Word2Vec report

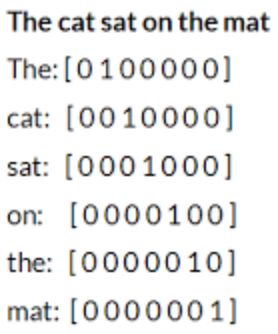
Status	Not Started
Project	Deep Learning
Tags	

Embedding Methods

Báo cáo các phương pháp One-hot Encoding, Windows based cooccurence matrix, SVD for dimension reduction, model CBOW, skip-gram.

One-hot encoding

Phương pháp này thực hiện đơn giản bằng cách đếm số lần xuất hiện của một từ của một câu trên toàn bộ corpus dữ liệu được sử dụng.



Tuy nhiên phương pháp này lại kém hiệu quả trong việc thể hiện mối quan hệ giữa các từ, hoặc được ngữ cảnh của câu và độ phức tạp tăng theo số lượng từ.

Việc đếm số lần xuất hiện khiến khi lượng từ trong corpus tăng lên thì xuất hiện rất nhiều khoảng trống thể hiện bằng số “0” gây dư thừa việc tính toán và bộ nhớ.

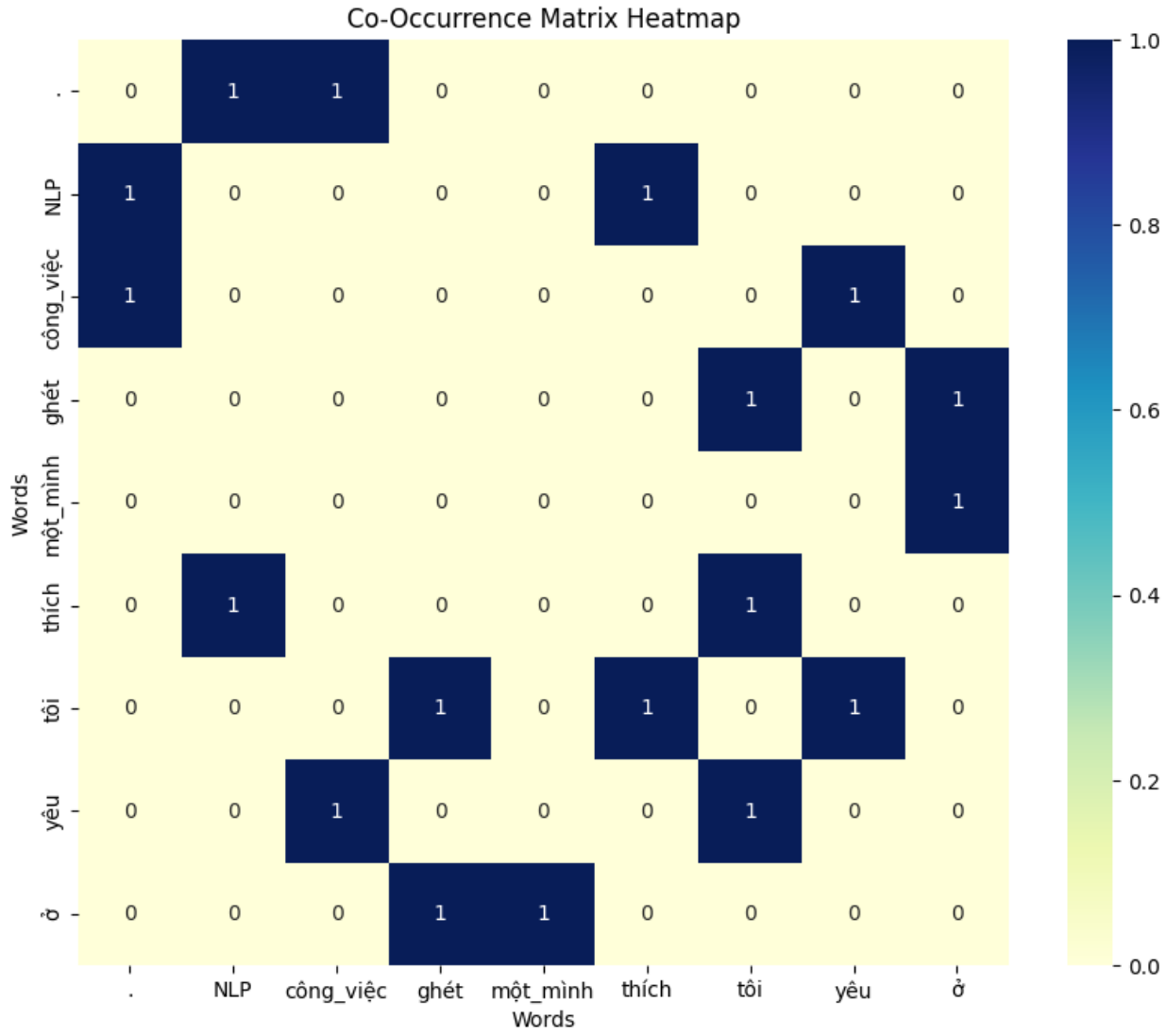
```
Câu gốc:  danh_sách tác_phẩm doraemon
Minh Họa One Hot Encoding:  [0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
 0 0 0 0 0 0 0 0 0]
```

