TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO GIỮA KÌ

MÔN NHẬP MÔN XỬ LÝ NGÔN NGỮ TỰ NHIÊN

ĐỒ ÁN GIỮA KÌ MÔN NHẬP MÔN XỬ LÝ NGÔN NGỮ TỰ NHIÊN

Người hướng dẫn: **GV. LÊ ANH CƯỜNG**

Người thực hiện: **TRẦN THỊ VẸN – 52100674**

TRƯƠNG BỈNH THUẬN - 52100322

NGUYỄN ĐÌNH DANH – 52100878

Lớp: **21050301**

Khoá: **25**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO GIỮA KÌ

MÔN NHẬP MÔN XỬ LÝ NGÔN NGỮ TỰ NHIÊN

ĐỒ ÁN GIỮA KÌ MÔN NHẬP MÔN XỬ LÝ NGÔN NGỮ TỰ NHIÊN

Người hướng dẫn: **GV. LÊ ANH CƯỜNG**

Người thực hiện: **TRẦN THỊ VẸN – 52100674**

TRƯƠNG BỈNH THUẬN - 52100322

NGUYỄN ĐÌNH DANH – 52100878

Lớp: **21050301**

Khoá: **25**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024

LỜI CẢM ƠN

Trong suốt quá trình học tập và rèn luyện, chúng em đã nhận được rất nhiều sự giúp đỡ tận tình, sự quan tâm, chăm sóc của thầy Lê Anh Cường. Ngoài ra, chúng em còn được thầy truyền đạt những kiến thức về xử lý ảnh hay ho và thú vị, thầy cô còn giúp sinh viên có được nhiều niềm vui trong việc học và cảm thấy thoải mái, … Chúng em xin chân thành cảm ơn các thầy cô rất nhiều trong suốt quá trình học tập này!

Bởi lượng kiến thức của chúng em còn hạn hẹp và gặp nhiều vấn đề trong quá trình học nên báo cáo này sẽ còn nhiều thiếu sót và cần được học hỏi thêm. Chúng em rất mong em sẽ nhận được sự góp ý của quý thầy cô về bài báo cáo này để chúng em rút kinh nghiệm trong những môn học sắp tới. Cuối cùng, chúng em xin chân thành cảm ơn quý thầy cô.

TP Hồ Chí Minh, ngày 20 tháng 03 năm 2024

Sinh viên:

TRẦN THỊ VẸN – 52100674

TRƯƠNG BỈNH THUẬN - 52100322

NGUYỄN ĐÌNH DANH - 52100878

BÁO CÁO CUỐI KÌ ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là sản phẩm của riêng tôi/chúng tôi và được sự hướng dẫn của thầy Lê Anh Cường. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày 20 tháng 03 năm 2024

Tác giả

(ký tên và ghi rõ họ tên)

Trần Thị Vẹn

Nguyễn Đình Danh

Trương Bỉnh Thuận

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

_____

_____

_____

_____

_____

_____

_____

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

Phần đánh giá của GV chấm bài

_____

_____

_____

_____

_____

_____

_____

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

TÓM TẮT

Hiểu biết về các phương pháp Word2Vec:

- CBOW và Skip-gram: Ý nghĩa, mô hình, và cách huấn luyện.
- Fasttext và Glove: Ý nghĩa, mô hình, và cách áp dụng.

Ứng dụng Word2Vec trong bài toán tìm kiếm câu tương tự:

- Hiểu cách sử dụng các phương pháp Word2Vec để tính toán sự tương đồng giữa các câu.
- Xây dựng mô hình để tìm câu trong tập S có sự tương đồng cao nhất với câu X.

Xây dựng mô hình phân loại văn bản:

- Gán nhãn cho các loại văn bản như harassment, hate speech, sexually explicit, và dangerous content.
- Sử dụng các mô hình phân loại văn bản khác nhau để nhận diện loại văn bản.
- Mục tiêu là đạt được độ chính xác cao nhất có thể trong việc phân loại văn bản.

Kỹ năng thực hành và áp dụng lý thuyết vào thực tế:

- Hiểu và áp dụng các phương pháp machine learning và xử lý ngôn ngữ tự nhiên vào các bài toán cụ thể.
- Làm quen với các công cụ và thực hiện các bước cần thiết để xây dựng và đánh giá mô hình.

Tóm lại, qua dự án này, chúng em có cơ hội nắm vững kiến thức về các phương pháp Word2Vec và mô hình phân loại văn bản, cũng như phát triển kỹ năng thực hành và xử lý các bài toán thực tế trong lĩnh vực xử lý ngôn ngữ tự nhiên.

MỤC LỤC

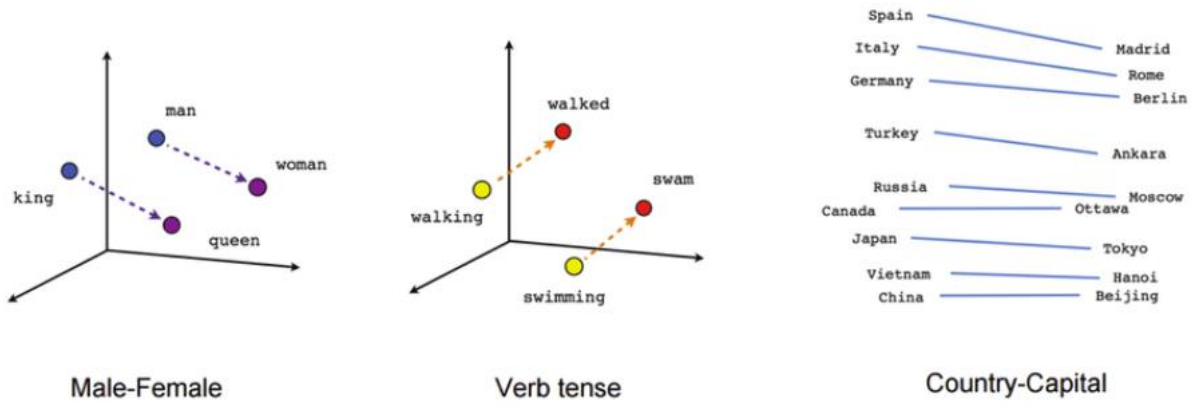
DANH MỤC HÌNH ẢNH, BẢNG BIỂU VÀ ĐỒ THỊ

Danh mục hình ảnh

Danh mục bảng biểu

CHƯƠNG 1 - CƠ SỞ LÝ THUYẾT

1.1 Định nghĩa Word Embeddings - Vector hóa văn bản



Hình 1.1 Word Embeddings

Word embedding là một nhóm các kỹ thuật đặc biệt trong xử lý ngôn ngữ tự nhiên, có nhiệm vụ ánh xạ một từ hoặc một cụm từ trong bộ từ vựng tới một vector số thực. Từ không gian một chiều cho mỗi từ tới không gian các vector liên tục. Các vector từ được biểu diễn theo phương pháp word embedding thể hiện được ngữ nghĩa của các từ, từ đó ta có thể nhận ra được mối quan hệ giữa các từ với nhau(tương đồng, trái nghịch,...).

