Trường Công nghệ Thông tin và Truyền thông
Khoa Công nghệ thông tin

XỬ LÝ NGÔN NGỮ TỰ NHIÊN

GVHD: Lâm Nhựt Khang
lnkhang@ctu.edu.vn

CHƯƠNG 3

VECTOR NGỮ NGHĨA

Nội dung Chương

1. Giới thiệu
2. Ngữ nghĩa của từ và vector ngữ nghĩa
3. Phương pháp biểu diễn từ
4. Đo lường độ tương đồng giữa các vector
5. Pointwise Mutual Information
6. Nhúng từ

Nội dung Chương

1. **Giới thiệu**
2. Ngữ nghĩa của từ và vector ngữ nghĩa
3. Phương pháp biểu diễn từ
4. Đo lường độ tương đồng giữa các vector
5. Pointwise Mutual Information
6. Nhúng từ

Giới thiệu

- Ngữ cảnh (*context*) đóng vai trò quan trọng đến ngữ nghĩa của từ trong văn bản
- Để hiểu nghĩa của từ, chúng ta cần kết hợp giữa ngữ pháp hay cấu trúc câu, cách các từ được sử dụng trong thực tế, và ngữ cảnh diễn ra.
- Những từ có ngữ nghĩa (*similar meanings*) càng gần giống nhau sẽ càng có xu hướng xuất hiện trong ngữ cảnh giống nhau (*similar contexts*).
- Các từ có mối liên hệ về ngữ nghĩa sẽ thường xuất hiện cùng nhau trong một ngữ cảnh cụ thể.
- Ví dụ, các từ "sinh viên", "giảng viên", "đại học", "học phần" và "tín chỉ" thường sẽ xuất hiện cùng nhau trong miền dữ liệu về môi trường đào tạo bậc đại học.

Giới thiệu

- Giả thuyết phân phối nghĩa (*distributional hypothesis of meaning*) [116] [117] cho rằng "nghĩa của từ được xác định bởi các từ đồng xuất hiện với chúng".

- **Man : Woman :: King : ??**

cow, drink, babies, calcium…

Nội dung Chương

1. Giới thiệu
2. **Ngữ nghĩa của từ và vector ngữ nghĩa**
3. Phương pháp biểu diễn từ
4. Đo lường độ tương đồng giữa các vector
5. Pointwise Mutual Information
6. Nhúng từ

Vector ngữ nghĩa

- Vector ngữ nghĩa (*semantic vector*) là phương pháp biểu diễn nghĩa của một từ trong NLP.
- Ý tưởng chính của phương pháp là sử dụng một điểm trong không gian ngữ nghĩa đa chiều để thể hiện nghĩa của từ và cách phân bổ của các từ lân cận (hoặc môi trường ngữ pháp).
- Các từ có cách phân bố giống nhau sẽ có nghĩa tương tự nhau.

➢ Các vector biểu diễn các từ như vậy được gọi là "nhúng" (*embeddings*).

Các khái niệm liên quan

Theo từ điểm Cambrdige, từ "corpus" có hai nghĩa là "a collection of written ..." và (ii) "a body or the main part of an organ".

- Mỗi nghĩa của từ là một *sense*.
- Từ "corpus" có từ loại là "danh từ" và danh từ số nhiều của "corpus" là "corpora".
- Từ điển không tách biệt định nghĩa từ "corpus" và "corpora";
- Từ "corpus" là một *lemma*.
- Mỗi lemma có thể có nhiều nghĩa (*homonymous*) và các nghĩa của các từ có thể có các mối liên hệ (*relationship*) với nhau.

Words, Lemmas, Senses, Definitions

lemma **sense** **definition**

pepper, *n.*

Pronunciation: Brit. /ˈpɛpə/ , U.S. /ˈpɛpər/

Forms: OE **peopor** (*rare*), OE **pipcer** (transmission error), OE **pipor**, OE **pipur** (*rare*

Frequency (in current use):

Etymology: A borrowing from Latin. **Etymon:** Latin *piper*.

< classical Latin *piper*, a loanword < Indo-Aryan (as in ancient Greek *πέπερι*); compare Sa

I. The spice or the plant.

1.

a. A hot pungent spice derived from the prepared fruits (peppercorns) of the pepper plant, *Piper nigrum* (see sense 2a), used from early times to season food, either whole or ground to powder (often in association with salt). Also (locally, chiefly with distinguishing word): a similar spice derived from the fruits of certain other species of the genus *Piper*; the fruits themselves.

The ground spice from *Piper nigrum* comes in two forms, the more pungent *black pepper*, produced from black peppercorns, and the milder *white pepper*, produced from white peppercorns: see BLACK *adj.* and *n.* Special uses 5a, PEPPERCORN *n.* 1a, and WHITE *adj.* and *n.*[1] Special uses 7b(a).

a. The plant *Piper nigrum* (family Piperaceae), a climbing shrub indigenous to South Asia and also cultivated elsewhere in the tropics, which has alternate stalked entire leaves, with pendulous spikes of small green flowers opposite the leaves, succeeded by small berries turning red when ripe. Also more widely: any plant of the genus *Piper* or the family Piperaceae.

b. Usu. with distinguishing word: any of numerous plants of other families having hot pungent fruits or leaves which resemble pepper (1a) in taste and in some cases are used as a substitute for it.

c. *U.S.* The California pepper tree, *Schinus molle*. Cf. PEPPER TREE *n.* 3.

3. Any of various forms of capsicum, esp. *Capsicum annuum* var. *annuum*. Originally (chiefly with distinguishing word): any variety of the *C. annuum* Longum group, with elongated fruits having a hot, pungent taste, the source of cayenne, chilli powder, paprika, etc., or of the perennial *C. frutescens*, the source of Tabasco sauce. Now frequently (more fully ***sweet pepper***): any variety of the *C. annuum* Grossum group, with large, bell-shaped or apple-shaped, mild-flavoured fruits, usually ripening to red, orange, or yellow and eaten raw in salads or cooked as a vegetable. Also: the fruit of any of these capsicums.

Sweet peppers are often used in their green immature state (more fully ***green pepper***), but some new varieties remain green when ripe.