Windows based cooccurence matrix

Phương pháp này đơn giản là việc lấy ngữ cảnh từ n bằng k từ xung quanh nó. Ví dụ như câu “Hôm nay, tôi đi học tại trường đại học quốc gia hà nội” với $windowLength = 2$ thì từ “học” sẽ tăng số lượng xuất hiện của 2 từ (”tôi”, “đi”, “tại”, “trường”) xung quanh (bên trái và bên phải) nó lên 1.

Một ví dụ minh họa cho Corpus:

- tôi yêu công_việc .
- tôi thích NLP .
- tôi ghét ở một_mình



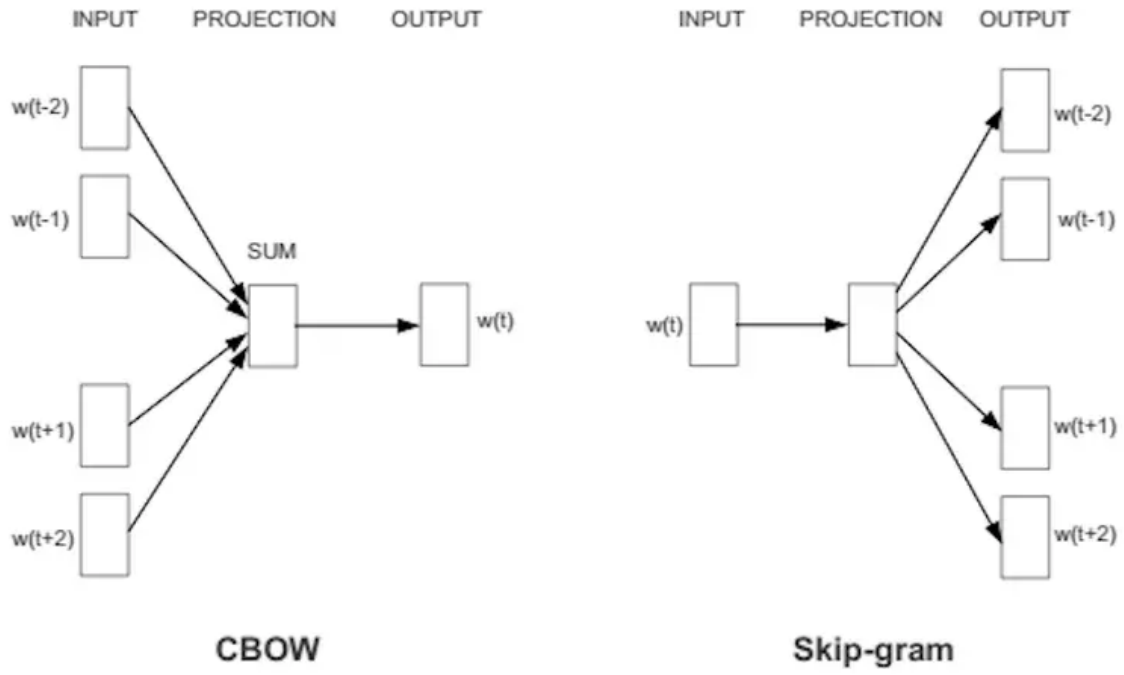
Tuy nhiên vấn đề ở đây là kích thước của bộ từ vựng nếu số lượng từ trong corpus tăng lên thì số chiều của ma trận cần sẽ rất lớn.

Singular Value Decomposition (SVD).

Giảm chiều dữ liệu sử dụng SVD. Chi tiết hơn tại: Singular value decomposition

Word2Vec

Hiện nay Word2Vec là phương pháp phổ biến và mạnh mẽ trong việc embedding với 2 mô hình Cbow và Skip-gram



Mô hình Skip-Gram

Mô hình skip-gam giả định rằng một từ có thể được sử dụng để sinh ra k từ xung quanh nó trong một chuỗi văn bản.