Ví dụ: "I like eating apple and banana", khi đổi chỗ "apple" và "banana" thì câu không thay đổi về mặt ngữ nghĩa nên 2 từ này có sự tương quan về mặt ngữ nghĩa và cả về ngữ pháp khi cả hai đều là danh từ. Hoặc như giữa "banana" và "yellow" cũng có sự tương quan nhất định khi quả chuối thường có màu vàng.

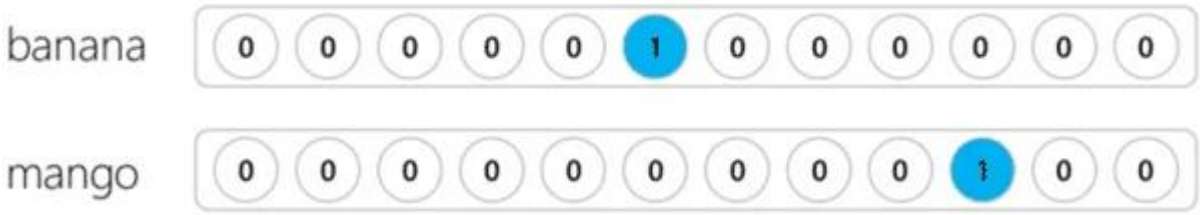
Word embeddings gồm 2 loại cơ bản:

- **Phương pháp Word Embedding cổ điển**
- **Neural Embedding (Vector hóa văn bản theo phương pháp mạng nơ-ron)**

1.2 Phương pháp Word Embedding cổ điển:

1.2.1 Bag of Words (BoW)

Đây là cách biểu diễn vector truyền thống phổ biến nhất được sử dụng. Mỗi từ hoặc n-gram từ sẽ được mô tả là một vector có số chiều bằng đúng số từ trong bộ từ vựng. Tại vị trí tương ứng với vị trí của từ đó trong túi từ, phần tử trong vector đó sẽ được đánh dấu là 1. Những vị trí còn lại sẽ được đánh dấu là 0.



Hình 1.2 Ví dụ biểu diễn One-hot BOW của mỗi từ trong văn bản

Phương pháp BoW thường được sử dụng trong những bài toán phân loại văn bản. Trong đó, tần suất của mỗi từ/ n-gram sẽ được coi là một feature trong văn bản phân loại.

Nhược điểm của phương pháp này là ta không thể xác định được nghĩa thực của mỗi từ và các từ tương quan với chúng.

Trong phương pháp BoW, từ giống nhau sẽ được đánh trọng số như nhau. Phương pháp này không xét đến tần suất xuất hiện của từ hay nhữ cảnh từ. Và trong thực tế, để cần hiểu được nghĩa của mỗi từ, ta cần xác định từ đó trong văn cảnh hơn là xét nghĩa độc lập từ.

1.2.2 TF-IDF

- Count Vector.
- tf-idf Vector.
- Co-occurrence Matrix

TF- IDF (term frequency–inverse document frequency) – tần suất- tần suất đảo nghịch từ. Đây là một phương pháp thống kê, nhằm phản ánh độ quan trọng của mỗi từ hoặc n-gram đối với văn bản trên toàn bộ tài liệu đầu vào. TF-IDF thể hiện trọng số của mỗi từ theo ngữ cảnh văn bản. TF-IDF sẽ có giá trị tăng tỷ lệ thuận với số lần xuất hiện

của từ trong văn bản và số văn bản có chứa từ đó trên toàn bộ tập tài liệu. Phương pháp này giúp cho TF-IDF có tính phân loại cao hơn so với phương pháp trước.

$$tf_i = \frac{n_i}{N_i}$$

Trong đó:
i : 1 .. D
n_i : tần số xuất hiện của từ trong văn bản i.
N_i : tổng số từ trong văn bản i.

$$idf_i = \log_2 \frac{D}{d}$$

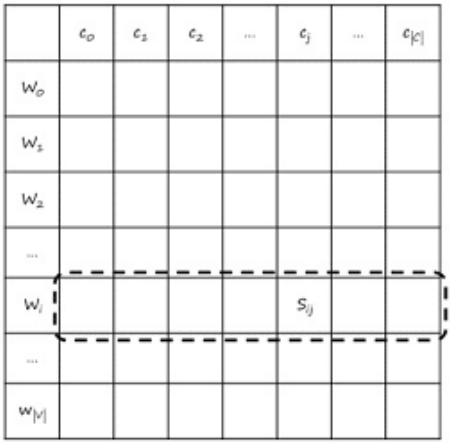
Trong đó:
D: tổng số documents trong tập dữ liệu.
d: số lượng documents có sự xuất hiện của từ.

$$tfidf_i = tf_i \times idf_i$$

Tuy nhiên, ngay cả khi phương pháp TF-IDF dựa trên BOW thể hiện được trọng số của các từ khác nhau trong văn bản, nhưng phương pháp này vẫn không biểu diễn được nghĩa của từ. Đây chính là nhược điểm của hai phương pháp này.

1.2.3 Distributional Embedding

Là phương pháp mà ta có thể xem xét được tổng quan trong toàn bộ ngữ cảnh. Mỗi từ sẽ được biểu diễn trên các thông tin tương hỗ (Mutual Information) với các từ khác trong tập dữ liệu. Thông tin tương hỗ có thể được biểu diễn dưới dạng tần suất xuất hiện trong ma trận đồng xuất hiện trên toàn bộ tập dữ liệu hoặc xem xét trong giới hạn tập dữ liệu lân cận hoặc xem xét trên giới hạn những từ xung quanh.



