{w_t}}$, trong đó **W1** là ma trận trọng số của **h**, có kích thước $|V| \times d$.
    - Giữa tầng ẩn đầu ra (output), ký hiệu (**u**): sẽ cũng là hàm tuyến tính có dạng: $\mathbf{f(h)} = \mathbf{u} = \mathbf{W2.h}$, trong đó **W2** là ma trận trọng số của **u**, có kích thước $d \times |V|$.

Word2Vec Bước 3: kiến trúc mạng nơ-ron của Word2Vec (tt)

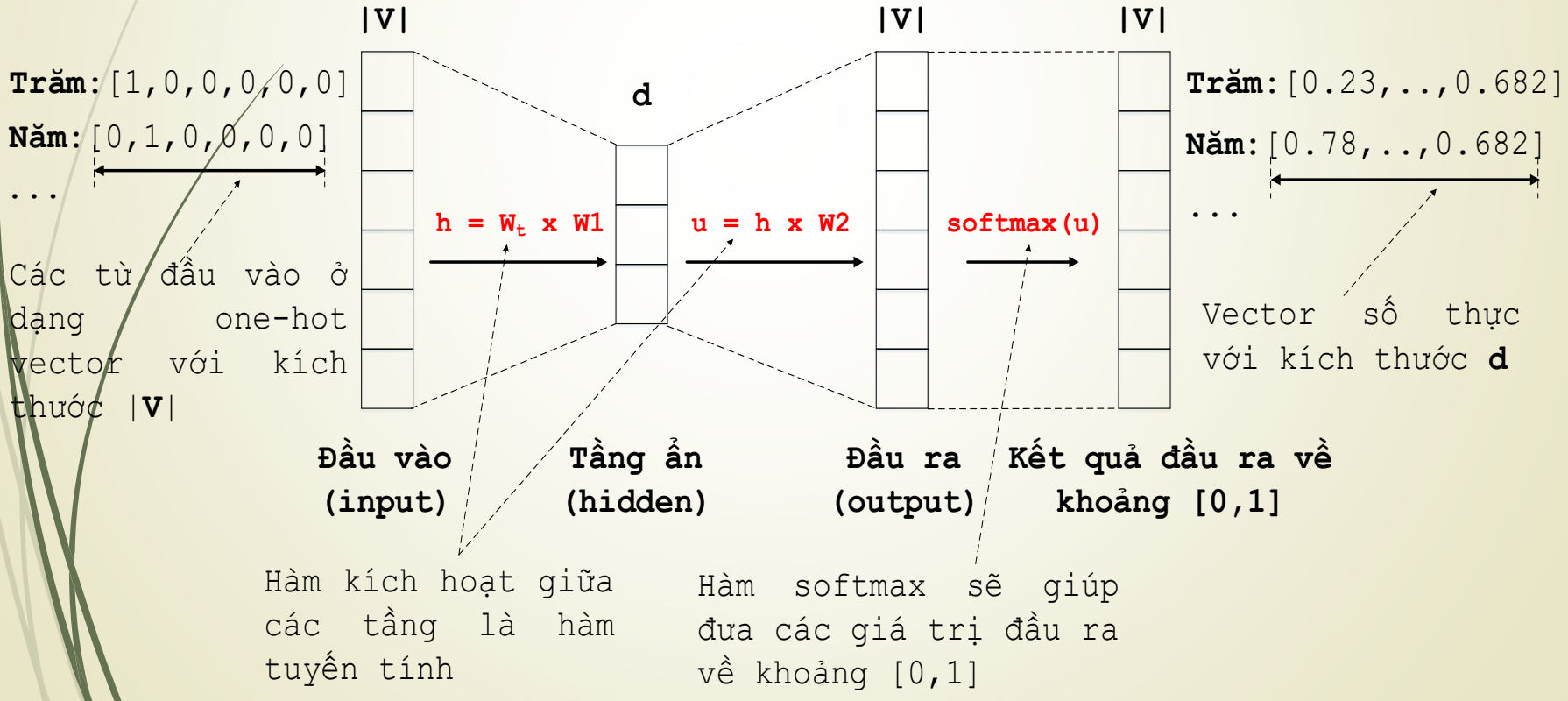
UIT-HCM

Tiền xử lý, tách từ & mã hóa tập từ về dạng one-hot vector

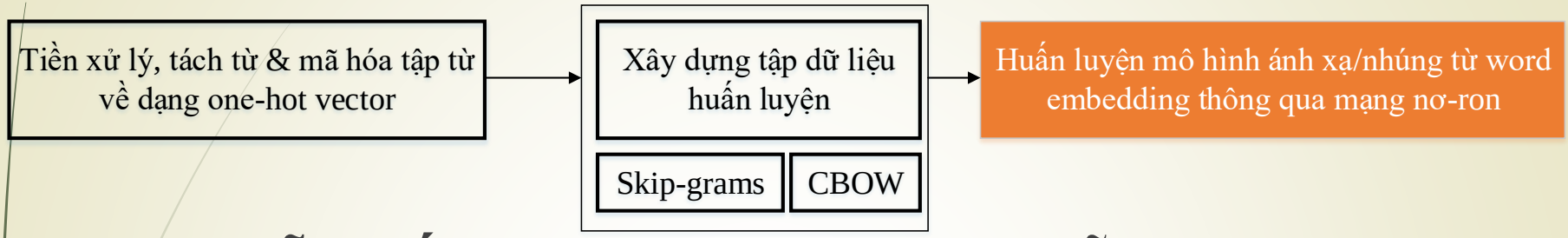
Xây dựng tập dữ liệu huấn luyện

Skip-grams | CBOW

Huấn luyện mô hình ánh xạ/nhúng từ word embedding thông qua mạng nơ-ron



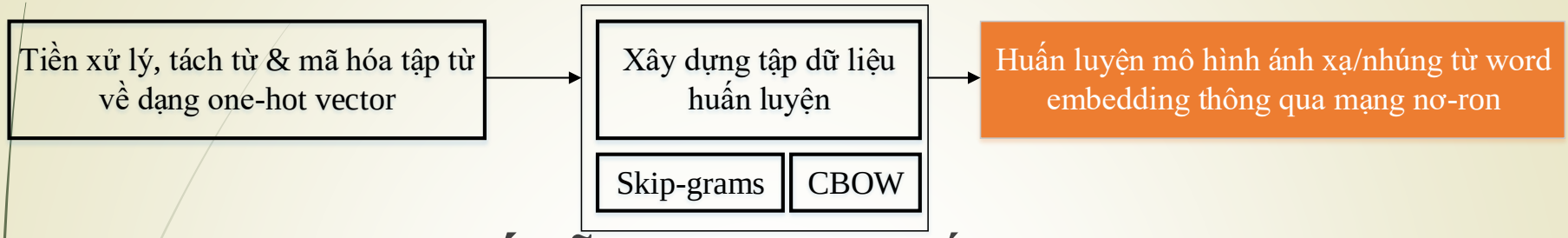
Word2Vec Bước 3: huấn luyện mạng nơ-ron với SGD



- **Suy diễn tiến (feed forward)**: từ mỗi từ mục tiêu: $\mathbf{w_t} \in \mathbf{W_t}$, ta sẽ tiến hành tính ra kết quả dự đoán $\hat{\mathbf{y}}$ thông qua việc suy diễn tiến qua hai hàm kích hoạt h và u như sau:

$$\hat{\mathbf{y}} = \mathbf{softmax}\left(\mathbf{u}\left(\mathbf{h}\left(\mathbf{w_t}\right)\right)\right)$$
$$= \mathbf{softmax}\left(\left[\mathbf{W2}^{\mathbf{T}} \cdot \left(\mathbf{W1}^{\mathbf{T}} \cdot \mathbf{w_t}\right)\right]\right)$$

Word2Vec Bước 3: huấn luyện mạng nơ-ron với SGD (tt)

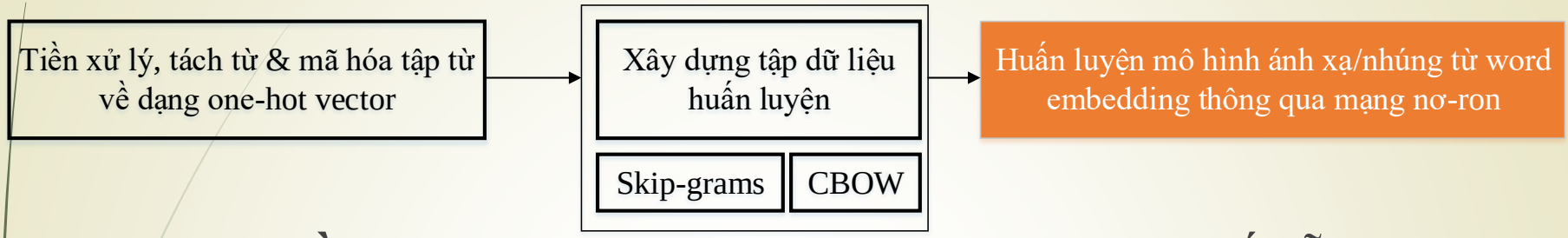


- **Xác định hệ số lỗi (loss)**: từ kết quả dự đoán $\hat{\mathbf{y}}$ và tập các từ ngữ cảnh kỳ vọng ở đầu ra $\mathbf{W_c}$, ta xác định hệ số lỗi với hàm lỗi (loss function) như sau:

$$\mathbf{loss} = \mathbf{y} - \hat{\mathbf{y}} = \sum_{\mathbf{w_c}}^{\mathbf{W_c}} (\mathbf{w_c} - \hat{\mathbf{y}})$$

- Với mỗi $\mathbf{w_c}, \mathbf{w_c} \in \mathbf{W_c}$ là từ ngữ cảnh kỳ vọng ở đầu ra cho mỗi từ mục tiêu được kỳ vọng sẽ xuất hiện tại đầu ra của mạng nơ-ron.

Word2Vec Bước 3: huấn luyện mạng nơ-ron với SGD (tt)



- **Lan truyền ngược (back propagration)**: từ hệ số lỗi (loss) đã được xác định, ta tiến hành cập nhật lại tập trọng số của **W1** và **W2** cho mỗi từ mục tiêu $\mathbf{w_t}$, theo hướng phương pháp Stochastic Gradient Descent (SGD), với $\boldsymbol{\eta}$ là tốc độ học (learning rate), như sau:

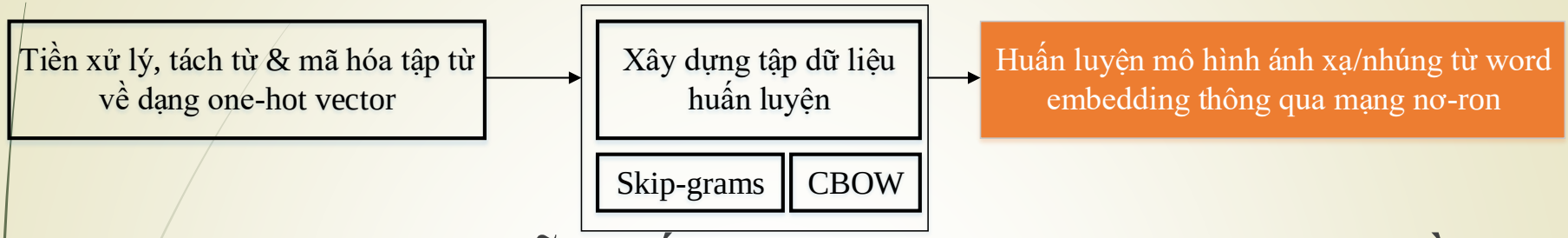
$$\mathbf{W2_{lệch}} = \mathbf{h} \otimes \mathbf{loss}$$

$$\mathbf{W2} = \mathbf{W2} - \boldsymbol{\eta}.\mathbf{W2_{lệch}}$$

$$\mathbf{W1_{lệch}} = \mathbf{w_t} \otimes (\mathbf{W2} \cdot \mathbf{loss}^{\mathbf{T}})$$

$$\mathbf{W1} = \mathbf{W1} - \boldsymbol{\eta}.\mathbf{W1_{lệch}}$$

Word2Vec Bước 3: huấn luyện mạng nơ-ron với SGD (tt)



- Quá trình suy diễn tiến (feed forward) và lan truyền ngược (back propagation) sẽ được lặp đi lặp lại liên tục, cho đến khi hệ số lỗi (loss) không còn giảm nữa. Thì lúc này mô hình đã được ngưỡng bão hòa (converged).
- Kết thúc quá trình huấn luyện, tập trọng số **W1**, kích thước $|V| \times d$, sau cùng sẽ chính là tập các vector số thực đại diện cho từng từ mục tiêu.

3. Xây dựng mô hình Word2Vec với ngôn ngữ Python

Xây dựng mô hình Word2Vec với ngôn ngữ Python

UIT-HCM

Trong phần này sẽ hướng dẫn cách xây dựng lại từ đầu mô hình Word2Vec với ngôn ngữ lập trình Python và thư viện NumPy.

Mục tiêu:

- Làm quen với việc xử lý các phép toán trên vector, ma trận, v.v. với thư viện NumPy.
- Hiểu được cơ chế và tự cài đặt lại Word2Vec.