Ví dụ : "Hôm nay tôi đi học", ta sẽ sử dụng từ "tôi" là từ trung tâm và đặt $k = 2$ khi đó ta có mô hình skip-gram quan tâm đến xác suất có điều kiện sinh ra các từ ngữ cảnh ("Hôm", "nay", "đi" và "học") nằm trong khoảng cách không quá 2 từ là :

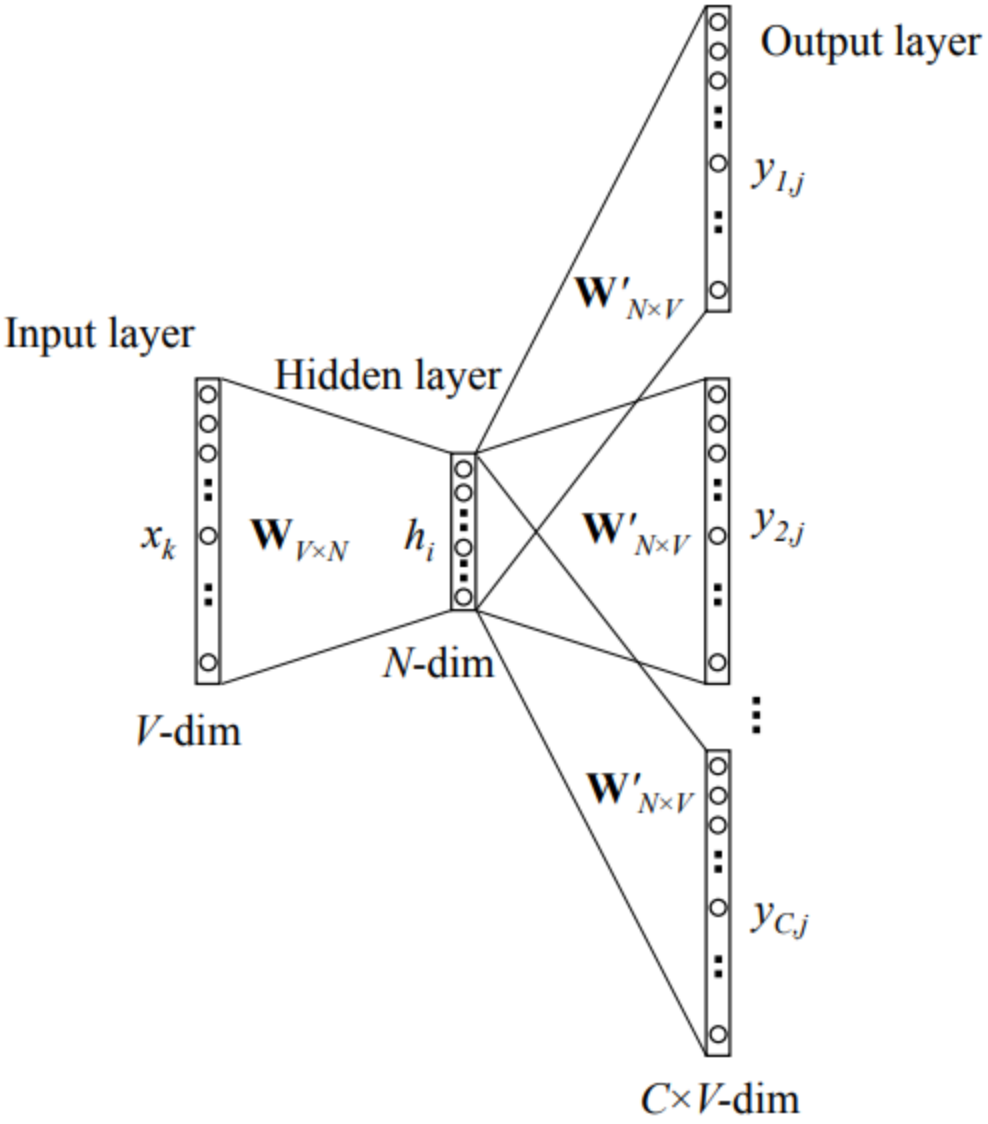
$$P(\text{"hôm"}, \text{"nay"}, \text{"đi"}, \text{"học"} \mid \text{"tôi"})$$

Ta giả định với từ trung tâm đã cho, các từ ngữ cảnh được generate một cách độc lập, khi đó công thức sẽ được viết dưới dạng

$$P(\text{"hôm"} \mid \text{"tôi"}) \cdot P(\text{"nay"} \mid \text{"tôi"}) \cdot P(\text{"đi"} \mid \text{"tôi"}) \cdot P(\text{"học"} \mid \text{"tôi"}).$$

Mô hình tổng quan :

$$\prod_{t=1}^{T} \prod_{-m \leq j \leq m, j \neq 0} P(w^{(t+j)} \mid w^{(t)}),$$