Hình 1.3 Ma trận thuật toán Distributional Embedding

Phương pháp Distributional Embedding ra đời trước phương pháp Neural Embedding. Nhưng các phương pháp Distributional Embedding giúp ta quan sát được quan trọng của mỗi từ tốt hơn so với Neural Embedding.

Thông thường, Co-occurrence Matrix là một ma trận vuông đối xứng, mỗi hàng hoặc mỗi cột sẽ chính là vector biểu thị của từ tương ứng. Tiếp tục ví dụ trên ta sẽ có ma trận Co-occurrence Matrix:

	John	is	not	fat	thin
John	0	2	0	1	0
is	2	0	1	0	1
not	0	1	0	1	0
fat	1	0	1	0	0
thin	0	1	0	0	0

Hình 1.4 Co-occurrence Matrix

Tuy nhiên, trong thực tế, do số lượng từ vựng nhiều, ta thường chọn cách bỏ đi một số từ không cần thiết (ví dụ như các stopwords) hoặc sử dụng phân tách SVD (Singular Value Decomposition) để giảm kích thước của vector từ nhằm giúp cho biểu

diễn của từ được rõ ràng hơn đồng thời tiết kiệm bộ nhớ dùng để lưu trữ Co-occurrence Matrix (do các Co-occurrence Matrix có kích thước rất lớn).

1.3 Glove (Global Vector)

1.3.1 Định nghĩa và ý nghĩa của GloVe

GloVe (Global Vector) là một trong những phương pháp mới để xây dựng vec-tơ từ (được giới thiệu vào năm 2014), nó thực chất được xây dựng dựa trên Co-occurrence Matrix. GloVe có bản chất là xác suất, ý tưởng xây dựng phương pháp này đến từ tỉ số sau:

$$\frac{P(k|i)}{P(k|j)} \quad P(k|i) = \frac{X_{ik}}{X_i} = \frac{X_{ik}}{\sum_m X_{im}}$$

P(k|i) là xác suất xuất hiện của từ k trong ngữ cảnh của từ i , tương tự với P(k|j) .
Xik : số lần xuất hiện của từ k trong ngữ cảnh của từ i (hoặc ngược lại).
Xi : số lần xuất hiện của từ i trong ngữ cảnh của toàn bộ các từ còn lại ngoại trừ i.
Các giá trị này chính là các mục nhập của Co-occurrence Matrix

Ý tưởng chính của GloVe: độ tương tự ngữ nghĩa giữa hai từ i, j có thể được xác định thông qua độ tương tự ngữ nghĩa giữa từ k với mỗi từ i, j, những từ k có tính xác định ngữ nghĩa tốt chính là những từ làm cho (1) >>1 hoặc xấp chỉ bằng 0. Ví dụ, nếu i là “table”, j là “cat” và k là “chair” thì (1) sẽ khá lớn do “chair” có nghĩa gần hơn với “table” hơn là “cat”, ở trường hợp khác, nếu ta thay k là “ice cream” thì (1) sẽ xấp xỉ bằng 1 do “ice cream” hầu như chẳng lên quan gì tới “table” và “cat”.

Dựa trên tầm quan trọng của (1) , GloVe khởi đầu bằng việc là nó sẽ tìm một hàm F sao cho nó ánh xạ từ các vec-tơ từ trong vùng không gian V sang một giá trị tỉ lệ với (1) . Việc tìm F không đơn giản, tuy nhiên, sau nhiều bước đơn giản hóa cũng như tối ưu, ta có thể đưa nó về bài toán hồi quy với việc minimum cost function sau:

$$J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2$$

Trong đó:

- wi,wj là các vector từ.
- bi, bj là các bias tương ứng (được thêm vào ở các bước đơn giản hóa và tối ưu).
- Xij: mục nhập tương ứng với cặp từ i,j trong Co-occurrence Matrix.

Hàm f được gọi là weighting function, được thêm vào để giảm bớt sự ảnh hưởng của các cặp từ xuất hiện quá thường xuyên, hàm này thỏa 3 tính chất:

- Có giới hạn tại 0.
- Là hàm không giảm.
- Có giá trị nhỏ khi x rất lớn.

Thực tế, có nhiều hàm số thỏa các tính chất trên, nhưng ta sẽ lựa chọn hàm số sau:

$$f(x) = \begin{cases} (x/x_{\max})^{\alpha} & \text{if } x < x_{\max} \\ 1 & \text{otherwise} \end{cases}$$

Với α=3/4

Việc thực hiện minimum cost function J để tìm ra các vec-tơ từ wi, wj thể được thực hiện bằng nhiều cách, trong đó cách tiêu chuẩn nhất là sử dụng Gradient Descent.

1.3.2 Mô hình Glove

Để giải quyết vấn đề trên, GloVe [Pennington et al., 2014], một mô hình embedding từ xuất hiện sau word2vec đã áp dụng mất mát bình phương và đề xuất ba thay đổi trong mô hình skip-gram dựa theo mất mát này.

Ở đây, ta sử dụng các biến phân phối phi xác suất:

$$p'_{ij} = x_{ij} \text{ và } q'_{ij} = \exp(\mathbf{u}_j^\top \mathbf{v}_i)$$

rồi tính log của chúng. Do đó, ta có mất mát bình phương:

$$\left(\log p'_{ij} - \log q'_{ij}\right)^2 = \left(\mathbf{u}_j^\top \mathbf{v}_i - \log x_{ij}\right)^2$$

Ta thêm hai tham số mô hình cho mỗi từ w_i: hệ số điều chỉnh b_i(cho các từ trung tâm) và c_i (cho các từ ngữ cảnh). Thay thế trọng số của mỗi giá trị mất mát bằng hàm $h(x_{ij})$. Hàm trọng số $h(x)$là hàm đơn điệu tăng trong khoảng $[0,1]$.