Lemma pepper

- Sense 1: spice from pepper plant
- Sense 2: the pepper plant itself
- Sense 3: another similar plant (Jamaican pepper)
- Sense 4: another plant with peppercorns (California pepper)
- Sense 5: *capsicum* (i.e. chili, paprika, bell pepper, etc)

Quan hệ đồng nghĩa

Synonym

- Là khi một từ có nghĩa giống nhau hoặc gần giống nhau với từ khác.
- Hai từ được xem là đồng nghĩa khi ta có thể thay đổi chúng cho nhau trong bất kỳ câu nào mà không thay đổi nghĩa của câu.
- Ví dụ, “xe lửa” và “tàu hỏa”, “con cọp” và “con hổ” là những cặp từ đồng nghĩa. Tuy nhiên, không có nhiều từ được xem là đồng nghĩa hoàn toàn.
- Từ “chết” và “hy sinh” có đồng nghĩa không?

Từ giống nhau

Word similarity

- Là các từ chia sẻ chung một số yếu tố nghĩa.
- Các từ không có nhiều từ đồng nghĩa (*synonym word*), nhưng lại có nhiều từ giống nhau.
- Ví dụ, từ “măng cụt” và “sầu riêng” không phải là từ đồng nghĩa nhưng là từ giống nhau do chúng đều là một loại trái cây.
- Trong NLP, từ giống nhau giúp xác định các cụm hoặc câu giống nhau.

Mối quan hệ trái nghĩa

Antonym

- Là khi một từ có nghĩa trái ngược với một từ khác ở một thuộc tính nào đó
- Ví dụ như từ "ngắn" và "dài", "đúng" và "sai".
- Các từ có mối quan hệ trái nghĩa chỉ khác nhau về một khía cạnh nào đó, nhưng lại giống nhau ở các khía cạnh còn lại.

Mối quan hệ liên quan của từ

Word relatedness hay word association

- Là khi nghĩa của từ có mối liên hệ theo những cách khác với mối quan hệ giống nhau.
- Ví dụ, từ "trà" và "ly" không giống nhau, nhưng có mối quan hệ liên quan với nhau trong sự việc chung là uống trà bằng ly.
- Các mối quan hệ liên quan phổ biến bao gồm:
  - Lĩnh vực ngữ nghĩa (semantic field): đây là mối quan hệ liên quan phổ biến nhất, giúp xác định cấu trúc chủ đề trong tài liệu. Các từ "máy bay", "phi công", "tiếp viên",... có mối quan hệ trong lĩnh vực liên quan "hàng không".
  - Khung ngữ nghĩa (semantic frame): mỗi khung ngữ nghĩa là một tập hợp các từ biểu thị quan điểm, thuộc tính, đặc trưng hoặc tham dự vào một sự kiện cụ thể. Học phần này sẽ không thảo luận chi tiết về khung ngữ nghĩa.
  - Mối quan hệ phân loại (taxonomic relation)

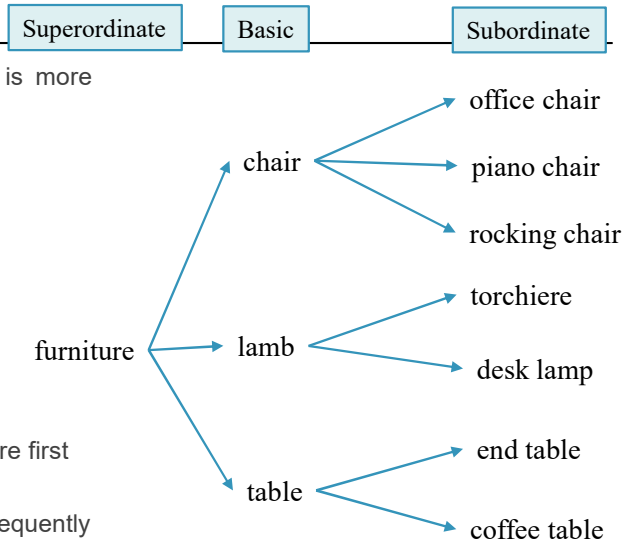
Mối quan hệ liên quan của từ

Word relatedness hay word association

- Các mối quan hệ liên quan phổ biến bao gồm:
  - Mối quan hệ phân loại (taxonomic relation):
    - Một từ có mối quan hệ hyponymy (hay subordinate) với một từ khác nếu nghĩa của từ cụ thể và chi tiết hơn;
    - Ngược lại, một từ có mối quan hệ hypernymy (hay superordinate) với một từ khác nếu nghĩa của từ là tổng quát hơn.
    - Từ "măng cụt" là hyponym của từ "trái cây" và "trái cây" là hypernym của từ "măng cụt".

Superordinate	vehicle	fruit	furniture
Subordinate	car	mango	chair

Super/Subordinate



One sense is a subordinate of another if the first sense is more specific, denoting a subclass of the other

- *car* is a subordinate of *vehicle*
- *mango* is a subordinate of *fruit*

Conversely superordinate

- *vehicle* is a superordinate of *car*
- *fruit* is a superordinate of *mango*

The basic level

It is the level of distinctive actions

It is the level which is learned earliest and at which things are first named

It is the level at which names are shortest and used most frequently

Mối quan hệ liên quan của từ

Word relatedness hay word association

- Ngoài ra còn có mối quan hệ kéo theo (entailment):
  - như từ "ngủ" và "ngáy" có mối quan hệ kéo theo do ngủ rồi mới ngáy;
  - từ "kết hôn" và "ly dị" cũng có mối quan hệ kéo theo do kết hôn rồi mới ly dị.