Sau đó chúng ta sử dụng phép $logarit$ để tạo một loss function và mục tiêu của mô hình là tối ưu function này

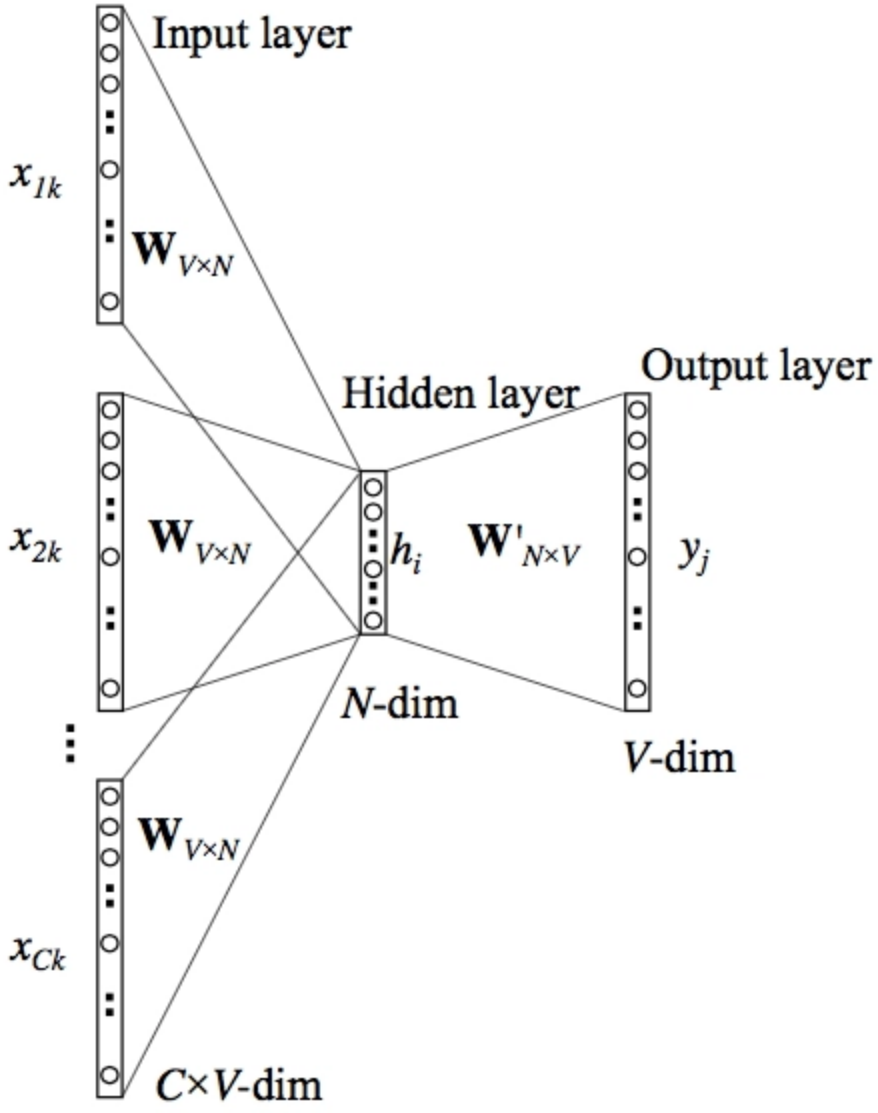
$$-\sum_{t=1}^{T} logP(w^{(t)}) \mid w^{(t-m)}, \ldots, w^{(t-1)}, w^{(t+1)}, \ldots, w^{(t+m)}).$$

Sau đó update mô hình thông qua gradient descend, mô hình có thể học được biểu diễn của từ trên n chiều với những từ có nghĩa giống nhau sẽ có khoảng cách trên vector space nhỏ hơn các từ khác nhau về mặt nghữ nghĩa.

Mô hình CBOW

Khác với Skip-gram, CBOW là mô hình tối ưu hóa xác suất sinh ra từ t phụ thuộc vào ngữ cảnh m.

$$\prod_{t=1}^{T} P(w^{(t)} \mid w^{(t-m)}, \ldots, w^{(t-1)}, w^{(t+1)}, \ldots, w^{(t+m)}).$$



Tương tự skip-gram, CBOW cũng có phương trình tính toán loss function.

$$-\sum_{t=1}^{T} log P(w^{(t)}) \mid w^{(t-m)}, \ldots, w^{(t-1)}, w^{(t+1)}, \ldots, w^{(t+m)}).$$

CBOW chỉ khác Skip-gram về mặt định nghĩa bài toán, tuy nhiên loss function, quá trình gradient descend vẫn tương tự Skip-gram