Do đó, mục tiêu của GloVe là cực tiểu hóa hàm mất mát.

$$\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{V}} h(x_{ij})\left(\mathbf{u}_j^\top \mathbf{v}_i + b_i + c_j - \log x_{ij}\right)^2$$

Ở đây, chúng tôi có một đề xuất đối với việc lựa chọn hàm trọng số $h(x)$: khi $x<c$ (ví dụ $c=100$) thì $h(x)=(x/c)^\alpha$ (ví dụ $\alpha=0.75$), nếu không thì $h(x)=1$. Do $h(0)=0$, ta có thể đơn thuần bỏ qua mất mát bình phương tại $x_{ij}=0$. Khi sử dụng minibatch SGD trong quá trình huấn luyện, ta tiến hành lấy mẫu ngẫu nhiên để được một minibatch x_{ij} khác không tại mỗi bước thời gian và tính toán gradient để cập nhật các tham số mô hình. Các giá trị x_{ij} khác không trên được tính trước trên toàn bộ tập dữ liệu và là thống kê toàn cục của tập dữ liệu. Do đó, tên gọi GloVe được lấy từ "Global Vectors (Vector Toàn cục)".

Chú ý rằng nếu từ w_i xuất hiện trong cửa sổ ngữ cảnh của từ w_j thì từ w_j cũng sẽ xuất hiện trong cửa sổ ngữ cảnh của từ w_i. Do đó, $x_{ij}=x_{ji}$. Không như word2vec, GloVe khớp $\log x_{ij}$ đối xứng thay vì xác suất có điều kiện p_{ij} bất đối xứng. Do đó, vector từ đích trung tâm và vector từ ngữ cảnh của bất kì từ nào đều tương đương nhau trong GloVe. Tuy vậy, hai tập vector từ được học bởi cùng một mô hình về cuối có thể sẽ khác nhau do giá trị khởi tạo khác nhau. Sau khi học tất cả các vector từ, GloVe sẽ sử dụng tổng của vector từ đích trung tâm và vector từ ngữ cảnh để làm vector từ cuối cùng cho từ đó.

- Trong một số trường hợp, hàm mất mát entropy chéo có sự hạn chế. GloVe sử dụng mất mát bình phương và vector từ để khớp các thống kê toàn cục được tính trước dựa trên toàn bộ dữ liệu.
- Vector từ đích trung tâm và vector từ ngữ cảnh của bất kì từ nào là như nhau trong GloVe.

1.4 Phương pháp Neural Embedding

1.4.1 Word2Vec

Thay vì đếm và xây dựng ma trận đồng xuất hiện, word2vec học trực tiếp word vector có số chiều thấp trong quá trình dự đoán các từ xung quanh mỗi từ. Đặc điểm của phương pháp này là nhanh hơn và có thể dễ dàng kết hợp một câu một văn bản mới hoặc thêm vào từ vựng.

Word2vec là một mạng neural 2 lớp với duy nhất 1 tầng ẩn, lấy đầu vào là một corpus lớn và sinh ra không gian vector(với số chiều khoảng vài trăm), với mỗi từ duy nhất trong corpus được gắn với một vector tương ứng trong không gian.

Các word vectors được xác định trong không gian vector sao cho những từ có chung ngữ cảnh trong corpus được đặt gần nhau trong không gian. Dự đoán chính xác cao về ý nghĩa của một từ dựa trên những lần xuất hiện trước đây.

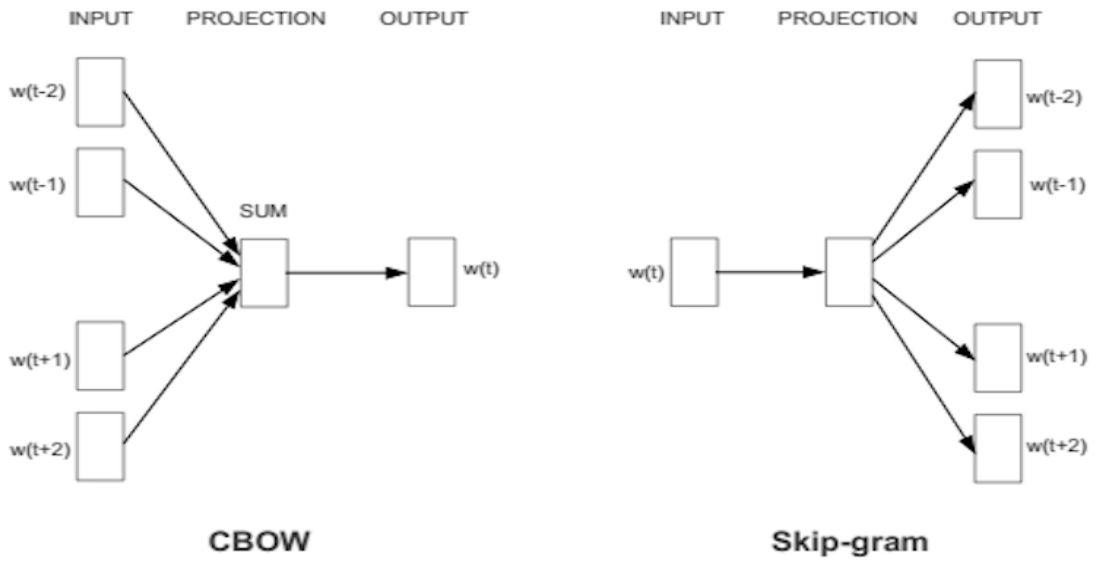
Nếu ta gán nhãn các thuộc tính cho một vector từ giả thiết, thì các vector được biểu diễn theo word2vec sẽ có dạng như sau:

		Vua	Hoàng hậu	Phụ nữ	Công chúa
Hoàng gia		0.99	0.99	0.02	0.98
Nam tính		0.99	0.05	0.01	0.02
Nữ tính		0.05	0.93	0.999	0.94
Tuổi		0.7	0.6	0.5	0.1

Hình 1.5 Vector biểu diễn word2vec

Có 2 cách xây dựng word2vec:

- Sử dụng ngữ cảnh để dự đoán mục tiêu(CBOW).
- Sử dụng một từ để dự đoán ngữ cảnh mục tiêu(skip-gram)(cho kết quả tốt hơn với dữ liệu lớn).