Mối quan hệ cảm xúc

- Mối quan hệ cảm xúc (connation): là các từ có hàm ý tình cảm hoặc cảm xúc, ý kiến của con người
  - positive connotations (happy)
  - negative connotations (sad)
- Evaluation (sentiment!)
  - positive evaluation (great, love)
  - negative evaluation (terrible, hate).
- Words seem to vary along 3 affective dimensions (Osgood et al.,1957):
  - valence: the pleasantness of the stimulus
  - arousal: the intensity of emotion provoked by the stimulus
  - dominance: the degree of control exerted by the stimulus

	Word	Score
Valence	love	1.000
	happy	1.000
Arousal	elated	0.960
	frenzy	0.965
Dominance	powerful	0.991
	leadership	0.983
Valence	toxic	0.008
	nightmare	0.005
Arousal	mellow	0.069
	napping	0.046
Dominance	weak	0.045
	empty	0.081

Nội dung Chương

1. Giới thiệu
2. Ngữ nghĩa của từ và vector ngữ nghĩa
3. **Phương pháp biểu diễn từ**
4. Đo lường độ tương đồng giữa các vector
5. Pointwise Mutual Information
6. Nhúng từ

Phương pháp biểu diễn từ

- Khi xử lý dữ liệu, văn bản đầu vào sẽ được chia thành các đơn vị được gọi là token. Mỗi token có thể là một từ (word), một số, hay một dấu câu.
- Trong ngôn ngữ tự nhiên, từ là đơn vị cơ bản nhỏ nhất mang ý nghĩa.
- Tuy nhiên, việc sử dụng chuỗi ký tự để biểu diễn một từ sẽ không hiệu quả trong lĩnh vực xử lý ngôn ngữ tự nhiên.

Phương pháp biểu diễn từ

Từ đơn (single word)		Từ ghép (compound word)			
Vie	**Eng**	**Vie**	**Eng**	**Vie**	**Eng**
nhà	house	mua bán	buy and sell	mẫu giáo	kindergarden
lụa	silk	đồng ruộng	rice field	thổ cẩm	brocade
nhặt	pick up	mè đen	black sesame	vàng vàng	yellowish
mua	buy	cây cối	trees	gật gà gật gù	nod repeatedly out of satisfaction
bán	sell	đường xá	street	lải nhải	annoyingly insistent

Phương pháp dictionary lookup

- Với tập văn bản *D*, từ điển *V* gồm các từ duy nhất w, từ điển "dictionary lookup" được xây dựng bằng cách tạo một ánh xạ (*map*) giữa từ w_i và một ID duy nhất (thường là số integer);
- Mỗi từ duy nhất trong từ điển được gán một ID.

Vậy với mỗi từ cho trước, chúng ta sẽ dò tìm trong dicionary lookup, nếu từ đó tồn tại dicionary lookup thì sẽ trả về một giá trị ID tương ứng của từ đó trong từ điển.

Dictionary lookup

ID	Từ
1	bạn
2	buồn
3	ghét
4	thương
5	yêu

Phương pháp one-hot encoding

- Ý tưởng của phương pháp này là xây dựng một vector biểu diễn từ *w* ở vị trí thứ *i* trong từ điển, gọi là w_i.
- Vector từ hay còn gọi là word vector là một vector có trọng số.
- Trong phương pháp one-hot encoding, nếu từ điển *V* có độ dài là *N*, thì vector từ sẽ có *(N+1)* chiều (*dimension*), trong đó chiều *N+1* được sử dụng cho các từ không có trong từ điển hay còn gọi là các từ UNK hay OOV.
- Vector biểu diễn từ w_i sẽ chỉ có duy nhất vị trí thứ *i* có giá trị 1 còn lại đều có giá trị 0.
- Các từ mã hóa sẽ phân biệt chữ hoa và chữ thường.

Phương pháp one-hot encoding

Xét ví dụ từ điển *V* gồm 5 từ {“bạn”, “buồn”, “ghét”, “thương”, “yêu”},

One-hot encoding của các từ được biểu diễn như sau:

One-hot encoding	bạn	buồn	ghét	thương	yêu	unknown
Vector biểu diễn từ “yêu”	0	0	0	0	1	0
Vector biểu diễn từ “bạn”	1	0	0	0	0	0
Vector biểu diễn từ “học”	0	0	0	0	0	1

Phương pháp one-hot encoding

Hạn chế của phương pháp biển diễn này:

- Trong trường hợp số lượng từ trong từ điển là lớn, vector từ có số chiều khá lớn và sẽ chứa rất nhiều số 0 hay còn gọi là "sparse vector".
- Phương pháp one-hot encoding cũng không hiệu quả khi biểu diễn một từ mới không có trong từ điển.
- Thêm vào đó phương pháp one-hot vector không biểu diễn (*capture*) được mối quan hệ ngữ nghĩa (*semantic relatednes*) giữa các từ.
- Ví dụ sự khác biệt giữa từ "học sinh" và "giáo viên" so với sự khác biệt giữa "học sinh" và "con cá" sẽ như nhau nếu sử dụng one-hot encoding để biểu diễn từ.

25

Mô hình túi từ

Mô hình túi từ BOW (Bag-of-Word):

- Đầu tiên sẽ xây dựng một bộ từ vựng *V* chứa các từ *w* duy nhất trong tài liệu,
- Sau đó mô hình hóa văn bản bằng cách đếm số lần xuất hiện (*count*) của mỗi từ *w* trong bộ từ vựng *V* xuất hiện trong tài liệu đó.

Phương pháp biểu diễn này không chứa thông tin về vị trí xuất hiện của từ mà chỉ chứa thông tin về số lần xuất hiện của từ trong văn bản.

Phương pháp BOW biểu diễn văn bản sẽ sử dụng một vector x=[0, 1, 2, 3, 0….] trong đó x_j là số lần xuất hiện của từ thứ *j*.