Công cụ cần thiết:

- **PyCharm** IDE (Community edition) (https://www.jetbrains.com/pycharm/download/)
- **Python 2.x, 3.x**
- Thư viện **NumPy** phiên bản >= 1.16

Xây dựng mô hình Word2Vec với ngôn ngữ Python (tt)

UIT-HCM

Source code demo: download tại link sau (https://github.com/phamtheanhphu/word2vec_tutorial_phupham)

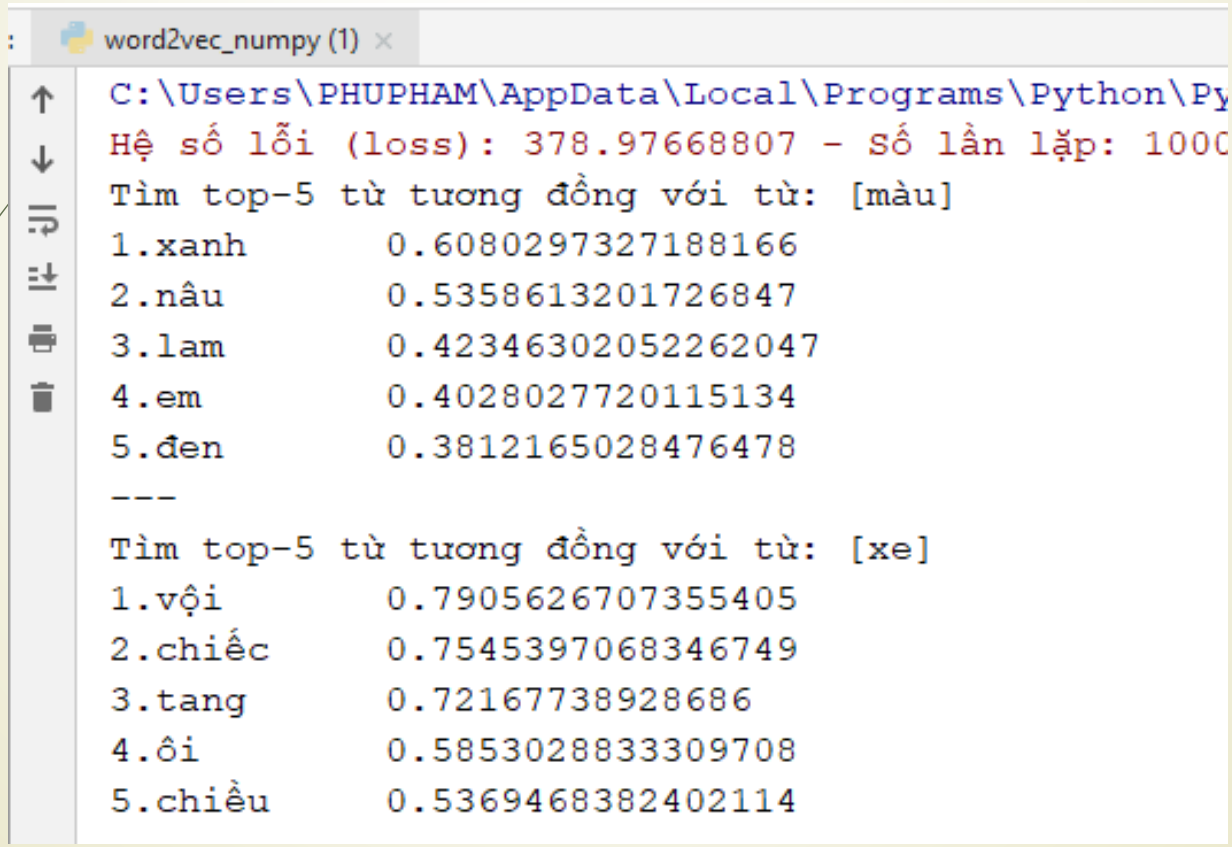
Ta có một tập dữ liệu văn bản với nội dung bài hát "**Sắc màu**" của nhạc sĩ Trần Tiến, như sau:

```
sentences = [
    'Một màu xanh xanh chấm thêm vàng vàng',
    'Một màu xanh chấm thêm vàng cánh đồng hoang vu',
    'Một màu nâu nâu một màu tím tím',
    'Màu nâu tím mắt em tôi ôi đẹp dịu dàng',
    'Một màu xanh lam chấm thêm màu chàm',
    'Thời chinh chiến đã xa rồi sắc màu tôi',
    'Một màu đen đen một màu trắng trắng',
    'Chiều hoang vắng chiếc xe tang đi vội vàng'
]
```

Xây dựng mô hình Word2Vec với ngôn ngữ Python (tt)

UIT-HCM

Thử nghiệm tìm kiếm các từ ngữ cảnh tương đồng của hai từ "**màu**" và "**xe**", ta có kết quả như sau:



4. Tài liệu tham khảo

Tài liệu tham khảo

[1] MIKOLOV, Tomas, et al. "***Efficient estimation of word representations in vector space***". arXiv preprint arXiv:1301.3781, 2013.

[2] GOLDBERG, Yoav; LEVY, Omer. "***word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method***". *arXiv preprint arXiv:1402.3722*, 2014.

[3] Video Word Vector Representations: word2vec (Christopher D Manning) https://www.youtube.com/watch?v=ERibwqs9p38

Xin cám ơn quý thầy/ cô và các bạn đã lắng nghe !