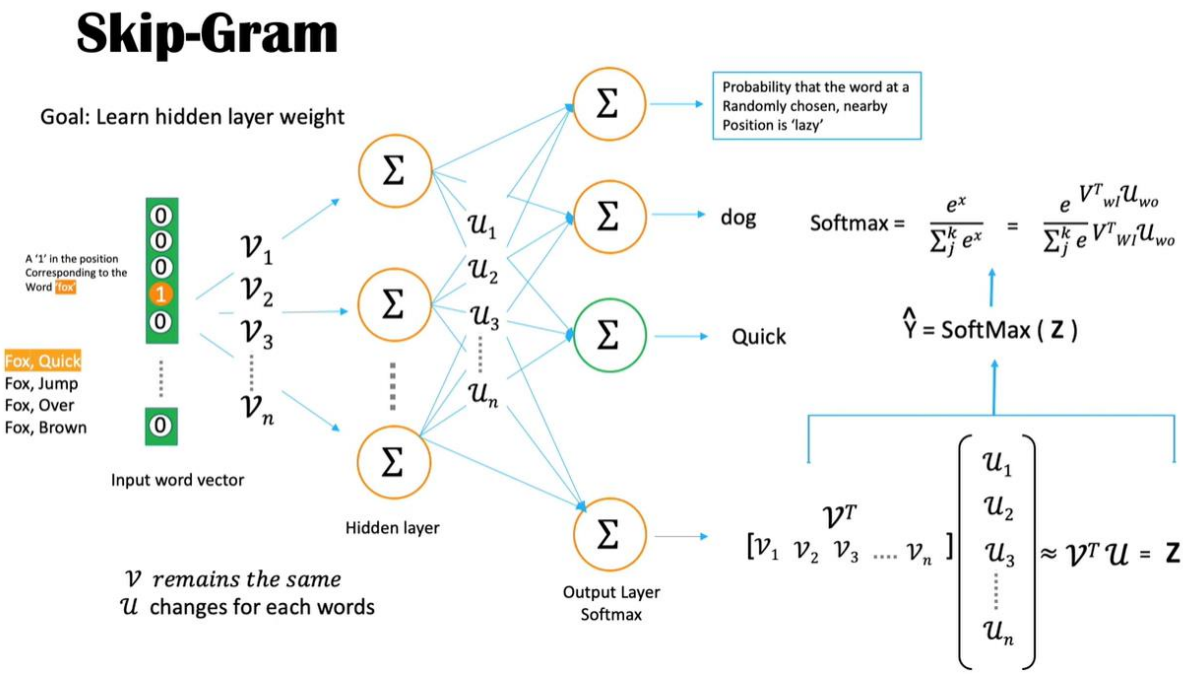


Hình 1.6 CBOW và Skip-gram

1.4.2 Continous Bag of Words

1.4.3 Skip-gram

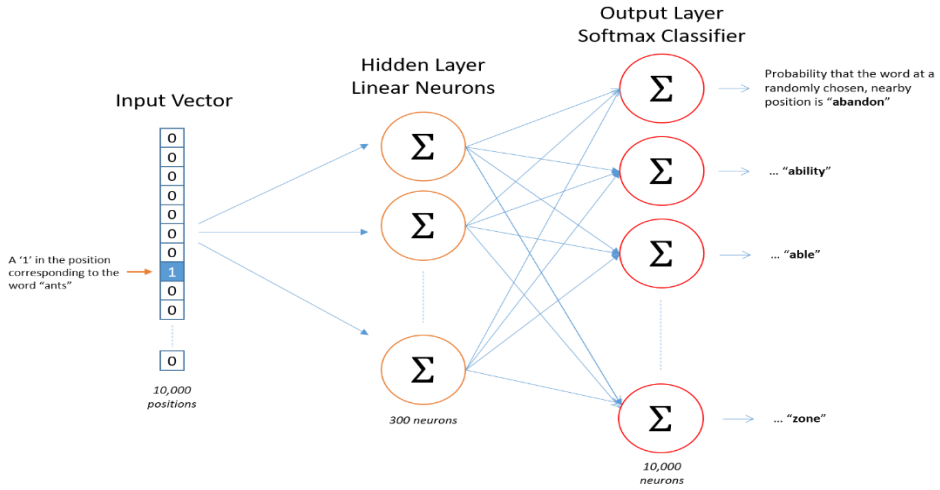
1.4.3.1 Ý nghĩa và định nghĩa mô hình Skip-gram



Hình 1.7 Mô hình và thuật toán kiến trúc của mạng Skip-gram

Mục tiêu: Học trọng số các lớp ẩn, các trọng số này là các words vector

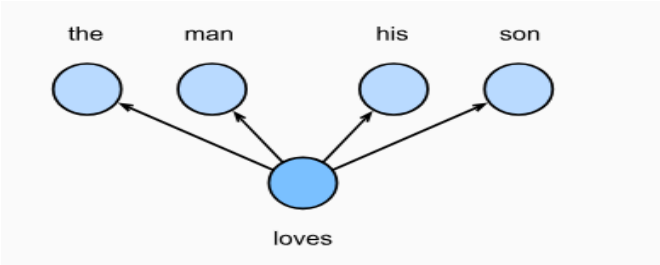
Cách thức: Cho một từ cụ thể ở giữa câu(input word), nhìn vào những từ ở gần và chọn ngẫu nhiên. Mạng neural sẽ cho chúng ta biết xác suất của mỗi từ trong từ vựng về việc trở thành từ gần đó mà chúng ta chọn.



Hình 1.8 Mô hình kiến trúc của mạng Skip-gram

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

Ví dụ, giả sử chuỗi văn bản là “the”, “man”, “loves”, “his” và “son”. Ta sử dụng “loves” làm từ đích trung tâm và đặt kích thước cửa sổ ngữ cảnh bằng 2. Như mô tả trong hình dưới, với từ đích trung tâm “loves”, mô hình skip-gram quan tâm đến xác suất có điều kiện sinh ra các từ ngữ cảnh (“the”, “man”, “his” và “son”) nằm trong khoảng cách không quá 2 từ:



Hình 1.9 Mô hình Skip-gram

P("the","man","his","son"|"loves")

Ta giả định rằng, với từ đích trung tâm cho trước, các từ ngữ cảnh được sinh ra độc lập với nhau. Trong trường hợp này, công thức trên có thể được viết lại thành **P("the"|"loves")·P("man"|"loves")·P("his"|"loves")·P("son"|"loves")**

Quá trình huấn luyện của Skip-gram thường sử dụng một mạng nơ-ron cơ bản, trong đó các biểu diễn từ của từng từ sẽ được cập nhật thông qua quá trình lan truyền ngược (backpropagation) để giảm thiểu sai số giữa xác suất dự đoán và xác suất thực tế. Skip-gram thường được sử dụng trong các tác vụ như phân loại văn bản, tìm kiếm semantec và các tác vụ liên quan đến việc hiểu ngôn ngữ tự nhiên. Đặc biệt, nó thường được ưa chuộng khi làm việc với các tập dữ liệu lớn với số lượng từ vựng lớn.