26

Phương pháp TF-IDF

- Giúp khắc phục vấn đề của phương pháp BOW.
- TF-IDF được dùng để đánh giá mức độ quan trọng của từ trong toàn văn bản, giúp xác định tần số tương đối của một từ, so với tỷ lệ của toàn văn bản.

$$TF - IDF_{w,d} = TF_{w,d} * IDF_w$$

$$TF - IDF_{w,d} = \frac{số\ lần\ w\ xuất\ hiện\ trong\ d}{tổng\ số\ từ\ trong\ d} * log \frac{số\ tài\ liệu\ M}{số\ tài\ liệu\ có\ w\ xuất\ hiện}$$

27

Phương pháp TF-IDF

$$TF - IDF_{w,d} = TF_{w,d} * IDF_w$$

$$TF - IDF_{w,d} = \frac{số\ lần\ w\ xuất\ hiện\ trong\ d}{tổng\ số\ từ\ trong\ d} * log \frac{số\ tài\ liệu\ M}{số\ tài\ liệu\ có\ w\ xuất\ hiện}$$

- Giá trị TF-IDF của từ *w* không những phụ thuộc vào tần suất xuất hiện của từ *w* trong tài liệu *d* đó mà còn phụ thuộc vào tần suất xuất hiện thường xuyên của từ *w* trong toàn bộ tập văn bản *D*.
- Giá trị TF-IDF còn được dùng để xác định các từ trọng tâm trong văn bản.
- Ví dụ: tính giá trị TF-IDF của từ trong hai câu
  - s*ent*$_1$: “Anh ấy đang chạy trên đường”
  - s*ent*$_2$: “Xe đang chạy trên cầuvượt”.

Giả sử 2 câu này sử dụng bộ công cụ tách từ có kết quả như sau s*ent*$_1$: “Anh_ấy / đang / chạy / trên / đường” và s*ent*$_2$: “Xe / đang / chạy / trên / cầu_vượt”.

28

Phương pháp LSA

Ma trận document-term X	w_1	w_2	w_3	...	w_N
d_1	1	5	0	...	0
d_2	0	3	2	...	2
...	...	...	...	...	...
d_M	2	0	4	1	2

- LSA (Latent Semantic Analysis) được dùng để khám phá (*explore*) các yếu tố "ẩn" (latent factors) của từ và tài liệu.
- Với *M* tài liệu và *N* từ trong từ điển, LSA xây dựng ma trận document-word $X \in R^{MxN}$ trong đó mỗi dòng biểu diễn một tài liệu và mỗi cột biểu diễn một từ
- Mỗi entry trong ma trận ***X*** có thể là số lần xuất hiện của từ trong tài liệu hay giá trị TF-IDF của từ trong tài liệu.
- LSA phân rã ma trận *X* thành ma trận *term-topic* $U \in R^{Mxt}$ và ma trận *document-topic* $V \in R^{Nxt}$ bằng phương pháp SVD

29

Phương pháp Brown cluster

- Phương pháp biểu diễn phân phối (*distributional representation*) giúp giải quyết các vấn đề của one-hot encoding. Từ giả thuyết phân phối (*distribution hypothesis*) được đề xuất bởi John Firth [117] và Zellig Harris [116],
- Các đối tượng ngôn ngữ "*linguistic objects*" có sự phân phối tương tự nhau (*similar distributions*) và có ngữ nghĩa giống nhau là cơ sở cho việc học biểu diễn ngữ nghĩa của từ.
- Dựa trên giả thuyết phân phối, phương pháp Brown cluster phân lớp các từ vào các cụm phân cấp (*hierarchical clusters*), trong đó các từ trong cùng cụm sẽ có nghĩa giống nhau.

30

Phương pháp Brown cluster

- Phương pháp Brown cluster học để xây dựng một cây nhị phân từ kho ngữ liệu, trong đó, nút lá của cây là các từ và các nút trung gian là các cụm phân cấp từ (*word hierarchical clusters*).
- Trong phương pháp này, mỗi từ chỉ thuộc chính xác vào một cụm.
- Phương pháp biểu diễn từ phân phối (*distributed word representation*) được đề xuất với ý tưởng là nhúng "*embed*" mỗi từ vào trong một vector giá trị thực liên tục (*continuous real-valued vector*), hay còn gọi là *dense vector*.
- Từ "*dense*" nghĩa là một khái niệm (*concept*) được biểu diễn bởi một vector nhiều chiều và mỗi chiều biểu diễn nhiều khái niệm.
- **Các phương pháp biểu diễn từ phân phối có thể kể đến như Word2Vec, Glove, và FastText.**

31

Phương pháp Brown cluster

Cluster #1	Friday	Monday	Thursday	Wednesday	Tuesday	Saturday
Cluster #2	June	March	July	April	January	December
Cluster #3	Water	Gas	Coal	Liquid	Acid	Sand
Cluster #4	Great	Big	Vast	Sudden	Mere	Sheer
Cluster #5	Man	Woman	Boy	Girl	Lawyer	Doctor
Cluster #6	American	Indian	European	Japanese	German	African

32

Nội dung Chương

1. Giới thiệu
2. Ngữ nghĩa của từ và vector ngữ nghĩa
3. Phương pháp biểu diễn từ
4. **Đo lường độ tương đồng giữa các vector**
5. Pointwise Mutual Information
6. Nhúng từ

Độ tương đồng Cosine

- Được tính dựa trên phép toán dot-product

$$dot-product(\vec{v},\vec{w}) = \vec{v}.\vec{w} = \sum_{i=1}^{N} v_i w_i = v_1 w_1 + v_2 w_2 + \cdots + v_N w_N$$

- Giá trị độ tương đồng cosine nằm trong khoảng từ 0 đến 1.
- Hai vector càng tương đồng với nhau thì giá trị cosine giữa chúng càng gần về 1 và ngược lại.
- Trong NLP, độ tương đồng cosine được sử dụng rất phổ biến.

$$cosine(\vec{v},\vec{w}) = \frac{\vec{v}.\vec{w}}{|\vec{v}||\vec{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$

Độ tương đồng Cosine

Hãy cho biết trong các từ "máy_in", "ca_sĩ" và "bài_hát", cặp từ nào gần "giống" nhau nhất?

Từ đồng xuất hiện / Từ đích	chuột	sân_khấu	hòa_nhạc
máy_in	1	0	0
ca_sĩ	0	1	2
bài_hát	1	6	1

Độ tương đồng Cosine

Từ đồng xuất hiện / Từ đích	chuột	sân_khấu	hòa_nhạc
máy_in	1	0	0
ca_sĩ	0	1	2
bài_hát	1	6	1

$$cosine(máy_{in}, bài_{hát}) = \frac{1+0+0}{\sqrt{1+0+0} * \sqrt{1+36+1}} = \frac{1}{\sqrt{38}} = 0{,}16$$

$$cosine(ca_{sĩ}, bài_{hát}) = \frac{0+6+2}{\sqrt{0+1+4} * \sqrt{1+36+1}} = \frac{8}{\sqrt{5}\sqrt{38}} = 0{,}58$$

$$cosine(máy_{in}, ca_{sĩ}) = \frac{0+0+0}{\sqrt{1+0+0} + \sqrt{0+1+4}} = 0$$

Khoảng cách Euclidean

$$|\vec{v} - \vec{w}| = \sqrt{\sum_{i=1}^{N} (v_i - w_i)^2}$$

Thực hiện chuẩn hóa vector, ta được:

$$\left|\frac{\vec{v}}{|\vec{v}|} - \frac{\vec{w}}{|\vec{w}|}\right| = \sqrt{\sum_{i=1}^{N} \left(\frac{v_i}{|\vec{v}|} - \frac{w_i}{|\vec{w}|}\right)^2}$$

Độ tương quan Pearson

Pearson correlation

$$corrPearson(\vec{v}, \vec{w}) = \frac{\sum_{i=1}^{N}(v_i - \bar{v})(w_i - \bar{w})}{|\vec{v} - \bar{v}||\vec{w} - \bar{w}|} = cosine(\vec{v} - \bar{v}, \vec{w} - \bar{w})$$

Trong đó, $\bar{v}$ (tương tự cho $\bar{w}$) là trung bình của các vector thành phần $\bar{v} = \frac{\sum_{i=1}^{N} v_i}{n}$, và

$$\langle v_1, \ldots, v_N \rangle - c = \langle v_1 - c, \ldots, v_N - c \rangle$$

Độ tương đồng Jaccard

$$simJaccard(\vec{v}, \vec{w}) = \frac{\sum_{i=1}^{N} min(v_i, w_i)}{\sum_{i=1}^{N} max(v_i, w_i)}$$

Độ tương đồng Dice

$$simDice(\vec{v}, \vec{w}) = \frac{2\sum_{i=1}^{N} min(v_i, w_i)}{\sum_{i=1}^{N} (v_i + w_i)}$$

Nội dung Chương

1. Giới thiệu
2. Ngữ nghĩa của từ và vector ngữ nghĩa
3. Phương pháp biểu diễn từ
4. Đo lường độ tương đồng giữa các vector
5. **Pointwise Mutual Information**
6. Nhúng từ

Pointwise Mutual Information (PMI)

PMI: Do events *x* and *y* co-occur more than if they were independent?

$$\text{PMI}(X,Y) = \log_2 \frac{P(X,Y)}{P(X)P(Y)}$$

- PMI between two words: (Church & Hanks 1989) Do words *x* and *y* co-occur more than if they were independent?

$$\text{PMI}(word_1, word_2) = \log_2 \frac{P(word_1, word_2)}{P(word_1)P(word_2)}$$

Positive Pointwise Mutual Information (PPMI)

- PMI between two words: $\text{PMI}(word_1, word_2) = \log_2 \frac{P(word_1, word_2)}{P(word_1)P(word_2)}$
  - PMI ranges from $-\infty$ to $+\infty$
  - But the negative values are problematic
    - Things are co-occurring **less than** we expect by chance
    - Unreliable without enormous corpora
    - Plus it's not clear people are good at "unrelatedness"
  - →So, we replace negative PMI values by 0 ➔ Positive PMI (PPMI) between word1 and word2:

$$\text{PPMI}(word_1, word_2) = \max\left(\log_2 \frac{P(word_1, word_2)}{P(word_1)P(word_2)}, 0\right)$$

43

Count(w,context)	computer	**data**	pinch	results	sugar	Count(w)
apricot	0	0	1	0	1	2
pineapple	0	0	1	0	1	2
digital	2	1	0	1	0	4
information	1	**6**	0	4	0	11
Count(context)	3	7	2	5	2	19

P(w,context)	computer	**data**	pinch	results	sugar	P(w)
apricot	0.00	0.00	0.05	0.00	0.05	0.11
pineapple	0.00	0.00	0.05	0.00	0.05	0.11
digital	0.11	0.05	0.00	0.05	0.00	0.21
information	0.05	**0.32**	0.00	0.21	0.00	0.58
P(context)	0.16	0.37	0.11	0.26	0.11	

$$P(w = information, c = data) = \frac{6}{19} = 0.32$$
$$P(w = information) = \frac{11}{19} = .58$$
$$P(c = data) = \frac{7}{19} = 0.37$$

$$\text{PPMI}(word_1, word_2) = \max\left(\log_2 \frac{P(word_1, word_2)}{P(word_1)P(word_2)}, 0\right)$$

$$PPMI(information, data) = \max\left(log_2 \frac{0.32}{0.58 * 0.37}, 0\right) = 0.57$$

PPMI	computer	**data**	pinch	results	sugar
apricot	-	-	2.25	-	2.25
pineapple	-	-	2.25	-	2.25
digital	1.66	0.00	-	0.00	-
information	0.00	**0.57**	-	0.47	-

44

Weighting PMI

$$\text{PPMI}_\alpha(w,c) = \max\left(\log_2 \frac{P(w,c)}{P(w)P_\alpha(c)}, 0\right)$$

PMI is biased toward infrequent events: very rare words have very high PMI values

Two solutions:

- Give rare words slightly higher probabilities
- Use add-one smoothing (which has a similar effect)

$$P_\alpha(c) = \frac{count(c)^\alpha}{\sum_c count(c)^\alpha}$$

$$P_\alpha(a) = \frac{.99^{.75}}{.99^{.75} + .01^{.75}} = 0.97$$

Weighting PMI: Giving rare context words slightly higher probability

- Raise the context probabilities to $\alpha = 0.75$:
- This helps because $P_\alpha(c) > P(c)$ for rare c
- Consider two events, P(a) = 0.99 and P(b)=0.01

$$P_\alpha(b) = \frac{.01^{.75}}{.01^{.75} + .01^{.75}} = 0.03$$

Use add-k smoothing

Count(w,context)	computer	data	pinch	results	sugar	Count(w)
apricot	0	0	1	0	1	2
pineapple	0	0	1	0	1	2
digital	2	1	0	1	0	4
information	1	6	0	4	0	11
Count(context)	3	7	2	5	2	19

