《语义计算与知识检索》研究生课程

词汇语义计算（二）

万小军

北京大学语言计算与互联网挖掘组

2018年3月14日

http://www.icst.pku.edu.cn/lcwm/course/sckr2018/

内容

- 词汇语义计算概述
- 基于语义词典的词汇语义计算
- 基于语料统计的词汇语义计算

词汇语义的向量表示

- **独热向量表示(one-hot vector)**

  university → **<0,0,...,1,...,0,0>**

- **稀疏向量表示(sparse vector)**

  university → **<0,0.3,...,0,...,0,1>**

- **密集向量表示(dense vector)**

  university → **<0.1,0.2,...,0.01,...,1,0.5>**

基于上述词汇的向量表示很容易计算词汇之间的相似度或语义距离。

词汇向量表示的获取

- 可利用的资源
  - 语料库
  - 互联网网页（搜索引擎）
  - 维基百科
- 可利用的线索
  - 词汇共现关系

基于上下文向量的词汇相似度计算

- A bottle of *tezgüino* is on the table
- Everybody likes *tezgüino*
- *Tezgüino* makes you drunk
- We make *tezgüino* out of corn.

- Intuition:
  - 人们从词的上下文中就能推测出*tezgüino*的意义
  - 基本思想: 如果两个词语具有相似的上下文，那么这两个词语相似

上下文向量(Context vector)

- 目标词为w
- 对于词表（包含N个词）中每个词v_i都对应一个二值特征f_i
  - 表示词 v_i 是否在w的附近出现
- $w=(f_1, f_2, f_3,\ldots, f_N)$
- If w= tezgüino, v_1 = bottle, v_2 = drunk, v_3 = matrix:
- w = (1,1,0,…)

Intuition

- 用稀疏特征向量定义词语
- 基于向量距离/相似度公式进行计算
- 两个词语向量相似，那么这两个词相似

	arts	boil	data	function	large	sugar	summarized	water
apricot	0	1	0	0	1	1	0	1
pineapple	0	1	0	0	1	1	0	1
digital	0	0	1	1	1	0	1	0
information	0	0	1	1	1	0	1	0

分布式相似度(Distributional similarity)

- 三个问题
  1. 如何定义“共同出现”？
  2. 词语权重如何度量?
     - (frequency? Logs? Mutual information?)
  3. 如何选择向量距离/相似度计算公式?
     - Cosine? Euclidean distance?

上下文(共现)向量定义

- 基于窗口
- 基于句法结构
  - 进行句法分析，抽取依存关系

I discovered dried tangerines:

discover (subject I)	I (subj-of discover)
tangerine (obj-of discover)	tangerine (adj-mod dried)
dried (adj-mod-of tangerine)	

基于依存关系的共现向量

	subj-of, absorb	subj-of, adapt	subj-of, behave	...	pobj-of, inside	pobj-of, into	...	nmod-of, abnormality	nmod-of, anemia	nmod-of, architecture	...	obj-of, attack	obj-of, call	obj-of, come from	obj-of, decorate	...	nmod, bacteria	nmod, body	nmod, bone marrow
cell	1	1	1		16	30		3	8	1		6	11	3	2		3	2	2

向量中词语特征权重计算

- 频率及其变形
- 考虑如下特征
  - f= (obj-of, *attack*)
  - P(f|w)=count(f, w) / count(w)
  - $Assoc_{prob}(w, f)=p(f|w)$

词语权重计算之互信息(Mutual Information)

- **Pointwise mutual information:** measure of how often two events x and y occur, compared with what we would expect if they were independent:

$$I(x,y) = \log_2 \frac{P(x,y)}{P(x)P(y)}$$

- **PMI between a target word *w* and a feature *f*:**

$$\text{assoc}_{\text{PMI}}(w,f) = \log_2 \frac{P(w,f)}{P(w)P(f)}$$

互信息举例

- **Objects of the verb *drink***

Object	Count	PMI assoc	Object	Count	PMI assoc
bunch beer	2	12.34	wine	2	9.34
tea	2	11.75	water	7	7.65
Pepsi	2	11.75	anything	3	5.15
champagne	4	11.75	much	3	5.15
liquid	2	10.53	it	3	1.25
beer	5	10.20	<SOME AMOUNT>	2	1.22

相似度计算公式

$$\mathrm{sim}_{\mathrm{cosine}}(\vec{v},\vec{w}) = \frac{\vec{v}\cdot\vec{w}}{|\vec{v}||\vec{w}|} = \frac{\sum_{i=1}^{N} v_i \times w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$

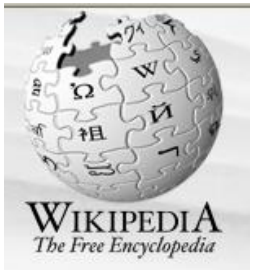
$$\mathrm{sim}_{\mathrm{Jaccard}}(\vec{v},\vec{w}) = \frac{\sum_{i=1}^{N} \min(v_i,w_i)}{\sum_{i=1}^{N} \max(v_i,w_i)}$$

$$\mathrm{sim}_{\mathrm{Dice}}(\vec{v},\vec{w}) = \frac{2\times\sum_{i=1}^{N} \min(v_i,w_i)}{\sum_{i=1}^{N} (v_i+w_i)}$$

$$\mathrm{sim}_{\mathrm{JS}}(\vec{v}||\vec{w}) = D\left(\vec{v}\middle|\frac{\vec{v}+\vec{w}}{2}\right) + D\left(\vec{w}\middle|\frac{\vec{v}+\vec{w}}{2}\right)$$

$$D(p_1(V)\,\|\,p_2(V)) = \sum_{v} p_1(v)\log\frac{p_1(v)}{p_2(v)}.$$

维基百科(Wikipedia)



- Wikipedia 用一篇文章描述一个概念
  - > 5,352,149 articles in English
  - 中文维基则有930,451篇条目=》可考虑百度/互动百科
- 每篇文章属于至少一个类别
  - 类别以层次式组织
- 文章之间的超链接反映了它们之间的语义关系
  - Equivalence (synonymy), hierarchical (hyponymy), association
    - Redirect hyperlinks , Disambiguation page , Association hyperlinks

Cougar

From Wikipedia, the free encyclopedia

(Redirected from Puma (animal))

Redirect hyperlinks

Disambiguation hyperlinks

This article is about the large cat species. For other uses, see Cougar (disambiguation).

"Catamount" redirects here. For the ski area, see Catamount Ski Area.

Associations hyperlinks

The **cougar** (*Puma concolor*), also **puma**, **mountain lion**, or **panther**, depending on region, is a mammal of the Felidae family, native to the Americas. This large, solitary cat has the greatest range of any wild terrestrial mammal in the Western Hemisphere,[3] extending from Yukon in Canada to the southern Andes of South America. An adaptable, generalist species, the cougar is found in every major American habitat type. It is the second heaviest cat in the American continents after the jaguar, and the fourth heaviest in the world, along with the leopard, after the tiger, lion, and jaguar, although it is most closely related to smaller felines.

A capable stalk-and-ambush predator, the cougar pursues a wide variety of prey. Primary food sources include ungulates such as deer, elk, and bighorn sheep, as well as domestic cattle, horses, and sheep, particularly in the northern part of its range, but it also hunts species as small as insects and rodents. Moreover, it prefers habitats with dense underbrush and rocky areas for stalking, but it can live in open areas. The cougar is territorial and persists at low population densities. Individual territory sizes depend on terrain, vegetation, and abundance of prey. While it is a large predator, it is not always the dominant species in its range, as when it competes for prey with other predators such as the jaguar, gray wolf, black bear, and the grizzly bear. It is a reclusive cat and usually avoids people. Attacks on humans remain rare, despite a recent increase in frequency.[4]

Due to persecution following the European colonization of the Americas, and continuing human development of cougar habitat, populations have dropped in many parts of its historical range. In particular, the cougar was extirpated in eastern North America, except an isolated sub-population in Florida; the animal may be recolonizing parts of its former eastern territory. With its vast range, the cougar has dozens of names and various references in the mythology of the indigenous Americans and in contemporary culture.

维基百科(Wikipedia)

- The Redirect hyperlink
  - 归并相同的概念
    - E.g. The synonyms between “puma/美洲狮” and “cougar ['kugər] ”
  - 除了同义词之外，还可以处理拼写变形，简写等
    - E.g. “United States” : acronyms (U.S.A , U.s., USA, US) ,Spanish translations (Los Estados, Unidos), common misspellings (Untied States) , synonyms (Yankee Land)

维基百科(Wikipedia)

- **The disambiguation page**
  - **列举相应概念所有可能的意义**
    - **E.g. the term “puma” lists 22 associated concepts, including animals , cars, and a sports brand**

Puma

From Wikipedia, the free encyclopedia
(Redirected from Puma (disambiguation))

Puma may refer to:

Animals

- Puma (animal), a large cat commonly called a cougar
- *Puma* (genus), the genus containing the cougar and the jaguarundi

Vehicles

Sports cars

- Puma (car), a Brazilian brand of sports cars
- Ford Puma, a sports car

Combat vehicles

- Puma (AFV), an Italian family of armoured fighting vehicles
- Puma (IFV), a German infantry fighting vehicle
- IDF Puma, an Israeli combat engineering vehicle
- Sd.Kfz. 234/2, a German armored car

Aerial

- Aérospatiale Puma, a helicopter
- Eurocopter Super Puma, an enlarged version of the Aérospatiale Puma
- Puma, an unmanned aerial vehicle produced by AeroVironment

Technology

- AMD Puma, a mobile computing platform
- Programmable Universal Machine for Assembly, an industrial robot arm
- Mac OS X v10.1, an operating system codenamed "Puma"

Sports

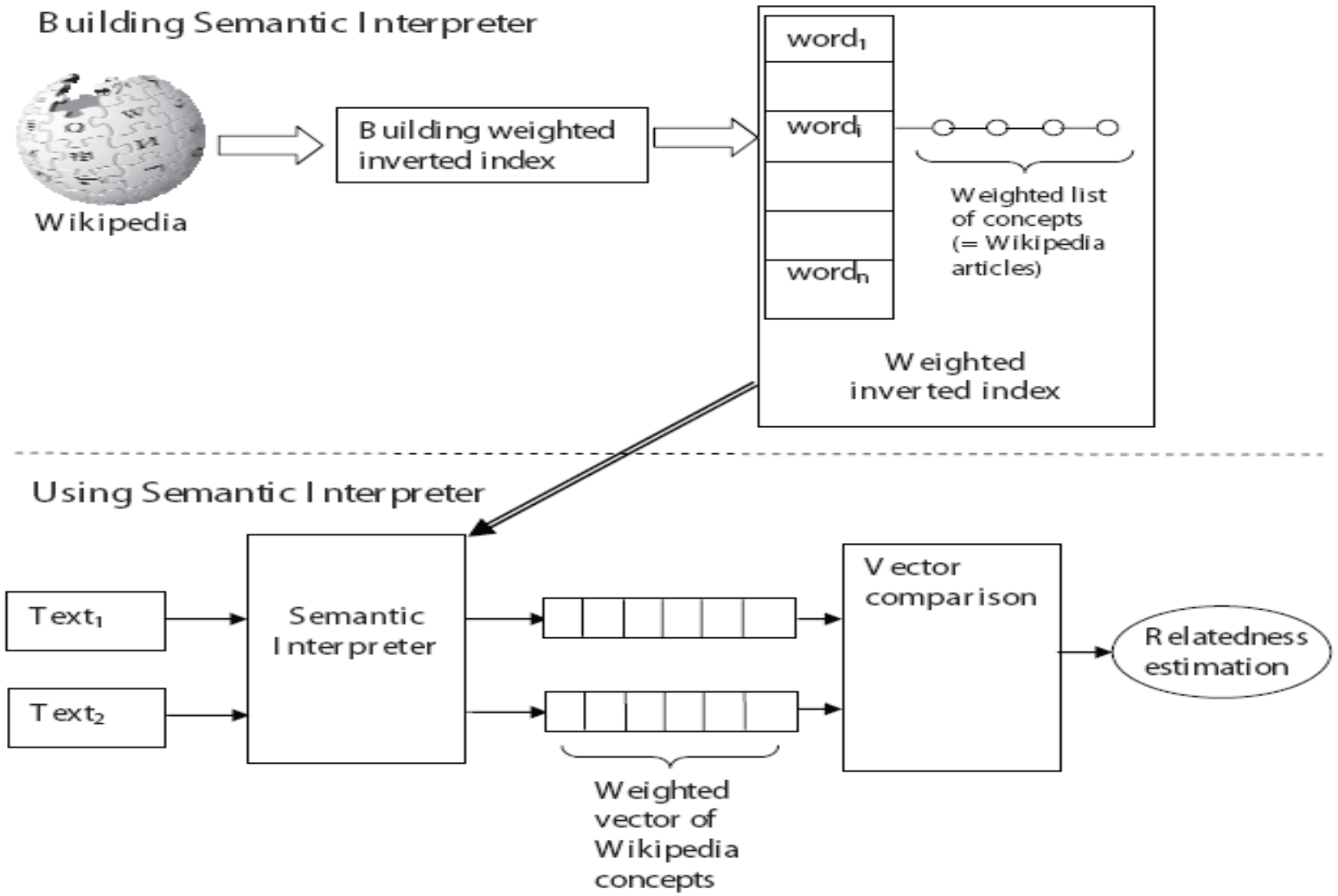
- Pumas de la UNAM, a Mexican professional football club
- Pumas (rugby team), a South African rugby team
- Argentina national rugby union team, nicknamed Los Pumas

基于Wikipedia的语义计算

- 两类方法
  - 将Wikipedia看作语义“词典”，可提取出词语之间的多种关系；
    - Wiktionary: a free lexical database in every language
      - > 5,081,299 entries with English definitions from over 3,150 languages
  - 将Wikipedia作为语料，进行统计分析
    - ESA - Explicit Semantic Analysis

[PDF] Computing semantic relatedness using wikipedia-based explicit semantic analysis.
E Gabrilovich, S Markovitch - IJcAI, 2007 - aaai.org
Abstract Computing semantic relatedness of natural language texts requires access to vast amounts of common-sense and domain-specific world knowledge. We propose Explicit Semantic Analysis (ESA), a novel method that represents the meaning of texts in a high-dimensional space of concepts derived from Wikipedia. We use machine learning techniques to explicitly represent the meaning of any text as a weighted vector of ...
被引用次数：1829 相关文章 所有 28 个版本 引用 保存 更多

Wikipedia-based Explicit Semantic Analysis



Wikipedia-based Explicit Semantic Analysis

- 根据词语在维基概念对应文章中是否出现，每个词由一个维基概念向量表示.

#	Input: *"equipment"*	Input: *"investor"*
1	Tool	Investment
2	Digital Equipment Corporation	Angel investor
3	Military technology and equipment	Stock trader
4	Camping	Mutual fund
5	Engineering vehicle	Margin (finance)
6	Weapon	Modern portfolio theory
7	Original equipment manufacturer	Equity investment
8	French Army	Exchange-traded fund
9	Electronic test equipment	Hedge fund
10	Distance Measuring Equipment	Ponzi scheme

Wikipedia-based Explicit Semantic Analysis

WordSimilarity-353

Algorithm	Correlation with humans
WordNet [Jarmasz, 2003]	0.33–0.35
Roget's Thesaurus [Jarmasz, 2003]	0.55
LSA [Finkelstein *et al.*, 2002]	0.56
WikiRelate! [Strube and Ponzetto, 2006]	0.19 – 0.48
ESA-Wikipedia	0.75
ESA-ODP	0.65

同样基于维基百科，但利用的是概念的类别层次结构进行计算

ESA可利用ODP等其他网站进行计算

Table 4: Computing word relatedness

1,225 pairs of documents from the Australian Broadcasting Corporation's news mail service

Algorithm	Correlation with humans
Bag of words [Lee *et al.*, 2005]	0.1–0.5
LSA [Lee *et al.*, 2005]	0.60
ESA-Wikipedia	0.72
ESA-ODP	0.69

Table 5: Computing text relatedness

Latent Semantic Analysis

- 隐含语义分析
  - 不依赖显示语义概念，自动挖掘隐含的语义概念
  - 但人类通常无法理解隐含语义概念，可供计算

Indexing by latent semantic analysis
S Deerwester, ST Dumais, GW Furnas… - Journal of the …, 1990 - search.proquest.com
Indexing by Latent Semantic Analysis Scott Deerwester Center for Information and Language Studies, University of Chicago, Chicago, IL 60637 Susan T. Dumais*, George W. Furnas, and Thomas K. Landauer Bell Communications Research, 445 South St.,
被引用次数：11978　相关文章　所有 87 个版本　引用　保存

隐含语义分析（Latent Semantic Analysis）

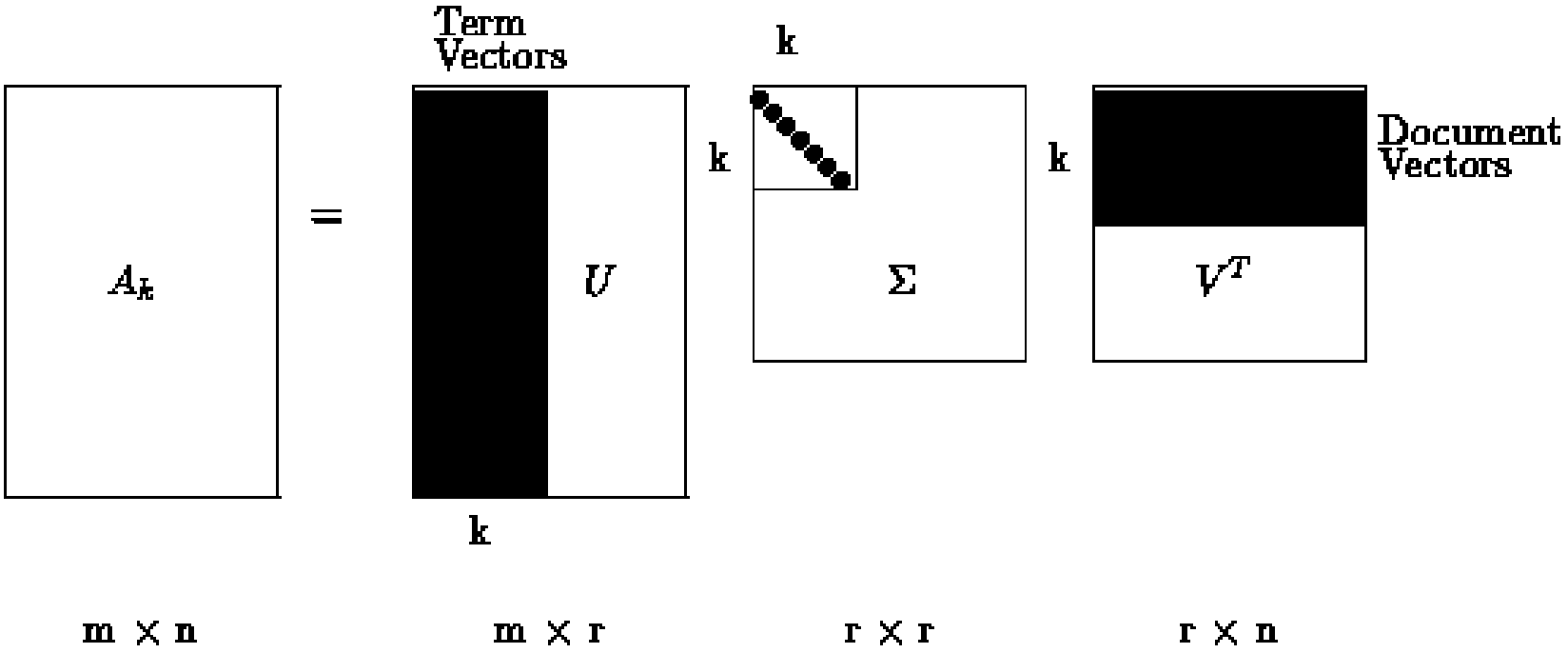
- 对词语-文档矩阵进行低秩逼近
- 词语与文档能够被隐含语义空间中的向量表示

Sample Term by Document matrix

	access	*document*	*retrieval*	*information*	*theory*	*database*	*indexing*	*computer*	REL	MATCH
Doc 1	x	x	x			x	x		R	
Doc 2				x*	x			x*		M
Doc 3			x	x*				x*	R	M

Query: "IDF in *computer*-based *information* look-up"

Latent Semantic Analysis



Latent Semantic Analysis

- 开始于Term-by-Document matrix (A)
- 应用Singular Value Decomposition (SVD):
  - m = # of terms
  - n= # of documents
  - r ≤ min(m, n)

正交矩阵　　对角矩阵　　正交矩阵

$$A_{m\times n} = U_{m\times r} \times \sum_{r\times r} \times (V_{n\times r})^T$$

- Approximate using k << r (semantic) dimensions:

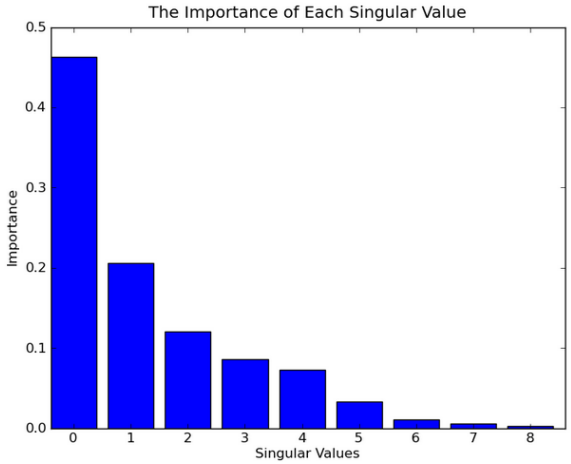
$$\hat{A}_{m\times n} = U_{m\times k} \times \sum_{k\times k} \times (V_{n\times k})^T$$

Latent Semantic Analysis

- 矩阵U中的每一行表示相应词语与隐含语义空间中语义维度之间的关联，Likewise for V
- 文档相似度: vector comparison in $\sum V^T$
- 词语相似度: vector comparison in $U\sum$

Latent Semantic Analysis

- 举例：



	T1	T2	T3	T4	T5	T6	T7	T8	T9
book			1	1					
dads						1			1
dummies		1						1	
estate							1		1
guide	1					1			
investing	1	1	1	1	1	1	1	1	1
market	1		1						
real							1		1
rich						2			1
stock	1		1					1	
value				1	1				

book	0.15	-0.27	0.04
dads	0.24	0.38	-0.09
dummies	0.13	-0.17	0.07
estate	0.18	0.19	0.45
guide	0.22	0.09	-0.46
investing	0.74	-0.21	0.21
market	0.18	-0.30	-0.28
real	0.18	0.19	0.45
rich	0.36	0.59	-0.34
stock	0.25	-0.42	-0.28
value	0.12	-0.14	0.23

*

3.91	0	0
0	2.61	0
0	0	2.00

*

T1	T2	T3	T4	T5	T6	T7	T8	T9
0.35	0.22	0.34	0.26	0.22	0.49	0.28	0.29	0.44
-0.32	-0.15	-0.46	-0.24	-0.14	0.55	0.07	-0.31	0.44
-0.41	0.14	-0.16	0.25	0.22	-0.51	0.55	0.00	0.34

Latent Semantic Analysis

book	0.15	-0.27	0.04
dads	0.24	0.38	-0.09
dummies	0.13	-0.17	0.07
estate	0.18	0.19	0.45
guide	0.22	0.09	-0.46
investing	0.74	-0.21	0.21
market	0.18	-0.30	-0.28
real	0.18	0.19	0.45
rich	0.36	0.59	-0.34
stock	0.25	-0.42	-0.28
value	0.12	-0.14	0.23

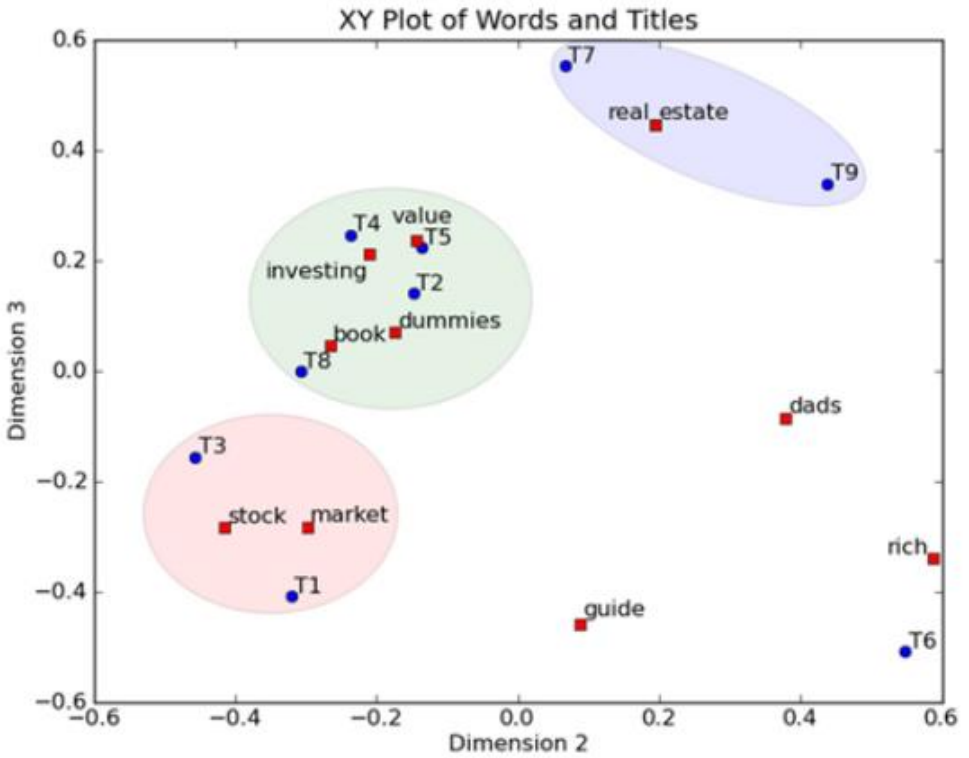
*

3.91	0	0
0	2.61	0
0	0	2.00

*

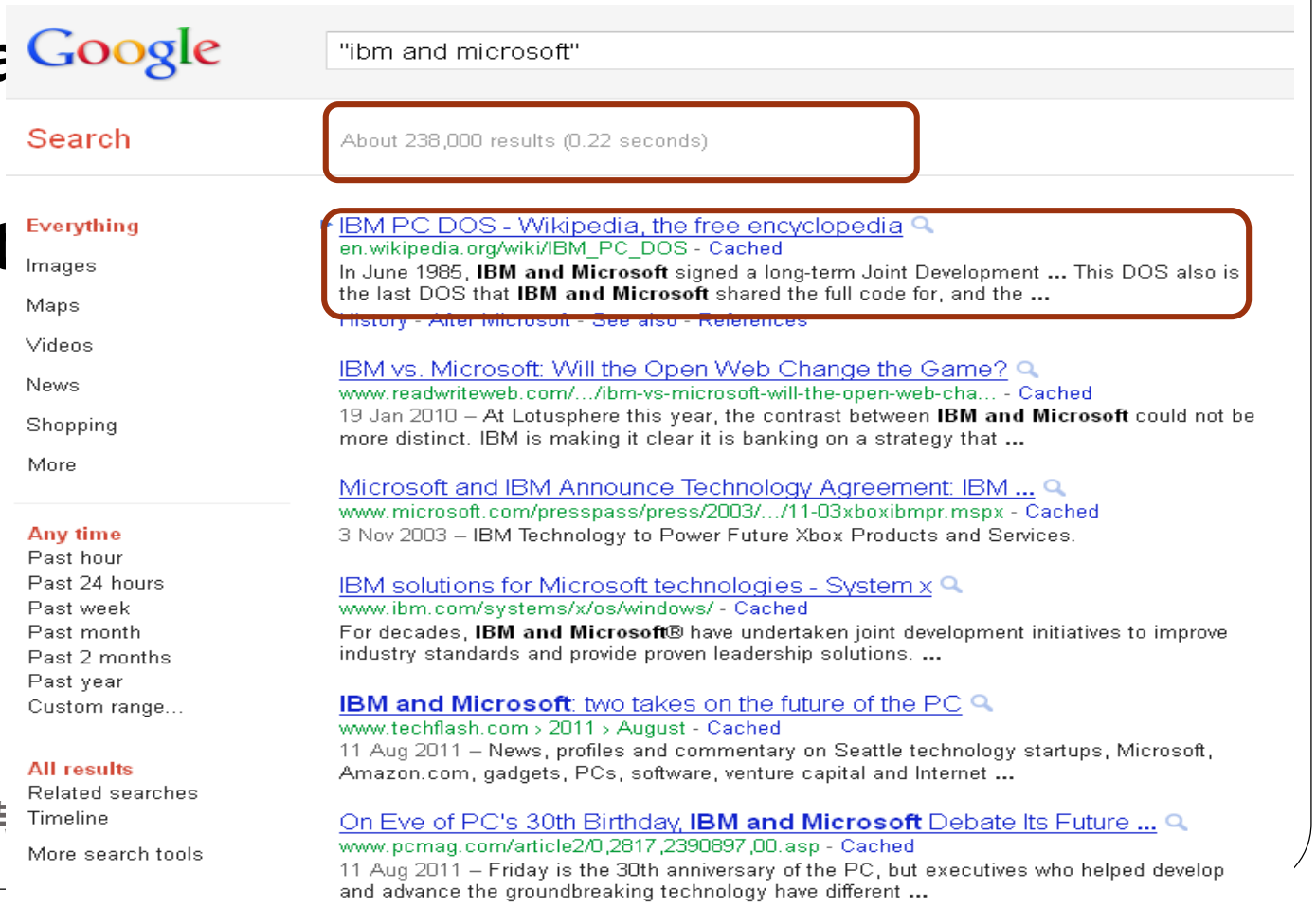
T1	T2	T3	T4	T5	T6	T7	T8	T9
0.35	0.22	0.34	0.26	0.22	0.49	0.28	0.29	0.44
-0.32	-0.15	-0.46	-0.24	-0.14	0.55	0.07	-0.31	0.44
-0.41	0.14	-0.16	0.25	0.22	-0.51	0.55	0.00	0.34

进行投影=>



基于Web Search的词汇语义计算

- 利用搜索引擎返回结果计算词汇之间共现关系
  - “A and B”
- HITS(Pa
- Snippet



基于检索页面数量的方法

- **Page-count-based Similarity Scores (co-occurrence measures)**

$$\text{WebJaccard}(P,Q) = \begin{cases} 0 & \text{if } H(P\cap Q) \le c \\ \frac{H(P\cap Q)}{H(P)+H(Q)-H(P\cap Q)} & \text{otherwise.} \end{cases} \quad (1)$$

$$\text{WebOverlap}(P,Q) = \begin{cases} 0 & \text{if } H(P\cap Q) \le c \\ \frac{H(P\cap Q)}{\min(H(P),H(Q))} & \text{otherwise.} \end{cases}$$

$$\text{WebDice}(P,Q) = \begin{cases} 0 & \text{if } H(P\cap Q) \le c \\ \frac{2H(P\cap Q)}{H(P)+H(Q)} & \text{otherwise.} \end{cases} \quad (3)$$

$$\text{WebPMI}(P,Q) = \begin{cases} 0 & \text{if } H(P\cap Q) \le c \\ \log_2\left(\frac{\frac{H(P\cap Q)}{N}}{\frac{H(P)}{N}\frac{H(Q)}{N}}\right) & \text{otherwise.} \end{cases} \quad (4)$$

***C* = 5**

Google Distance

- **计算的是距离**

The google similarity distance

RL Cilibrasi, PMB Vitanyi - IEEE Transactions on knowledge …, 2007 - ieeexplore.ieee.org

Abstract: Words and phrases acquire meaning from the way they are used in society, from their relative semantics to other words and phrases. For computers, the equivalent of" society" is" database," and the equivalent of" use" is" a way to search the database". We

被引用次数：1570　相关文章　所有 26 个版本　引用　保存

$$\begin{aligned} \mathrm{NGD}(x,y) &= \frac{G(x,y)-\min(G(x),G(y))}{\max(G(x),G(y))} \qquad \text{(III.3)} \\ &= \frac{\max\{\log f(x),\log f(y)\}-\log f(x,y)}{\log N-\min\{\log f(x),\log f(y)\}}, \end{aligned}$$

$\mathrm{NGD}(horse,rider)\approx 0.443.$

f(x), f(y): 包含x或y的网页数量
f(x,y): 同时包含x与y的网页数量

基于词语共现的CODC方法 [Chen et al. 2006]

$$CODC(X,Y) = \begin{cases} 0 & if\ f(Y@X)=0\ \ or\ \ f(X@Y)=0 \\ e^{\log\left(\frac{f(Y@X)}{f(X)}\times\frac{f(X@Y)}{f(Y)}\right)^{\alpha}} & Otherwise \end{cases}$$

- Co-Occurrence Double-Check
- f(Y@X): 在查询X的前N个检索结果摘要中Y的出现总次数.
- f(X):在查询X的前N个检索结果摘要中X的出现总次数.

基于检索结果snippet相似度的方法
[Sahami et al. 2006]

1. Issue x as a query to a search engine S.
2. Let $R(x)$ be the set of (at most) n retrieved documents $d_1, d_2, \ldots, d_n$

R(x)：查询x的前n个检索文档集合

3. Compute the TFIDF term vector v_i for each document $d_i \in R(x)$

为R(x)中每个文档d_i计算其TFIDF向量表示v_i

4. Truncate each vector v_i to include its m highest weighted terms

对v_i进行简化，只包含向量中权重最高的m个词语

5. Let $C(x)$ be the centroid of the L_2 normalized vectors v_i:

$$C(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{v_i}{\|v_i\|_2}$$

C(x)：文档集合的中心向量（规范化之后文档向量的平均值）

6. Let $QE(x)$ be the L_2 normalization of the centroid $C(x)$:

$$QE(x) = \frac{C(x)}{\|C(x)\|_2}$$

QE(x)：对C(x)进一步规范化

最终计算QE向量相似度得到词语相似度

基于分类的方法[Bollegala 2007]

- 融合多种特征：包括模式特征，共现值特征等
  - 模式：is a （X is a Y)，and （X and Y), 等等
  - 共现值：WebJaccard, WebDice, WebOverlap, WebPMI
  - F= [200 Pattern Freq, 4 co-occurrence measures]
- Two-class SVM
  - synonymous word-pairs (Positive): extracted from WordNet
  - non-synonymous word-pairs (Negative)

Table 4: Semantic Similarity of Human Ratings and Baselines on Miller-Charles' dataset

Word Pair	Miller-Charles'	Web Jaccard	Web Dice	Web Overlap	Web PMI	Sahami [36]	CODC [6]	Proposed SemSim
cord-smile	0.13	0.102	0.108	0.036	0.207	0.090	0	0
rooster-voyage	0.08	0.011	0.012	0.021	0.228	0.197	0	0.017
noon-string	0.08	0.126	0.133	0.060	0.101	0.082	0	0.018
glass-magician	0.11	0.117	0.124	0.408	0.598	0.143	0	0.180
monk-slave	0.55	0.181	0.191	0.067	0.610	0.095	0	0.375
coast-forest	0.42	0.862	0.870	0.310	0.417	0.248	0	0.405
monk-oracle	1.1	0.016	0.017	0.023	0	0.045	0	0.328
lad-wizard	0.42	0.072	0.077	0.070	0.426	0.149	0	0.220
forest-graveyard	0.84	0.068	0.072	0.246	0.494	0	0	0.547
food-rooster	0.89	0.012	0.013	0.425	0.207	0.075	0	0.060
coast-hill	0.87	0.963	0.965	0.279	0.350	0.293	0	0.874
car-journey	1.16	0.444	0.460	0.378	0.204	0.189	0.290	0.286
crane-implement	1.68	0.071	0.076	0.119	0.193	0.152	0	0.133
brother-lad	1.66	0.189	0.199	0.369	0.644	0.236	0.379	0.344
bird-crane	2.97	0.235	0.247	0.226	0.515	0.223	0	0.879
bird-cock	3.05	0.153	0.162	0.162	0.428	0.058	0.502	0.593
food-fruit	3.08	0.753	0.765	1	0.448	0.181	0.338	0.998
brother-monk	2.82	0.261	0.274	0.340	0.622	0.267	0.547	0.377
asylum-madhouse	3.61	0.024	0.025	0.102	0.813	0.212	0	0.773
furnace-stove	3.11	0.401	0.417	0.118	1	0.310	0.928	0.889
magician-wizard	3.5	0.295	0.309	0.383	0.863	0.233	0.671	1
journey-voyage	3.84	0.415	0.431	0.182	0.467	0.524	0.417	0.996
coast-shore	3.7	0.786	0.796	0.521	0.561	0.381	0.518	0.945
implement-tool	2.95	1	1	0.517	0.296	0.419	0.419	0.684
boy-lad	3.76	0.186	0.196	0.601	0.631	0.471	0	0.974
automobile-car	3.92	0.654	0.668	0.834	0.427	1	0.686	0.980
midday-noon	3.42	0.106	0.112	0.135	0.586	0.289	0.856	0.819
gem-jewel	3.84	0.295	0.309	0.094	0.687	0.211	1	0.686
Correlation	1	0.259	0.267	0.382	0.548	0.579	0.693	0.834

RG-65上的各种结果比较

http://www.aclweb.org/aclwiki/index.php?title=RG-65_Test_Collection_%28State_of_the_art%29

Algorithm	Reference for algorithm	Reference for reported results	Type	Spearman correlation (ρ)	Pearson correlation (r)
ADW	Pilehvar and Navigli (2015)	Pilehvar and Navigli (2015)	Knowledge-based (Wiktionary)	0.920	0.910
Y&Q	Yih and Qazvinian (2012)	Yih and Qazvinian (2012)	Hybrid	0.890	-
NASARI	Camacho-Collados et al. (2015)	Camacho-Collados et al. (2015)	Hybrid	0.880	0.910
ADW	Pilehvar et al. (2013)	Pilehvar et al. (2013)	Knowledge-based (WordNet)	0.868	0.810
PPR	Hughes and Ramage (2007)	Hughes and Ramage (2007)	Knowledge-based	0.838	-
SSA	Hassan and Mihalcea (2011)	Hassan and Mihalcea (2011)	Corpus-based	0.833	0.861
PPR	Agirre et al. (2009)	Agirre et al. (2009)	Knowledge-based	0.830	-
H&S	Hirst and St-Onge (1998)	Hassan and Mihalcea (2011)	Knowledge-based	0.813	0.732
Roget	Jarmasz (2003)	Hassan and Mihalcea (2011)	Knowledge-based	0.804	0.818
J&C	Jiang and Conrath (1997)	Hassan and Mihalcea (2011)	Knowledge-based	0.804	0.731
WNE	Jarmasz (2003)	Hassan and Mihalcea (2011)	Knowledge-based	0.801	0.787
L&C	Leacock and Chodorow (1998)	Hassan and Mihalcea (2011)	Knowledge-based	0.797	0.852
Lin	Lin (1998)	Hassan and Mihalcea (2011)	Corpus-based	0.788	0.834
ESA*	Gabrilovich and Markovitch (2007)	Hassan and Mihalcea (2011)	Corpus-based	0.749	0.716
SOCPMI*	Islam and Inkpen (2006)	Hassan and Mihalcea (2011)	Corpus-based	0.741	0.729
Resnik	Resnik (1995)	Hassan and Mihalcea (2011)	Knowledge-based	0.731	0.800
WLM	Milne and Witten (2008)	Milne and Witten (2008)	Knowledge-based	0.640	-
LSA*	Landauer et al. (1997)	Hassan and Mihalcea (2011)	Corpus-based	0.609	0.644
WikiRelate	Strube and Ponzetto (2006)	Strube and Ponzetto (2006)	Knowledge-based	-	0.530

Note: values reported by (Hassan and Mihalcea, 2011) are "based on the collected raw data from the respective authors", and those highlighted by (*) are re-implementations.

基于表示学习的方法

- 学习得到每个词语的（密集）向量表示
  - 基于机器学习、深度学习等
  - 一般基于生语料
  - 基于一定的假设构造优化目标与目标函数，然后通过算法求解获得词语的向量表示

- one-hot vector vs. dense vector

star [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, ···]
sun [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, ···]

star [0.1, 0.25,0.34, 0.25]
sun [0.25,0.33,0.44, 0.54]

sim(star, sun) = 0

基于表示学习的方法

- 基本假设
  - 文本语料中词语与其上下文词语的语义表示应该尽可能一致/相容

“习大大领导中国走向现代化！”

基于表示学习的方法

- 在深度学习/神经网络之外早有其他学习方法
- CLEAR [Halawi et al. 2012]
  - 利用生语料学习词语向量表示
  - 利用词典中词汇关系作为约束

基于表示学习的方法

- CLEAR [Halawi et al. 2012]
  - 做法
    - 从文本中构造词序列(不超过一个句子)
      (前k个词，中心词(i_s)，后k个词)
      s表示前k个+后k个词组成的序列/句子（不包括i_s）
    - 每个词i对应的向量q_i
    - 那么每个句子s对应的向量为p_s，且 $p_s = \frac{1}{|s|}\sum_{i \in s} q_i$
    - 词i和句子s的相似性为 $r_{si} = b_i + q_i^T p_s$ ，其中b_i为偏置
    - Θ 为模型参数，包括词向量与偏置

基于表示学习的方法

- CLEAR [Halawi et al. 2012]
  - 做法
    - 从句子s中观察到词i的可能性定义为

softmax

$$P(i|s;\Theta) = \frac{\exp(r_{si})}{\sum_j \exp(r_{sj})}$$

    - 优化目标为在训练集上最大化如下对数似然：

$$L(\mathcal{S};\Theta) \stackrel{\text{def}}{=} \sum_{s\in\mathcal{S}} \log P(i_s|s;\Theta)$$

    - 利用随机梯度下降算法进行学习

基于表示学习的方法

- CLEAR [Halawi et al. 2012]
  - 进一步，引入词典中词汇关系作为约束
  - P： 考虑的WordNet中得到的词义关系集合（例如，同义词，上位词、下位词等）
  - 优化目标变为：

$$L(\mathcal{S};\Theta) \stackrel{\text{def}}{=} \sum_{s\in\mathcal{S}} \log P(i_s|s,\Theta) - \lambda \sum_{(i_1,i_2)\in\mathcal{P}} \|q_{i_1} - q_{i_2}\|^2$$

  - 最终词汇相似度计算 $\text{rel}_{ij} = \dfrac{q_i^T q_j}{\|q_i\|\|q_j\|}$

基于表示学习的方法

- **CLEAR [Halawi et al. 2012]**

coke	mile	religion	sheep	university	webcam
pepsi (0.813)	mi (0.826)	relgion (0.779)	ox (0.648)	suny (0.809)	cam (0.636)
cola (0.567)	kilometer (0.690)	abrahamic (0.771)	ewe (0.620)	nyu (0.806)	logitech (0.581)
sprite (0.491)	nautical (0.607)	spirituality (0.717)	goat (0.614)	devry (0.804)	messenge (0.519)
coca (0.477)	kilometre (0.589)	religon (0.715)	herd (0.612)	polytechnic (0.801)	messenger (0.484)
pepsico (0.471)	furlong (0.563)	secularism (0.674)	cattle (0.596)	univeristy (0.795)	camera (0.474)

Method	WS-353	MTURK-287	MTURK-771
CLEAR	**0.810**	**0.737**	**0.727**
TSA	0.8	$0.61^{\S}$	$0.606^{\S}$
ESA	$0.75^{\S}$	$0.607^{\S}$	$0.603^{\S}$
LDA	$0.736^{\S}$	$0.677^{\dagger}$	$0.619^{\S}$
DS	$0.61^{\S}$	$0.625^{\S}$	$0.578^{\S}$

基于表示学习的方法

- C&W embedding [Collobert et al. 2011]
  - 一个词与其上下文构成正样例，而其他随机的词与该上下文构成负样例
    - 上下文大小由窗口控制

cat chills on a mat

cat chills Jeju a mat

score(cat chills on a mat) > score(cat chills Jeju a mat)

基于表示学习的方法

- **C&W embedding**

Score的计算

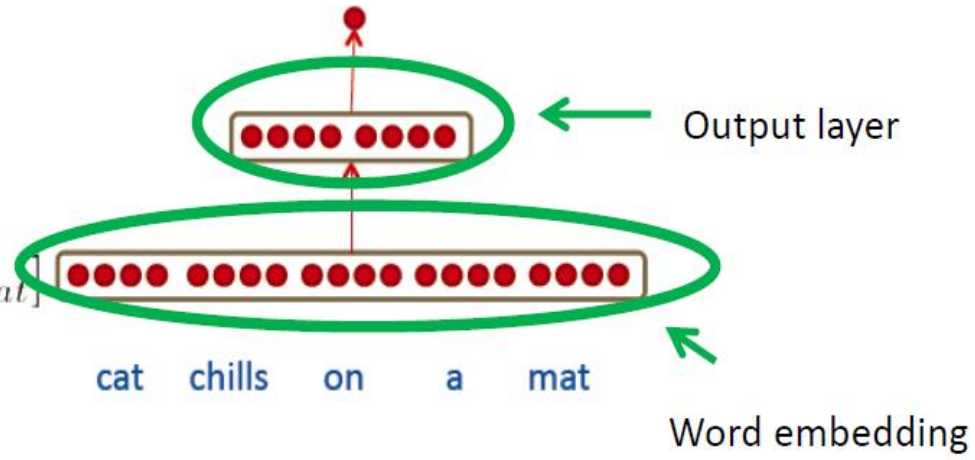
$$s = U^T f(Wx + b) \qquad x \in \mathbb{R}^{20\times 1}, W \in \mathbb{R}^{8\times 20}, U \in \mathbb{R}^{8\times 1}$$

$$\begin{aligned} s &= U^T a \\ a &= f(z) \\ z &= Wx + b \end{aligned}$$

句子的的表示（拼接词向量）

$$x = [x_{cat} \quad x_{chills} \quad x_{on} \quad x_a \quad x_{mat}]$$

$$L \in \mathbb{R}^{n\times|V|}$$



s = score(cat chills on a mat)

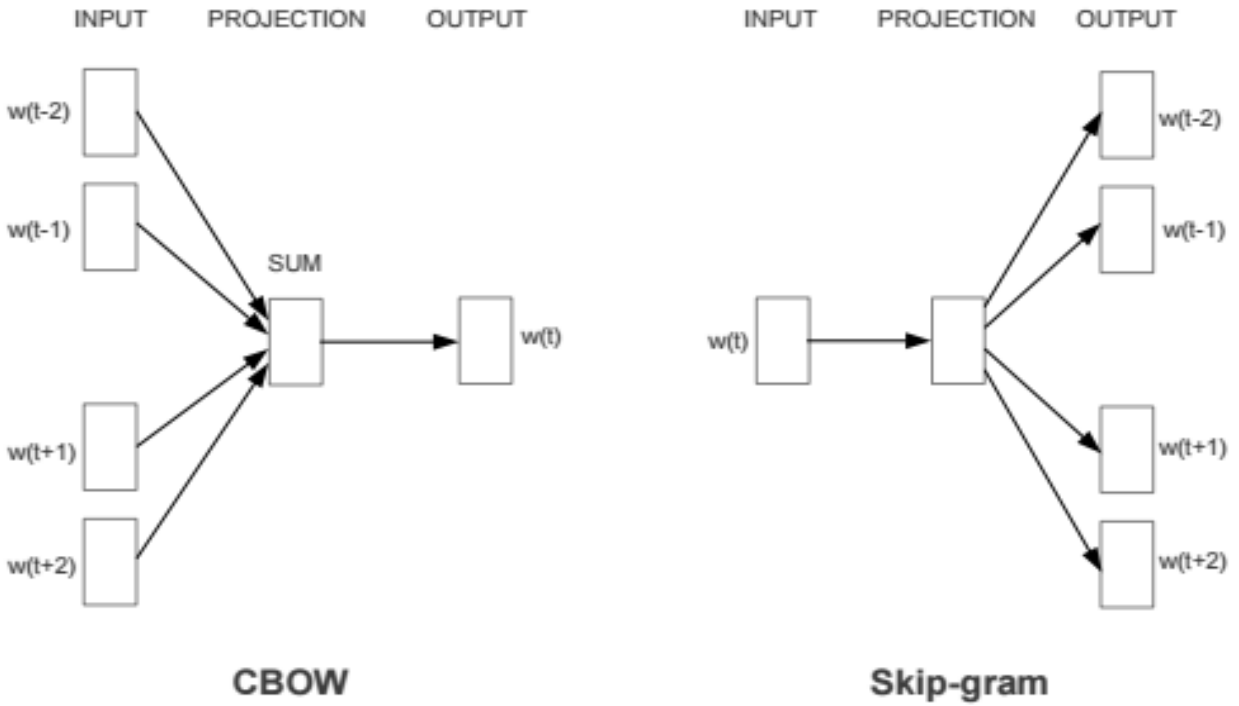
s_c = score(cat chills Jeju a mat)

max-margin loss

Minimize $J = \max(0, 1 - s + s_c)$

基于表示学习的方法

- 类似的: CBOW and Skip-gram
- 在Word2Vec中有具体实现



基于表示学习的方法

- 可以自定义优化目标函数，为特定应用服务
- 可以利用有监督的训练数据获得更好的词向量

$$X_{apple} - X_{apples} \approx X_{car} - X_{cars} \approx X_{family} - X_{families}$$

$$X_{shirt} - X_{clothing} \approx X_{chair} - X_{furniture}$$

词汇向量表示的可解释性

- 基于维基百科的ESA向量可解释性最好
- 上下文向量可解释性也不错
- 隐含语义分析与表示学习得到的向量解释性很差

词汇向量表示的可解释性

- 如何解释表示学习得到的向量?
  - 有工作[IJCAI16]提出利用L1正则项得到稀疏矩阵表示

Max $\mathcal{L}_{s-cbow} = \mathcal{L}_{cbow} - \lambda \sum_{w \in W} \|\vec{w}\|_1$ $\|\boldsymbol{x}\|_1 := \sum_{i=1}^{n} |x_i|$.

$$\mathcal{L}_{cbow} = \sum_{i=1}^{N} \Big(\log p(w_i|h_i) \Big)$$

$$\vec{h}_i = \frac{1}{2l} \sum_{\substack{j=i-l \\ j \neq i}}^{i+l} \vec{c}_j$$

$$p(w_i|h_i) = \frac{\exp(\vec{w}_i \cdot \vec{h}_i)}{\sum_{w \in W} \exp(\vec{w} \cdot \vec{h}_i)}$$

词汇向量表示的可解释性

- 如何解释表示学习得到的向量?
  - 有工作提出利用L1正则项得到稀疏矩阵表示
  - 向量每一维用该维度上权重最大的几个词来表示

Table 3: Top 5 words of some dimensions in CBOW and Sparse CBOW.

Model	Top 5 Words
CBOW	beat, finish, wedding, prize, read
	rainfall, footballer, breakfast, weekdays, angeles
	landfall, interview, asked, apology, dinner
	becomes, died, feels, resigned, strained
	best, safest, iucn, capita, tallest
Sparse CBOW	poisson, parametric, markov, bayesian, stochastic
	ntfs, gzip, myfile, filenames, subdirectories
	hugely, enormously, immensely, wildly, tremendously
	earthquake, quake, uprooted, levees, spectacularly
	bosons, accretion, higgs, neutrinos, quarks

第一次作业

- 词汇相关度计算
- 实现3种词汇相关度计算方法，尽量保证方法的多样性
- 基于Mturk-771进行实验和分析（开放式）
  - http://www2.mta.ac.il/~gideon/mturk771.html
- 提交压缩文件包到 sckr2018@126.com
  - 姓名、学号
  - 自己编写的代码（调用的大型工具包不用提交，只需在报告中说明即可），编程语言不限
  - 2～3页小报告（包括调用的工具或资源，实验方法、结果比较与分析、想法等）
- 提交时间3月31日

相关文献

- “Evaluating WordNet-based measures of lexical semantic relatedness” by Budanitsky, A. and Hirst, G. (2006)
- “Using Wikionary for Computing Semantic Relatedness” by T. Zesch, C. Muller and I. Gürevych. (2008)
- “Measures of distributional similarity” by L. Lee. (1999)
- “Co-occurrence retrieval: A flexible framework for lexical distributional similarity” by J. Weeds and D. Weir. (2005)
- “Indexing by Latent Semantic Analysis” by S. Deerwester, S. Dumais, G. Furnas, T. Landauer, R. Harshman. (1990)
- “Computing semantic relatedness using Wikipedia-based Explicit Semantic Analysis” by E. Gabrilovich and S. Markovitch (2007)
- “Measuring semantic similarity between words using Web search engines” by D. Bollegala, Y. Matsuo and M. Ishizuka. (2007)
- “A web-based kernel function for measuring the similarity of short text snippets” by M. Sahami and T. Heilman. (2006)
- “Novel association measures using web search with double checking” by H. Chen, M. Lin, and Y. Wei. (2006)
- “The google similarity distance”, by RL Cilibrasi, P. Vitanyi. TKDE 2007.
- Halawi, Guy, et al. "Large-scale learning of word relatedness with constraints." *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012.
- Collobert, Ronan, et al. "Natural language processing (almost) from scratch." *The Journal of Machine Learning Research* 12 (2011): 2493-2537.

Acknowledgements

- Some slides were taken or adapted from related slides written by George A. Miller, Cosmin Adrian Bejan, Marian Olteanu, Giuseppe Carenini, Pu Wang, Keith Trnka, Danushka Bollegala, etc. Thank them for sharing their slides.