Trong mô hình skip-gam, mỗi từ được biểu diễn bằng hai vector d chiều để tính xác suất có điều kiện. Giả sử chỉ số của một từ trong từ điển là i, vector của từ được biểu diễn là $v_i \in R^d$ khi từ này là từ đích trung tâm và là $u_i \in R^d$ khi từ này là một từ ngữ cảnh. Gọi c và o lần lượt là chỉ số của từ đích trung tâm w_c và từ ngữ cảnh w_o trong từ điển. Có thể thu được xác suất có điều kiện sinh ra từ ngữ cảnh cho một từ đích trung tâm cho trước bằng phép toán softmax trên tích vô hướng của vector:

$$P(w_o \mid w_c) = \frac{\exp(\mathbf{u}_o^\top \mathbf{v}_c)}{\sum_{i \in \mathcal{V}} \exp(\mathbf{u}_i^\top \mathbf{v}_c)},$$

trong đó, tập chỉ số trong bộ từ vựng là $V=\{0,1,\ldots,|V|-1\}$. Giả sử trong một chuỗi văn bản có độ dài T , từ tại bước thời gian t được ký hiệu là $w^{(t)}$. Giả sử rằng các từ ngữ cảnh được sinh độc lập với từ trung tâm cho trước. Khi kích thước cửa sổ ngữ cảnh là m , hàm hợp lý (likelihood) của mô hình skip-gam là xác suất kết hợp sinh ra tất cả các từ ngữ cảnh với bất kỳ từ trung tâm cho trước nào:

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

Ở đây, bất kỳ bước thời gian nào nhỏ hơn 1 hoặc lớn hơn T đều có thể được bỏ qua.

1.4.3.2 Huấn luyện Mô hình Skip-Gram

Các tham số trong mô hình skip-gram là vector từ đích trung tâm và vector từ ngữ cảnh cho từng từ riêng lẻ. Trong quá trình huấn luyện, chúng ta sẽ học các tham số mô hình bằng cách cực đại hóa hàm hợp lý, còn gọi là ước lượng hợp lý cực đại. Việc này tương tự với việc giảm thiểu hàm mất mát sau đây:

$$-\sum_{t=1}^{T} \sum_{-m \le j \le m,\, j \ne 0} \log P(w^{(t+j)} \mid w^{(t)})$$

Nếu ta dùng SGD, thì trong mỗi vòng lặp, ta chọn ra một chuỗi con nhỏ hơn bằng việc lấy mẫu ngẫu nhiên để tính toán mất mát cho chuỗi con đó, rồi sau đó tính gradient để cập nhật các tham số mô hình. Điểm then chốt của việc tính toán gradient là tính gradient của logarit xác suất có điều kiện cho vector từ trung tâm và vector từ ngữ cảnh. Đầu tiên, theo định nghĩa ta có:

$$\log P(w_o \mid w_c) = \mathbf{u}_o^\top \mathbf{v}_c - \log\left(\sum_{i \in \mathcal{V}} \exp(\mathbf{u}_i^\top \mathbf{v}_c)\right)$$

Thông qua phép tính đạo hàm, ta nhận được giá trị gradient v_c từ công thức trên:

$$\begin{aligned}
\frac{\partial \log P(w_o \mid w_c)}{\partial \mathbf{v}_c} &= \mathbf{u}_o - \frac{\sum_{j \in \mathcal{V}} \exp(\mathbf{u}_j^\top \mathbf{v}_c)\mathbf{u}_j}{\sum_{i \in \mathcal{V}} \exp(\mathbf{u}_i^\top \mathbf{v}_c)} \\
&= \mathbf{u}_o - \sum_{j \in \mathcal{V}} \left(\frac{\exp(\mathbf{u}_j^\top \mathbf{v}_c)}{\sum_{i \in \mathcal{V}} \exp(\mathbf{u}_i^\top \mathbf{v}_c)}\right) \mathbf{u}_j \\
&= \mathbf{u}_o - \sum_{j \in \mathcal{V}} P(w_j \mid w_c)\mathbf{u}_j.
\end{aligned}$$

Phép tính cho ra xác suất có điều kiện cho mọi từ có trong từ điển với từ đích trung tâm w_c cho trước. Sau đó, ta lại sử dụng phương pháp đó để tìm gradient cho các vector từ khác.

Sau khi huấn luyện xong, với từ bất kỳ có chỉ số là i trong từ điển, ta sẽ nhận được tập hai vector từ v_i và u_i. Trong các ứng dụng xử lý ngôn ngữ tự nhiên, vector từ đích trung tâm trong mô hình skip-gram thường được sử dụng để làm vector biểu diễn một từ.

Model details:

- Xây dựng bộ từ vựng
- Biểu diễn mỗi từ thành các one-hot-vector
- Đầu ra là một vector duy nhất, có kích thước bằng kích thước của bộ từ vựng, thể hiện xác suất của mỗi từ được là lân cận của từ đầu vào.
- Không có hàm kích hoạt trên tầng ẩn
- Hàm kích hoạt trên tầng output là softmax
- Trong quá trình training, input là 1 one-hotvector, ouput cũng là 1 one-hot-vector
- Trong quá trình đánh giá sau khi training, đầu ra phải là 1 phân bố xác suất

Word2vec cải tiến

Có 3 cải tiến cơ bản cho mô hình word2vec truyền thống:

- Xử lý các cặp từ thông dụng hoặc cụm từ như là một từ đơn
- Loại bỏ các từ thường xuyên lặp lại để giảm số lượng các ví dụ huấn luyện
- Sửa đổi mục tiêu tối ưu hóa bằng một kỹ thuật gọi là "Negative Sampling"

Cải tiến 1: Xử lý cụm từ như một từ đơn

Ví dụ các từ như "thành_phố_Cảng" có nghĩa khác nhau với từng từ "thành_phố" và "cảng",...

- Chúng ta sẽ coi như đó là một từ duy nhất, với word vector của riêng mình.
- Điều này sẽ làm tăng kích thước từ vựng. (Tìm hiểu thêm về word2phrase)

Cải tiến 2: Loại bỏ các từ thường xuyên lặp lại

Các từ thường xuyên lặp lại như "các", "những",… không cho chúng ta biết thêm nhiều hơn về ý nghĩa của những từ đi kèm nó, và chúng cũng xuất hiện trong ngữ cảnh của khá nhiều từ.