	Add-2 smoothed					
Count(w,context)	computer	data	pinch	results	sugar	Count(w)
Apricot	2	2	3	2	3	12
Pineapple	2	2	3	2	3	12
Digital	4	3	2	3	2	14
Information	3	8	2	6	2	21
Count(context)	11	15	10	13	10	59

	Add-2 smoothed					
P(w,context)	computer	data	pinch	results	sugar	P(w)
Apricot	0.03	0.03	0.05	0.03	0.05	0.20
Pineapple	0.03	0.03	0.05	0.03	0.05	0.20
Digital	0.07	0.05	0.03	0.05	0.03	0.24
Information	0.05	0.14	0.03	0.10	0.03	0.36
P(context)	0.19	0.25	0.17	0.22	0.17	

PPMI	computer	data	pinch	results	sugar
apricot	-	-	2.25	-	2.25
pineapple	-	-	2.25	-	2.25
digital	1.66	0.00	-	0.00	-
information	0.00	0.57	-	0.47	-

Compare

Add-2 smoothed PPMI	computer	data	pinch	results	sugar
Apricot	0.00	0.00	0.56	0.00	0.56
Pineapple	0.00	0.00	0.56	0.00	0.56
Digital	0.62	0.00	0.00	0.00	0.00
Information	0.00	0.58	0.00	0.37	0.00

Exercise

Consider the following word-context matrix with the three words “orange”, “banana” and “car” and the three context words “juice”, “the” and “drive”.

	juice	**the**	**drive**
orange	10	20	0
banana	8	20	0
car	1	20	10

$$\text{PPMI}(word_1, word_2) = \max\left(\log_2 \frac{P(word_1, word_2)}{P(word_1)P(word_2)}, 0\right)$$

1. Compute the MLEs using frequencies for the probabilities *P(w)*, *P(c)* and *P(w,c)* for each word *w* and each context word *c*.
2. Based on these, compute the PPMI values for the cells in the matrix.
3. Now compute the cosine similarity values of the PPMI vectors for “orange” and “banana”, and for “orange” and “car”.

Solution ?

47

Exercise -solution

	juice	the	drive
orange	10	20	0
banana	8	20	0
car	1	20	10

1. Compute the MLEs using frequencies for the probabilities P(w), P(c) and P(w, c) for each word *w* and each context word *c*.
2. Based on these, compute the PPMI values for the cells in the matrix
3. Now compute the cosine similarity values of the PPMI vectors for “orange” and “banana” and for “orange” and “car”.

	juice	the	drive	P(w)
orange	10/89	20/89	0	30/89
banana	8/89	20/89	0	28/89
car	1/89	20/89	10/89	31/89
P(context)	19/89	60/89	10/89	

PPMI	juice	the	drive
orange	0.64	0	0
banana	0.42	0.08	0
car	0	0	1.52

$$\text{PPMI}(word_1, word_2) = \max\left(\log_2 \frac{P(word_1, word_2)}{P(word_1)P(word_2)}, 0\right)$$

$$CosSim(\vec{v}, \vec{w}) = \frac{\vec{v}\cdot\vec{w}}{|\vec{v}|\,|\vec{w}|} = \frac{\sum_{i=1}^{n} v_i w_i}{\sqrt{\sum_{i=1}^{n} v_i^2}\sqrt{\sum_{i=1}^{n} w_i^2}}$$

orange, banana: $\frac{0.64 \cdot 0.42}{0.64 \cdot \sqrt{0.42^2 + 0.0833^2}} = 0.98$

orange, car: 0

48

Exercise

Below are distributional vectors of co-occurrence counts for three words of English: “king”, “woman” and “queen” (in a shuffled order)

Which of the three vectors corresponds to each of the three words (“king”, “woman” and “queen”). How did you tell?

	Context 1	Context 2	Context 3	Context 4	Context 5
Word 1	10	82	0	4	275
Word 2	85	5	4	0	8
Word 3	237	20	4	1	9

Solution ?

49

Exercise - solution

Below are distributional vectors of co-occurrence counts for three words of English: “king”, “woman” and “queen” (in a shuffled order)

Which of the three vectors corresponds to each of the three words (“king”, “woman” and “queen”). How did you tell?

	Context 1	Context 2	Context 3	Context 4	Context 5
Word 1	10	82	0	4	275
Word 2	85	5	4	0	8
Word 3	237	20	4	1	9

Solution: cosine

- Word1 vs. word2: 0.141
- Word1 vs. word3: 0.095
- Word2 vs. word3: 0.998

➔Words2 and Word3 are similar, Word1 differs from them. One can conclude that Word1 is *woman*. We also note that Word1 is closer to Word2 (which must be *queen*) than to Word1➔ Word3 is *king*

50

Exercise

The contexts in this example are words *knighthood*, *delivery*, *says*, *child* and *crowned* (in a shuffled order), as observed within a 5-word window around the target word. Can you identify which of these context words correspond to each column in the co-occurrence matrix above? Even if you are not sure about the complete answer, make an educated guess for some of the contexts and argue for it.

		Context 1	Context 2	Context 3	Context 4	Context 5
Woman	Word 1	10	82	0	4	275
Queen	Word 2	85	5	4	0	8
King	Word 3	237	20	4	1	9

Solution:

Context1 and Context3 are associated with royalty, context1 being more frequent.

→So Context1=*crowned*, Context 3=*knighthood*

Context5 is frequent and associated with feminine gender → *child*

Of the remaining, Context2 is frequent → *says*, so Context4 → *delivery*

51

Nội dung Chương

1. Giới thiệu
2. Ngữ nghĩa của từ và vector ngữ nghĩa
3. Phương pháp biểu diễn từ
4. Đo lường độ tương đồng giữa các vector
5. Pointwise Mutual Information
6. **Nhúng từ**

52

What does "*ongchoi*" mean?

Suppose you see these sentences:

- *Ongchoi* is delicious sautéed with **garlic**.
- *Ongchoi* is superb over **rice**
- *…ongchoi* **leaves** with salty sauces

And you've also seen these:

- …**spinach** sautéed with **garlic** over **rice**
- Chard stems and **leaves** are **delicious**
- Collard greens and other **salty** leafy greens

→ Conclusion:

- *Ongchoi* is a leafy green like  ,  or collard greens

Ong choy: *Ipomoea Aquatica "Water Spinach"* Yamaguchi, Wikimedia Commons, public domain

LNK 53

We'll build a new model of meaning focusing on similarity

Each word = a vector

Similar words are "nearby in semantic space"

Called an "embedding" because it's embedded into a space

Every modern NLP algorithm uses embeddings as the representation of word meaning

A two-dimensional projection of embeddings for some words and phrases, showing that words with similar meanings are nearby in space. The original 60- dimensional embeddings were trained for sentiment analysis. Simplified from (Li et al.,2015).1

LNK 54

Word vector or embedding

Called an "embedding" because it's embedded into a space

The standard way to represent meaning in NLP: consider sentiment analysis
- With words, a feature is a word identity
  - Requires exact same word to be in training and test
- With embeddings, feature is a word vector.
  - The previous word was vector [35,22,17…], now in the test set we might see a similar vector [34,21,14]
  - We can generalize to similar but unseen words!!!
  - Ok if similar words occurred!!!

Every modern NLP algorithm uses embeddings as the representation of word meaning

Example: discuss 2 kinds of embeddings
- Tf-idf
  - A common baseline model
  - *Sparse* vectors
  - Words are represented by a simple function of the *counts* of nearby words
- Word2vec
  - *Dense* vectors
  - Representation is created by training a classifier to *predict* whether a word is likely to appear nearby

Term-document matrix

- The *term-document* matrix for four words in four Shakespeare plays.
- Each cell contains the number of times the (row) word occurs in the (column) document.
- Each document is represented by a vector of words

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

- The term-document matrix for four words in four Shakespeare plays.
- The red boxes show that each document is represented as a column vector of length four.

Reminders

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

A vector is just a list or array of numbers. So,

- "As You Like It" is represented as the list [1,114,36,20]
- "Julius Caesar" is represented as the list [7,62,1,2].

A vector space is a collection of vectors, dimension characterized by their dimension.

The vectors representing each document would have dimensionality |V|, the vocabulary size

The ordering of the numbers in a vector space is not arbitrary

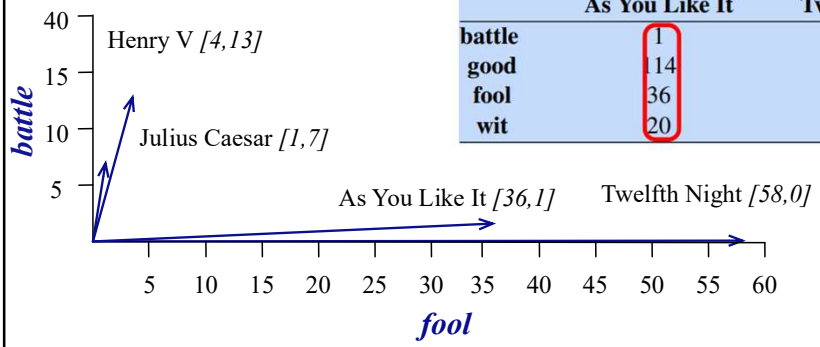
- Each position indicates a meaningful dimension on which the documents can vary.

→The 1st dimension for both these vectors corresponds to the number of times the word "battle" occurs, and we can compare each dimension,

- Vectors for "As You Like It" and "Twelfth Night" have similar values for the first dimension

LNK 57

Visualizing document vectors



	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

Vectors are the basis of information retrieval

- Vectors are similar for the two comedies
- Different than the history
- Comedies have more *fools* and *wit* and fewer *battles*.

LNK 58

Words as vectors

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

battle is "the kind of word that occurs in Julius Caesar and Henry V"

fool is "the kind of word that occurs in comedies, especially Twelfth Night"

2 words are similar in meaning if their context vectors are similar

is traditionally followed by **cherry** pie, a traditional dessert
often mixed, such as **strawberry** rhubarb pie. Apple pie
computer peripherals and personal **digital** assistants. These devices usually
a computer. This includes **information** available on the internet

	aardvark	...	computer	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	...
strawberry	0	...	0	0	1	60	19	...
digital	0	...	1670	1683	85	5	4	...
information	0	...	3325	3982	378	5	13	...

Discussion

TF-IDF and PPMI are sparse representations. TF-IDF and PPMI vectors are **long** (length |V|= 20,000 to 50,000) and **sparse** (most elements are zero)

Sparse long vectors ignore synonymy:

- *car* and *automobile* are synonyms; but are represented as distinct coordinates; this fails to capture similarity between a word with *car* as a neighbor and a word with *automobile* as a neighbor

Alternative: dense vectors → Vectors which are **short** (length 50-1000) and **dense** (most elements are non-zero). Why dense vectors?

- Short vectors may be easier to use as **features** in machine learning (less weights to tune)
- Dense vectors may **generalize** better than storing explicit counts
- They may do better at capturing synonymy:
  - *car* and *automobile* are synonyms; but are distinct dimensions
    - a word with *car* as a neighbor and a word with *automobile* as a neighbor should be similar, but aren't
- **In practice, they work better**

From sparse vectors to dense embeddings

So far, our vectors are high-dimensional and sparse.

Singular Value Decomposition (SVD) is a classic method for generating dense vectors.

Idea:

- Change the dimensions such that they are still orthogonal to each other.
- The new dimensions are such that the first describes the largest amount of variance in the data, the second the second large variance amount etc.
- Then, instead of keeping all the *m* dimensions resulting from this, we only keep the first k.
- We obtain context vectors of dimension *k* for each word →These are dense embeddings

61

From sparse vectors to dense embeddings

Assume that we have *w* words and *c* context words. Then, in general, SVD decomposes the *w* × *c* word-context matrix *X* into a product of three matrices W, Σ, C:

$$\begin{bmatrix} \\ X \\ \\ \\ \\ \end{bmatrix} = \begin{bmatrix} \\ W \\ \\ \\ \\ \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & 0 & \dots & 0 \\ 0 & 0 & \sigma_3 & \dots & 0 \\ & & \dots & & \\ 0 & 0 & 0 & \dots & \sigma_m \end{bmatrix} \begin{bmatrix} \\ C \\ \\ \\ \end{bmatrix}$$

$w \times c$ $\quad$ $w \times m$ $\quad$ $m \times m$ $\quad$ $m \times c$

Each row in *X* is a PPMI context word vector of a word. Each row in *W* is a word embedding of a word in a new *m*-dimensional vector space.

62

From sparse vectors to dense embeddings

Matrix *W*: The first dimension (i.e., vector ⟨1, 0⟩) corresponds to ⟨1, 2⟩ in original matrix X;

Matrix *Σ*: multiply length 1 with the length of ⟨1, 2⟩;

Matrix *C*: rotation from *x-axis* to the axis along ⟨1, 2⟩;

Last step: truncation the second dimension in *W* can be left out, which leads to [1 2] instead of the original [1 2 2 4], giving 1- dimensional embedding vectors for each word

$$\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} \sqrt{5} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \\ -\frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \end{bmatrix}$$

63

Embeddings

Word2vec https://code.google.com/archive/p/word2vec/

Fasttext http://www.fasttext.cc/

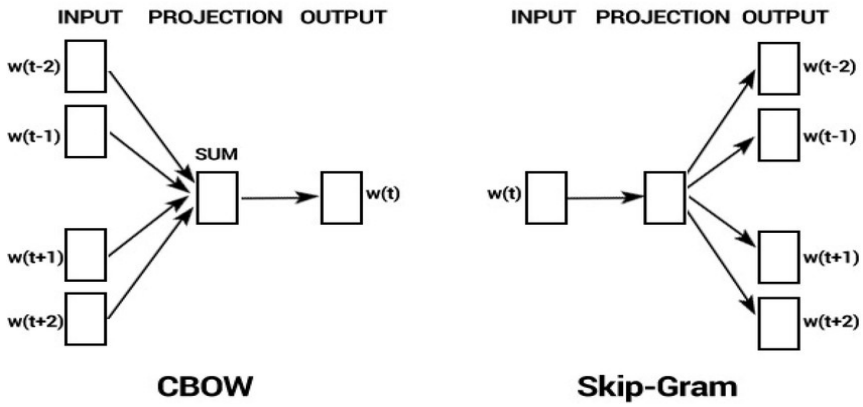
Glove http://nlp.stanford.edu/projects/glove/

WordPiece

Byte Pair Encoding (BPE)

64

Word2vec



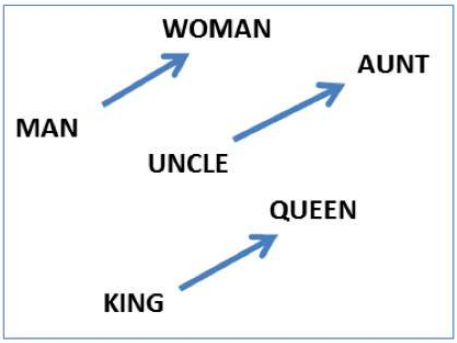
Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781. 2013 Jan 16.

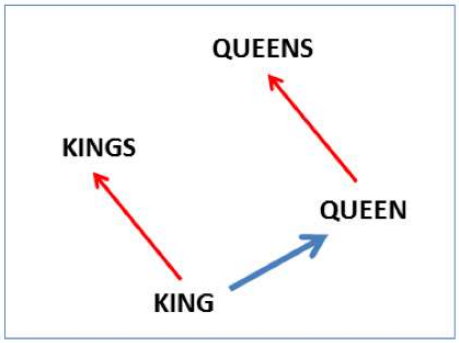
65

Analogy: Embeddings capture relational meaning!

vector(*'king'*) - vector(*'man'*) + vector(*'woman'*) ≈ vector('queen')

vector(*'Paris'*) - vector(*'France'*) + vector(*'Italy'*) ≈ vector('Rome')





66

Word2Vec - CBOW		Word2Vec - Skipgram		Glove	
“Giọng hát”					
Từ	Sim	Từ	Sim	Từ	Sim
chinh phục	0,881	bolero	0,827	vđv	0,992
bolero	0,864	nội lực	0,813	nhí	0,992
nhảy	0,859	nhí	0,813	tìm kiếm	0,992
nhí	0,859	truyền cảm	0,809	truyền hình	0,989
manga	0,853	thính phòng	0,799	quyết	0,987
nhạc phẩm	0,853	nhạc phẩm	0,787	Carolina	0,981
“Con cái”					
suy nghĩ	0,955	chăm lo	0,96	khỏe	0,988
đuổi kịp	0,946	nghèo khó	0,899	cân bằng	0,982
tan vỡ	0,943	bất hạnh	0,887	tình dục	0,982
sinh hoạt	0,943	thấu hiểu	0,881	nghĩa là	0,981
“Tình yêu”					
tình cảm	0,954	tình bạn	0,816	thân thiết	0,989
cuộc đời	0,947	tình cảm	0,792	cảm xúc	0,988
yêu	0,946	ký ức	0,971	cuộc đời	0,988
hạnh phúc	0,928	tuổi thơ	0,788	tình cảm	0,983
cảm xúc	0,921	cảm động	0,784	ngoại giao	0,983
câu chuyện	0,921	bình yên	0,784	chàng	0,981
vui	0,914	mối tình	0,783	hạnh phúc	0,981

Word embedding cho tập dữ liệu tiếng Việt

Khang Nhut Lam, Tuan Huynh To, Thong Tri Tran, and Jugal Kalita. 2019. Improving Vietnamese WordNet using word embedding. In Proceedings of the 2019 3rd International Conference on Natural Language Processing and Information Retrieval (NLPIR 2019). Association for Computing Machinery, New York, NY, USA, 110–114. DOI:https://doi.org/10.1145/3342827.3342854