Chúng ta sẽ xác định xác suất loại bỏ, giữ lại một từ trong từ vựng thông qua tần suất xuất hiện của nó.

Cải tiến 3: Negative Sampling

Mỗi mẫu huấn luyện chỉ thay đổi một tỷ lệ phần trăm nhỏ các trọng số, thay vì tất cả chúng.

Nhớ lại: Khi huấn luyện mạng với 1 cặp từ, đầu ra của mạng sẽ là 1 one-hot vector, neural đúng thì đưa ra 1 còn hàng ngàn neural khác thì đưa ra 0.

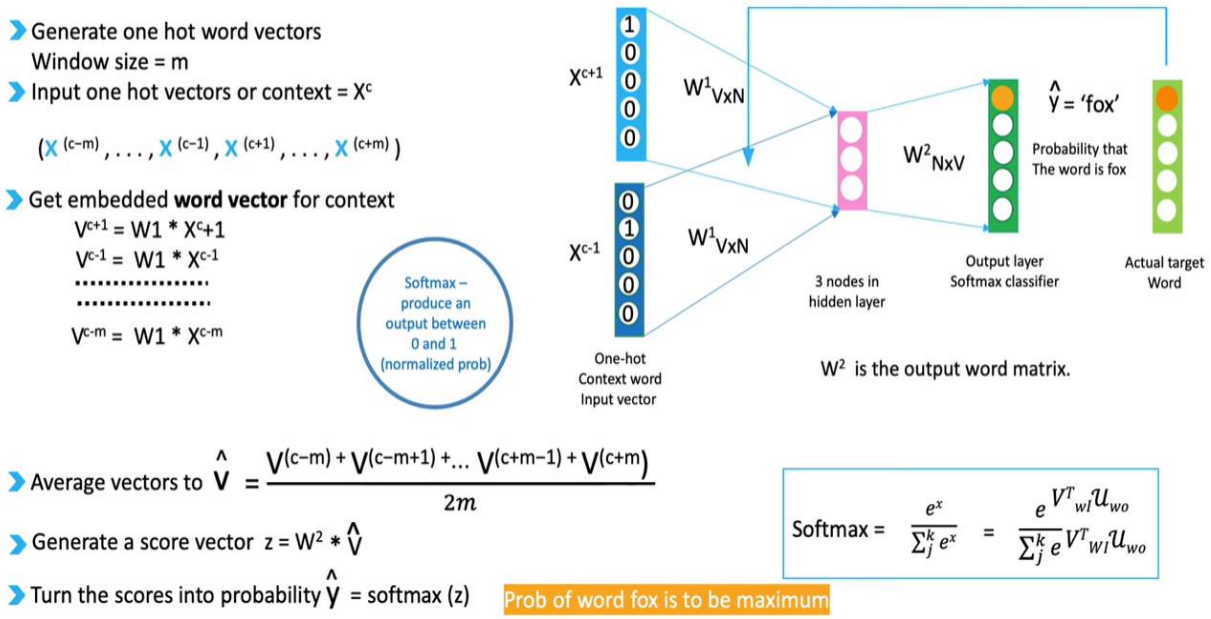
Chọn ngẫu nhiên 1 số lượng nhỏ các neural "negative" kết hợp với neural "positive" để cập nhật trọng số.(chọn là 5-20 hoạt động tốt với các bộ dữ liệu nhỏ, 2-5 với bộ dữ liệu lớn).

1.4.3.3 Khi nào dùng Skip-gram

- Hoạt động tốt với lượng nhỏ dữ liệu đào tạo.(training data)
- Khi từ và cụm từ rất hiếm.

1.4.4 Mô hình Túi từ Liên tục (CBOW)

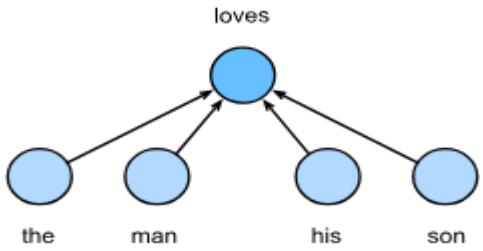
1.4.4.1 Ý nghĩa và định nghĩa mô hình CBOW



Hình 1.10 Mô hình và thuật toán CBOW

Mô hình túi từ liên tục (Continuous bag of words - CBOW) tương tự như mô hình skip-gram. Khác biệt lớn nhất là mô hình CBOW giả định rằng từ đích trung tâm được tạo ra dựa trên các từ ngữ cảnh phía trước và sau nó trong một chuỗi văn bản.

Với cùng một chuỗi văn bản gồm các từ “the”, “man”, “loves”, “his” và “son”, trong đó “love” là từ đích trung tâm, với kích thước cửa sổ ngữ cảnh bằng 2, mô hình CBOW quan tâm đến xác suất có điều kiện để sinh ra từ đích “love” dựa trên các từ ngữ cảnh “the”, “man”, “his” và “son”



Hình 1.11 Mô hình CBOW

P("loves"|"the","man","his","son").

Vì có quá nhiều từ ngữ cảnh trong mô hình CBOW, ta sẽ lấy trung bình các vector từ của chúng và sau đó sử dụng phương pháp tương tự như trong mô hình skip-gram để tính xác suất có điều kiện. Giả sử $v^i \in R^d$ và $u^i \in R^d$ là vector từ ngữ cảnh và vector từ đích trung tâm của từ có chỉ số i trong từ điển (lưu ý rằng các ký hiệu này ngược với các ký hiệu trong mô hình skip-gram). Gọi c là chỉ số của từ đích trung tâm w_c, và $o_1,\ldots,o_{2m}$ là chỉ số các từ ngữ cảnh $w_{o1},\ldots,w_{o2m}$ trong từ điển. Do đó, xác suất có điều kiện sinh ra từ đích trung tâm dựa vào các từ ngữ cảnh cho trước là:

$$P(w_c \mid w_{o_1}, \ldots, w_{o_{2m}}) = \frac{\exp\left(\frac{1}{2m}\mathbf{u}_c^\top (\mathbf{v}_{o_1} + \ldots, +\mathbf{v}_{o_{2m}})\right)}{\sum_{i \in \mathcal{V}} \exp\left(\frac{1}{2m}\mathbf{u}_i^\top (\mathbf{v}_{o_1} + \ldots, +\mathbf{v}_{o_{2m}})\right)}$$

Để rút gọn, ký hiệu $W_o=\{w_{o1},\ldots,w_{o2m}\}$, và $\bar{\mathbf{v}}_o=(v_{o1}+\ldots,+v_{o2m})/(2m)$. Phương trình trên được đơn giản hóa thành:

$$P(w_c \mid \mathcal{W}_o) = \frac{\exp\left(\mathbf{u}_c^\top \bar{\mathbf{v}}_o\right)}{\sum_{i \in \mathcal{V}} \exp\left(\mathbf{u}_i^\top \bar{\mathbf{v}}_o\right)}$$

Cho một chuỗi văn bản có độ dài T, ta giả định rằng từ xuất hiện tại bước thời gian t là $w^{(t)}$, và kích thước của cửa sổ ngữ cảnh là m. Hàm hợp lý của mô hình CBOW là xác suất sinh ra bất kỳ từ đích trung tâm nào dựa vào những từ ngữ cảnh.

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

Quá trình huấn luyện của CBOW cũng thường sử dụng một mạng nơ-ron cơ bản, trong đó các biểu diễn từ của từng từ sẽ được cập nhật thông qua quá trình lan truyền ngược (backpropagation) để giảm thiểu sai số giữa xác suất dự đoán và xác suất thực tế.

CBOW thường được sử dụng trong các tác vụ như phân loại văn bản, tìm kiếm semantec và các tác vụ liên quan đến việc hiểu ngôn ngữ tự nhiên. Đặc biệt, CBOW thường phù hợp khi làm việc với các tập dữ liệu lớn với số lượng từ vựng lớn.

1.4.4.2 Huấn luyện mô hình CBOW

Quá trình huấn luyện mô hình CBOW khá giống với quá trình huấn luyện mô hình skip-gram. Ước lượng hợp lý cực đại của mô hình CBOW tương đương với việc cực tiểu hóa hàm mất mát:

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

Lưu ý rằng

$$\log P(w_c \mid \mathcal{W}_o) = \mathbf{u}_c^\top \bar{\mathbf{v}}_o - \log \left(\sum_{i \in \mathcal{V}} \exp\left(\mathbf{u}_i^\top \bar{\mathbf{v}}_o\right) \right)$$

Thông qua phép đạo hàm, ta có thể tính log của xác suất có điều kiện của gradient của bất kỳ vector từ ngữ cảnh nào v_{oi}(i=1,…,2m) trong công thức trên.

$$\frac{\partial \log P(w_c \mid \mathcal{W}_o)}{\partial \mathbf{v}_{o_i}} = \frac{1}{2m} \left(\mathbf{u}_c - \sum_{j \in \mathcal{V}} \frac{\exp(\mathbf{u}_j^\top \bar{\mathbf{v}}_o)\mathbf{u}_j}{\sum_{i \in \mathcal{V}} \exp(\mathbf{u}_i^\top \bar{\mathbf{v}}_o)} \right) = \frac{1}{2m} \left(\mathbf{u}_c - \sum_{j \in \mathcal{V}} P(w_j \mid \mathcal{W}_o)\mathbf{u}_j \right)$$

Sau đó, ta sử dụng cùng phương pháp đó để tính gradient cho các vector của từ khác. Không giống như mô hình skip-gam, trong mô hình CBOW ta thường sử dụng vector từ ngữ cảnh làm vector biểu diễn một từ.

Tóm tắt:

- Vector từ là một vector được sử dụng để biểu diễn một từ. Kỹ thuật ánh xạ các từ sang vector số thực còn được gọi là kỹ thuật embedding từ.

- Word2vec bao gồm cả mô hình túi từ liên tục (CBOW) và mô hình skip-gam. Mô hình skip-gam giả định rằng các từ ngữ cảnh được sinh ra dựa trên từ đích trung tâm. Mô hình CBOW giả định rằng từ đích trung tâm được sinh ra dựa trên các từ ngữ cảnh.

1.4.4.3 Khi nào dùng CBOW

- Train nhanh hơn nhiều so với Skip-gram
- Bộ nhớ thấp. CBOW không cần phải có yêu cầu RAM lớn.
- Độ chính xác tốt hơn một chút cho frequent words

Cả CBOW và Skip-Gram đều là các mô hình dự đoán. Trong đó, các thuật toán chỉ xem xét được ngữ cảnh xung quanh từ mục tiêu nhưng không đề cập được về ngữ cảnh toàn văn bản. Thuật toán GloVe dựa trên tương phản có lợi với cùng dự đoán của ma trận đồng xuất hiện sử dụng trong thuật toán Distributional Embedding, nhưng sử dụng phương pháp Neural Embedding để phân tích ma trận đồng xuất hiện thành những vector có ý nghĩa và tỷ trọng hơn.

Mặc dù thuật toán GloVe nhanh hơn Word2Vec, nhưng cả GloVe và Word2Vec đều không hiển thị để cung cấp kết quả tốt và rõ ràng hơn thay vì cả hai nên được đánh giá cho một tập dữ liệu nhất định.

1.5 FastText

FastText, được xây dựng trên Word2Vec bằng cách học các biểu diễn vectơ cho mỗi từ và n-gram được tìm thấy trong mỗi từ. Các giá trị của các biểu diễn sau đó được tính trung bình thành một vectơ ở mỗi bước đào tạo. Trong khi điều này bổ sung rất nhiều tính toán bổ sung cho việc đào tạo, nó cho phép nhúng từ để mã hóa thông tin từ phụ. Các vectơ FastText đã được chứng minh là chính xác hơn các vectơ Word2Vec bằng một số biện pháp khác nhau.

TÀI LIỆU THAM KHẢO

[1] "Natural Language Processing with Python" by Steven Bird, Ewan Klein, and Edward Loper

[2] "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition" by Daniel Jurafsky and James H. Martin

[3] "Foundations of Statistical Natural Language Processing" by Christopher D. Manning and Hinrich Schütze

[4] "Deep Learning for Natural Language Processing" by Palash Goyal, Sumit Pandey, Karan Jain, and Karthik Raman

[5] "Introduction to Information Retrieval" by Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze