

Módulo 4: RNNs

Aula 1: Modelos Neurais de Linguagem

Fabricio Murai

[murai at dcc.ufmg.br](mailto:murai@dcc.ufmg.br)

Aula de hoje

- Modelagem de Linguagem
- Modelos n-grama
- Representações Distribuídas
- Resolvendo analogias
- GloVe (opcional)

Modelagem de Linguagem

Modelagem de Linguagem

Motivação: suponha que queiramos construir um sistema de reconhecimento de fala. (Entrada? Saída?)

Tarefa: $p(\text{frase } \mathbf{s} | \text{sinal } \mathbf{a})$?

Abordagem **generativa** tem 2 componentes:

- **Distribuição a priori** $p(\mathbf{s})$: quão provável é uma frase \mathbf{s} .
Deve saber que “the apple and pear salad” é mais provável que “the apple and pair salad”.
- **Modelo observacional** $p(\mathbf{a} | \mathbf{s})$: quão provável é que a frase \mathbf{s} leve ao sinal acústico \mathbf{a} .

Podemos usar regra de Bayes para inferir distribuição a posteriori sobre frases dado um sinal:

$$p(\mathbf{s} | \mathbf{a}) = \frac{p(\mathbf{s})p(\mathbf{a} | \mathbf{s})}{\sum_{\mathbf{s}'} p(\mathbf{s}')p(\mathbf{a} | \mathbf{s}')}.$$

Modelagem de Linguagem

Foco da aula: como aprender uma boa distribuição $p(\mathbf{s})$ sobre frases? Problema conhecido como modelagem de linguagem. (sugestões?)

Suponha um corpus de frases $\mathbf{s}^{(1)}, \dots, \mathbf{s}^{(N)}$. O critério da **máxima verossimilhança** diz que queremos que nosso modelo **maximize** a probabilidade atribuída às sentenças observadas.

Assumindo que frases são independentes, probabilidades são multiplicadas na otimização

$$\max \prod_{i=1}^N p(\mathbf{s}^{(i)}).$$

Modelagem de Linguagem

Em estimação de máxima verossimilhança, queremos maximizar $\prod_{i=1}^N p(\mathbf{s}^{(i)})$.

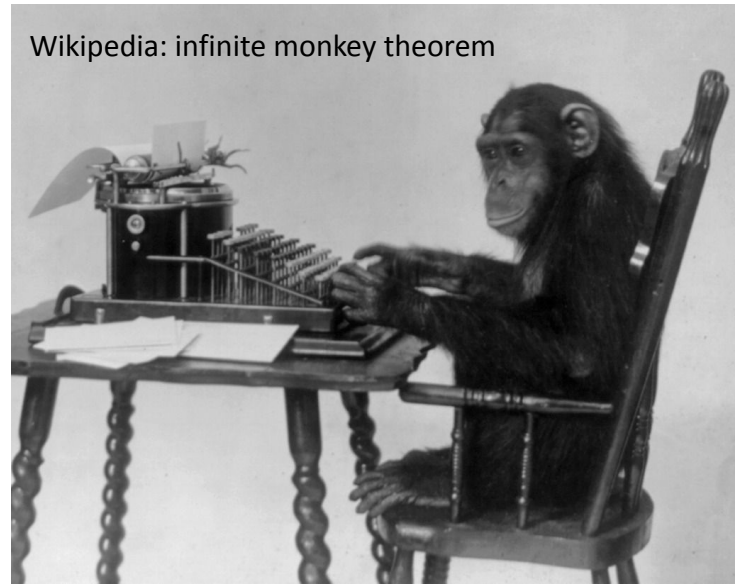
Qualquer que seja o modelo, o produtório acima será praticamente zero -- como a chance de um macaco digitar uma obra de Shakespeare.

- Como de praxe, tomamos o **log das probabilidades**. Isso também ajuda a decompor a função objetivo em somas

$$\log \prod_{i=1}^N p(\mathbf{s}^{(i)}) = \sum_{i=1}^N \log p(\mathbf{s}^{(i)}).$$

- Vamos usar o *negativo* das log probabilidades, de forma a trabalhar com números positivos.

Wikipedia: infinite monkey theorem



Modelagem de Linguagem

- Mas o que significa **probabilidade de uma frase**?
 - Calcular a frequência relativa de cada frase exata **não** é um boa ideia.
 - Frase \mathbf{s} é uma sequência de palavras w_1, w_2, \dots, w_T . Usando a regra da cadeia da probabilidade condicional, podemos decompor $p(\mathbf{s})$ como

$$p(\mathbf{s}) = p(w_1, \dots, w_T) = p(w_1)p(w_2 | w_1) \cdots p(w_T | w_1, \dots, w_{T-1}).$$

- Portanto, o problema de modelagem de linguagem é equivalente a prever a próxima palavra.

Modelagem de Linguagem

- Tipicamente fazemos uma **suposição Markoviana**: distribuição sobre a próxima palavra depende apenas das últimas K palavras.

Se usarmos um contexto de tamanho $K=3$,

$$p(w_t \mid w_1, \dots, w_{t-1}) = p(w_t \mid w_{t-3}, w_{t-2}, w_{t-1}).$$

- Modelo é dito **sem memória**.
- Agora temos problema de aprendizado supervisionado. Objetivo é prever a distribuição condicional de cada palavra dadas as K anteriores.

Modelos n-grama

Modelos n-grama

- Um tipo comum de modelo Markoviano usa tabela de probabilidade condicional.

E.g. (K=2):

	cat	and	city	...
the fat	0.21	0.003	0.01	
four score	0.0001	0.55	0.0001	...
New York	0.002	0.0001	0.48	
⋮		⋮		

- Modo mais simples de estimar tabela é distribuição empírica:

$$p(w_3 = \text{cat} \mid w_1 = \text{the}, w_2 = \text{fat}) = \frac{\text{count}(\text{the fat cat})}{\text{count}(\text{the fat})}$$

- Este é o estimador de máxima verossimilhança
- Sequências que estamos contando são chamadas n-gramas (no caso, n=3), por isso é um **modelo de linguagem n-grama**.

Modelos n-grama

“os estudantes abriram as _____”

Q: Como aprender um modelo de linguagem?

A: Com um modelo n-grama!

Definição: um **n-grama** é uma sequência de n palavras consecutivas

unigrama: “os”, “estudantes”, “abriram”, “as”

bigrama: “os estudantes”, “estudantes abriram”, “abriram as”

trigrama: “os estudantes abriram”, “estudantes abriram as”

4-grama: “os estudantes abriram as”

Modelos n-grama

Premissa (simples): $\mathbf{x}^{(t+1)}$ depende apenas das **$n-1$** palavras anteriores

$$P(\mathbf{x}^{(t+1)} | \mathbf{x}^{(t)}, \dots, \mathbf{x}^{(1)}) = P(\mathbf{x}^{(t+1)} | \overbrace{\mathbf{x}^{(t)}, \dots, \mathbf{x}^{(t-n+2)}}^{n-1 \text{ words}}) \quad (\text{assumption})$$

prob of a n-gram \rightarrow

prob of a (n-1)-gram \rightarrow

$$\begin{aligned} &= \boxed{P(\mathbf{x}^{(t+1)}, \mathbf{x}^{(t)}, \dots, \mathbf{x}^{(t-n+2)})} \\ &= \boxed{P(\mathbf{x}^{(t)}, \dots, \mathbf{x}^{(t-n+2)})} \end{aligned} \quad \left| \begin{array}{l} \text{(definition of} \\ \text{conditional prob)} \end{array} \right.$$

Q: Como obtemos essas probabilidades?

A: Apenas contamos elas em um grande conjunto de texto!

$$\approx \frac{\text{count}(\mathbf{x}^{(t+1)}, \mathbf{x}^{(t)}, \dots, \mathbf{x}^{(t-n+2)})}{\text{count}(\mathbf{x}^{(t)}, \dots, \mathbf{x}^{(t-n+2)})} \quad \begin{array}{l} \text{(statistical} \\ \text{approximation)} \end{array}$$

Modelos n-grama

Suponha que estamos aprendendo um modelo 4-grama:

“assim que o professor permitiu, os estudantes abriram as _____”

$$P(\mathbf{w} | \text{estudantes abriram as}) = \frac{\text{count}(\text{estudantes abriram as } \mathbf{w})}{\text{count}(\text{estudantes abriram as})}$$

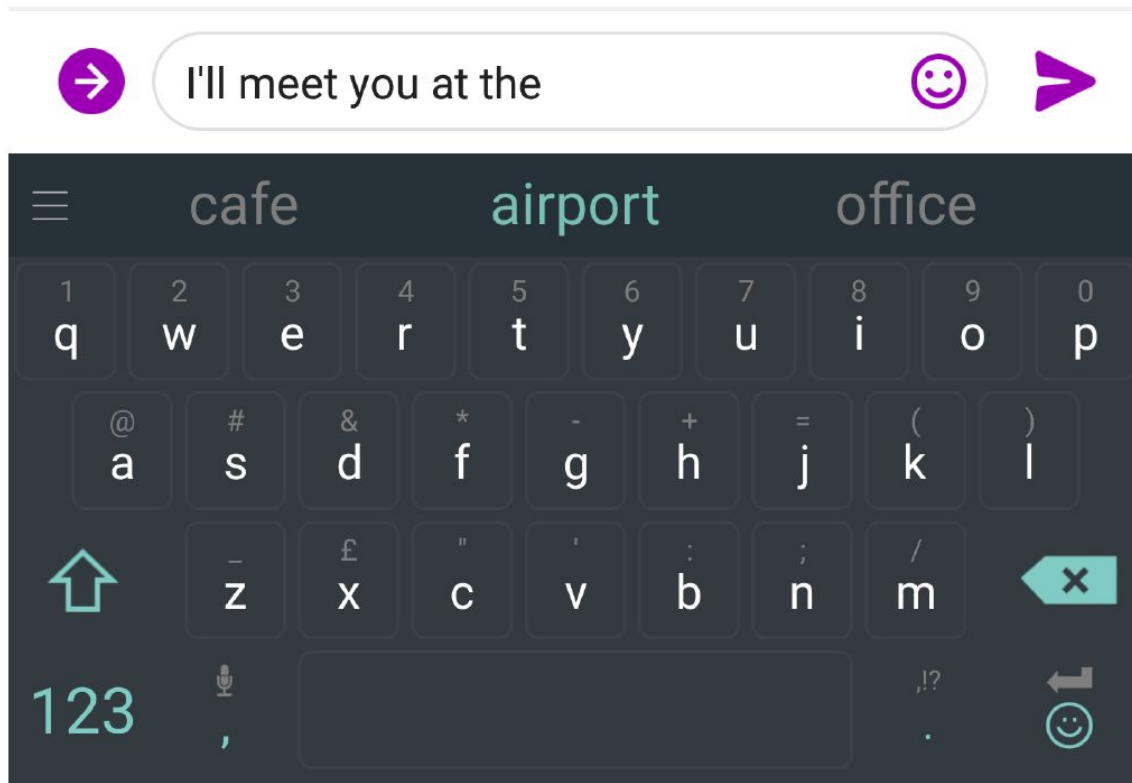
“estudantes abriram as” ocorreu 1000 vezes

“estudantes abriram as mochilas” ocorreu 400 vezes

“estudantes abriram as provas” ocorreu 100 vezes


Deveríamos ter
ignorado o texto
sublinhado?

Aplicações de Modelos de Linguagem



Aplicações de Modelos de Linguagem



what is the | 

what is the **weather**
what is the **meaning of life**
what is the **dark web**
what is the **xfl**
what is the **doomsday clock**
what is the **weather today**
what is the **keto diet**
what is the **american dream**
what is the **speed of light**
what is the **bill of rights**

[Google Search](#) [I'm Feeling Lucky](#)

Modelos n-grama

Geração de texto (Wall Street Journal):

1
gram

Months the my and issue of year foreign new exchange's september
were recession exchange new endorsed a acquire to six executives

2
gram

Last December through the way to preserve the Hudson corporation N.
B. E. C. Taylor would seem to complete the major central planners one
point five percent of U. S. E. has already old M. X. corporation of living
on information such as more frequently fishing to keep her

3
gram

They also point to ninety nine point six billion dollars from two hundred
four oh six three percent of the rates of interest stores as Mexico and
Brazil on market conditions

Modelos n-grama

- Para brincar

https://www.dropbox.com/s/8lndowxyklorst/vanilla_linguagem_modelos.ipynb?dl=1

- Gerando letras de Ratos do Porão

Ficará manipulada por burgueses moralistas E não
há lugar Para você Farsa Nacionalista Farsa
Nacionalista Farsa Nacionalista A pátria armada
nas mãos dessa cambada De extrema direita

Modelos n-grama

- Problemas com modelos n-gramas:
 - À medida que **n** aumenta, o que acontece com a memória necessária para armazenar os n-gramas?
 - **Esparsidade dos dados**: a maioria dos n-gramas nunca aparece no corpus, mesmo quando são possíveis (e.g.: e se “estudantes abriram as *mochilas*” nunca apareceu nos dados?)
- Maneiras de lidar com esparsidade dos dados?
 - Usar contexto menor (trade-off: modelo menos poderoso)
 - Suavizar probabilidades (e.g., adicionando ocorrências imaginárias)
 - Prevendo com um ensemble de modelos n-grama com n diferentes

Representações Distribuídas

Representações distribuídas

- Tabelas de probabilidade condicionais são **representações localistas**: toda informação sobre uma palavra é armazenada em um lugar (i.e., coluna da tabela)
- Mas diferentes palavras são relacionadas, então devemos ser capazes de compartilhar informação entre elas. E.g.: Considere essa matriz de atributos de palavras:

	academic	politics	plural	person	building
student					
colleges					
legislators					
schoolhouse					

- E essa matriz de como cada atributo influencia próxima palavra

	bill	is	are	papers	built	standing
academic						
politics						
plural						
person						
building						

Representações distribuídas

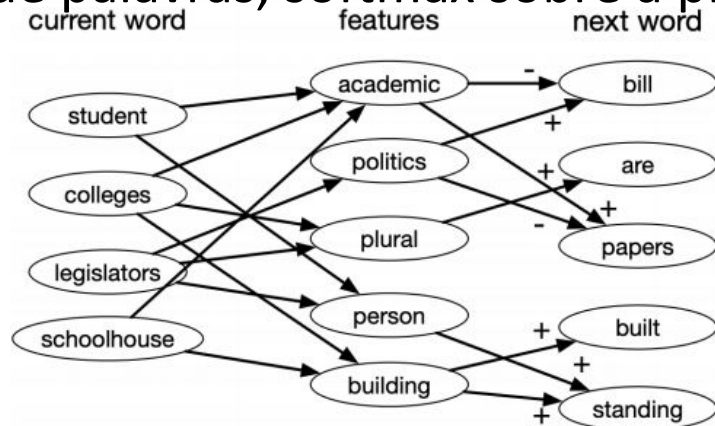
- Tabelas de probabilidade condicionais são **representações localistas**: toda informação sobre uma palavra é armazenada em um lugar (i.e., coluna da tabela)
- Mas diferentes palavras são relacionadas, então devemos ser capazes de compartilhar informação entre elas. E.g.: Considere essa matriz de atributos de palavras:

	academic	politics	plural	person	building
student	1	0	0	1	0
colleges	1	0	1	0	1
legislators	0	1	1	1	0
schoolhouse	1	0	0	0	1

- E essa matriz de como cada atributo influencia próxima palavra

	bill	is	are	papers	built	standing
academic	-			+		
politics	+			-		
plural		-	+			
person						+
building					+	+

- Imagine estas matrizes como camadas de um MLP (one-hot representations de palavras, softmax sobre a próxima palavra)



- Informação sobre uma palavra é distribuída sobre a representação (features), por isso a chamamos **representação distribuída**.
- Atenção:** em geral, quando treinamos MLP com backprop, unidades ocultas não terão significados intuitivos como nesta ilustração. Mas ela é útil para dar intuição do que os MLPs podem representar.

Representações Distribuídas

- Nós queremos ser capazes de compartilhar informação entre palavras relacionadas.

E.g.: suponha tenhamos visto a frase

*A **jarra** de **vidro** contém suco de **laranja**.*

- Isto deve nos ajudar a prever as palavras na frase

*A **garrafa** de **plástico** contém suco de **cupuaçu**.*

- Um modelo n-grama não consegue generalizar assim, mas uma representação distribuída pode.

Modelo Neural de Linguagem

- Prever a distribuição da próxima palavra dadas as K anteriores é apenas um problema de classificação (multi-classe).
- **Entrada:** K palavras anteriores
- **Target:** próxima palavra
- **Perda:** entropia cruzada. Equivale a máxima verossimilhança:

$$\begin{aligned} -\log p(\mathbf{s}) &= -\log \prod_{t=1}^T p(w_t | w_{t-1}, \dots, w_{t-K}) \\ &= -\sum_{t=1}^T \log p(w_t | w_{t-1}, \dots, w_{t-K}) \\ &= -\sum_{t=1}^T \sum_{v=1}^V \mathbf{e}_{tv} \log y_{tv} \end{aligned}$$

onde \mathbf{e}_t é o one-hot encoding da t -ésima palavra

y_{tv} é a probabilidade prevista que t -ésima palavra tenha índice v .

Modelo Neural de Linguagem

- Eis um **modelo de linguagem probabilístico neural** clássico (ou apenas **modelo neural de linguagem**)

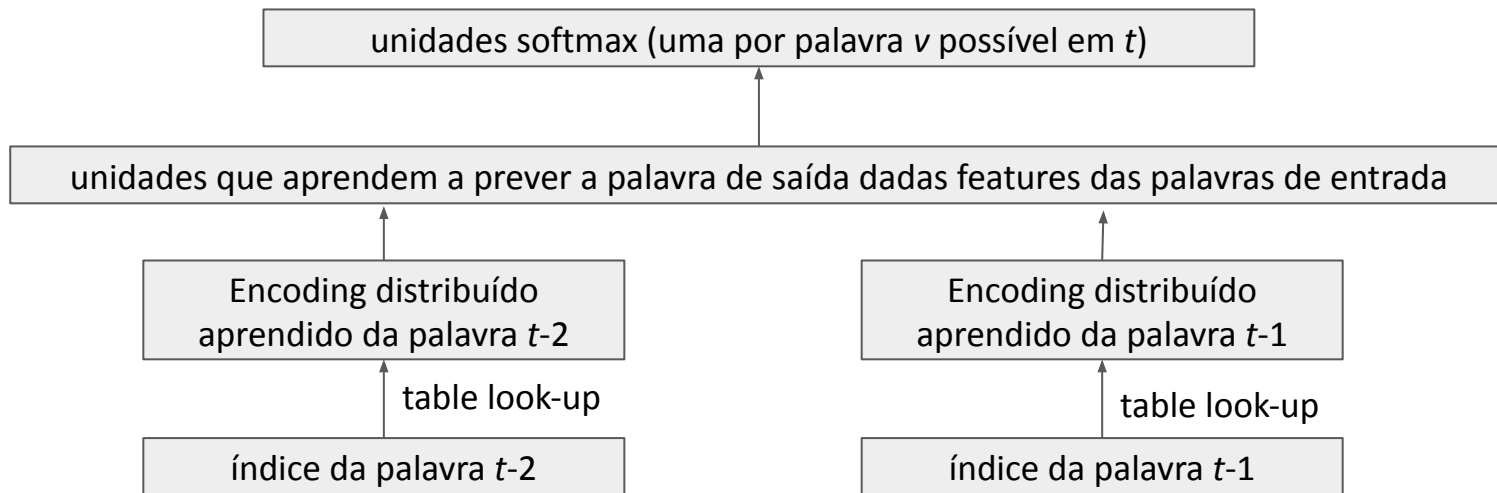


Table look-up

Exemplo de table look-up usando one-hot encoding

a	1
boa	2
é	5
filme	6
horrrível	7
mas	10
música	11
o	12
péssimo	13
roteiro	14

filme														
sentimento														
gênero														

O	filme	é	péssimo	roteiro	horrrível	mas	a	música	é	boa

Exemplo de table look-up usando one-hot encoding

a	1
boa	2
é	5
filme	6
horrrível	7
mas	10
música	11
o	12
péssimo	13
roteiro	14

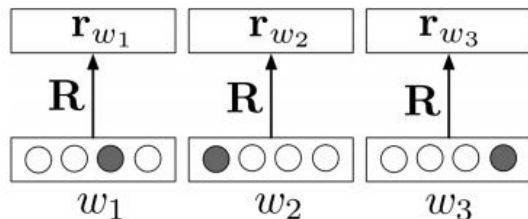
filme
sent.
gênero

0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
0	1	0	0	0	0	-1	0	0	0	0	0	-1	0	0
1	1	0	0	0	-1	0	0	0	1	-1	-1	1	0	0
1	2	3	4	5	6	7				12	13	14	15	

0	1	0	0	1	0	0	0	0	0	0
0	0	0	-1	0	-1	0	0	0	0	1
-1	-1	0	-1	-1	0	0	1	1	0	1
o	filme	é	péssimo	roteiro	horrrível	mas	a	música	é	boa

Modelo Neural de Linguagem

- Se usarmos um 1-of-K (one-hot) encoding para palavras, a 1ª camada pode ser vista como uma camada com pesos amarrados



- Matriz de pesos age como uma lookup table (seleção de coluna). Cada coluna é a **representação** de uma palavra, aka **embedding**, **feature vector** ou **encoding**.
 - “Embedding” enfatiza que é uma localização em um espaço de alta dimensão; palavras próximas são mais similares semanticamente
 - “Feature vector” enfatiza que é só um vetor que pode ser usado para fazer previsões, assim como outros mapeamentos que vimos (imagem→encoding)

Modelo Neural de Linguagem

- Podemos medir a (dis)similaridade de duas palavras usando
 - O dot product, aka produto interno $\mathbf{r}_1 \cdot \mathbf{r}_2 = \mathbf{r}_1^\top \mathbf{r}_2$
 - A distância Euclidiana $\|\mathbf{r}_1 - \mathbf{r}_2\|$
- Se os dois vetores tiverem norma unitária, eles são equivalentes:

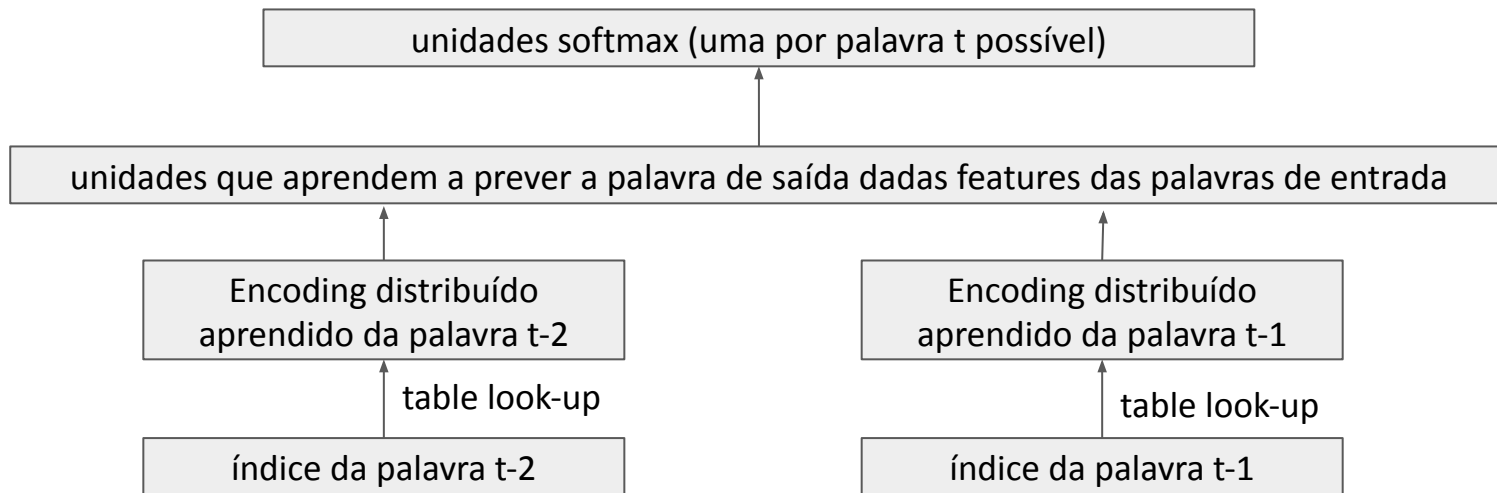
$$\begin{aligned}\|\mathbf{r}_1 - \mathbf{r}_2\|^2 &= (\mathbf{r}_1 - \mathbf{r}_2)^\top (\mathbf{r}_1 - \mathbf{r}_2) \\ &= \mathbf{r}_1^\top \mathbf{r}_1 - 2\mathbf{r}_1^\top \mathbf{r}_2 + \mathbf{r}_2^\top \mathbf{r}_2 \\ &= 2 - 2\mathbf{r}_1^\top \mathbf{r}_2\end{aligned}$$

- Muitas técnicas forçam normas unitárias. Neste caso, a **similaridade de cosseno** é dada pelo produto interno.

$$\cos(\theta) = \frac{\mathbf{r}_1 \cdot \mathbf{r}_2}{\|\mathbf{r}_1\| \|\mathbf{r}_2\|} = \mathbf{r}_1^\top \mathbf{r}_2$$

Modelo Neural de Linguagem

- Modelo é muito compacto: o número de parâmetros é linear no tamanho n do contexto (no n -grama, era exponencial em n)



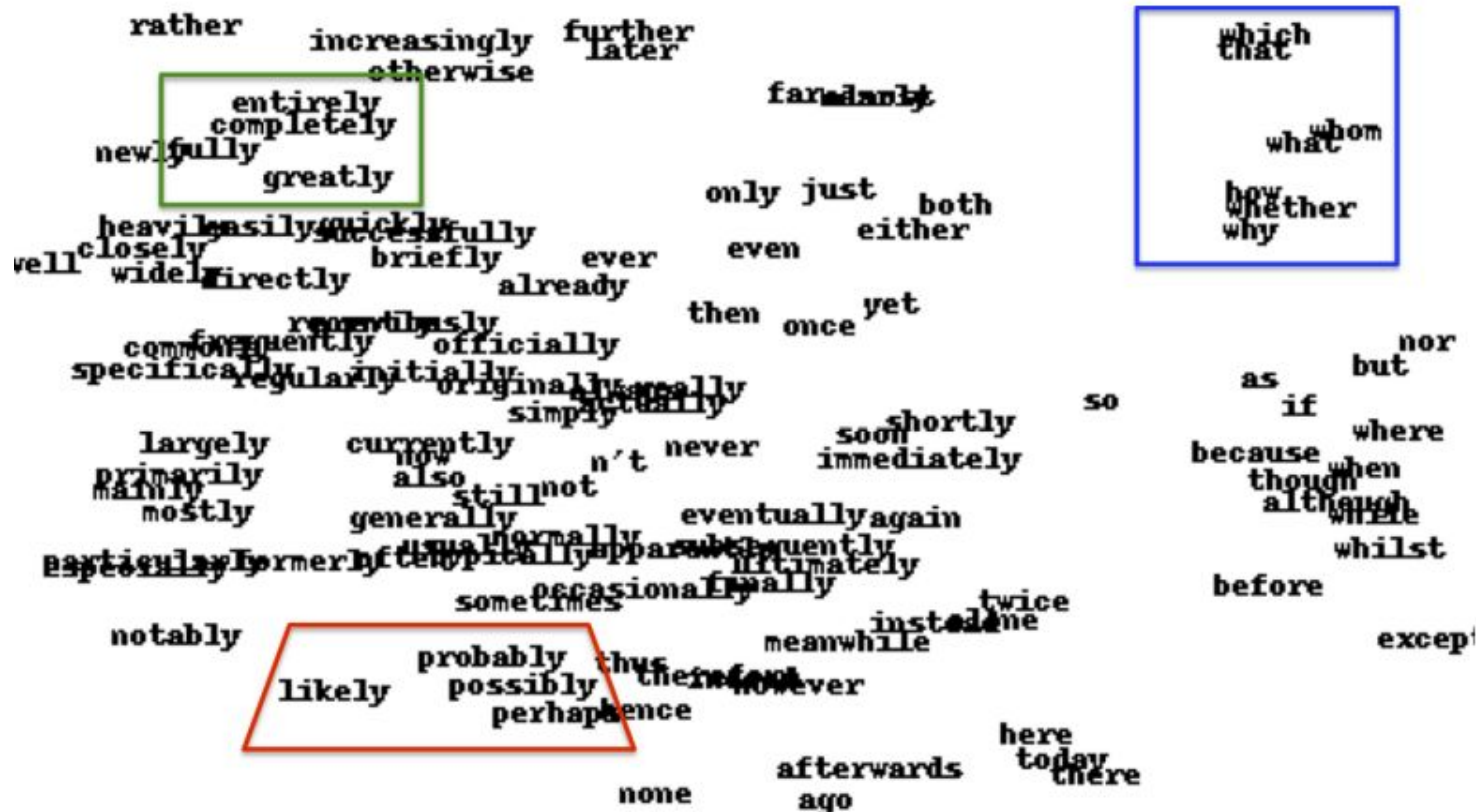
Modelo Neural de Linguagem

- Como se parecem visualmente estes embeddings?
- É difícil visualizar um espaço n -dimensional, mas já ouvimos falar de algoritmos de redução de dimensionalidade (quais?)
- Os embeddings 2-D a seguir foram criados usando tSNE sobre as representações obtidas pelo modelo GloVe com 30-D.
 - O t-SNE tenta fazer com que as distâncias no embedding 2-D casem com as distâncias originais (em 30-D) tão bem quanto possível
- Você pode brincar com word embeddings aqui:
<https://projector.tensorflow.org>

Visualizando word embeddings



Visualizando word embeddings



Visualizando word embeddings

- Pense sobre embeddings de alta dimensão
 - Maioria dos vetores é praticamente ortogonal (produto interno quase 0)
 - Maior parte dos pontos estão distantes entre si
 - “In a 30-dimensional grocery store, anchovies can be next to fish and next to pizza toppings.” – Geoff Hinton
- **Palavra de cautela:** embeddings 2-D podem enganar, dado que eles não conseguem preservar as distâncias originais de um espaço de alta dimensão (i.e., palavras não-relacionadas podem estar próximas em 2-D, mas distantes em 30-D)

Resolvendo analogias com word embeddings

Resolvendo analogias com word embeddings

	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
Gender	-1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.70	0.69	0.03	-0.02
Food	0.09	0.01	0.02	0.01	0.95	0.97

Man \rightarrow Woman assim como King \rightarrow ?

Analogias

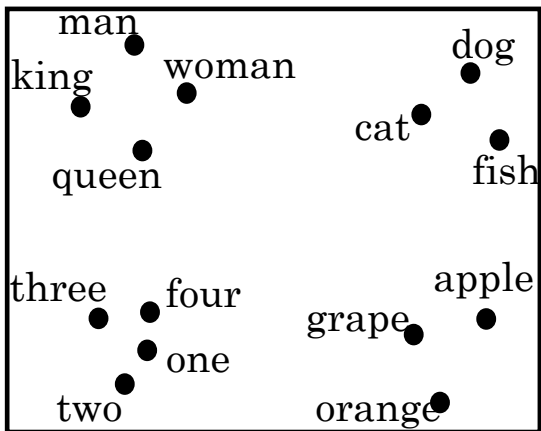
	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
Gender	-1	1	-2	0.97	0.00	0.01
Royal	0.01	0.02	-0.01	0.93	-0.01	0.00
Age	0.03	0.02	0.70	0.69	0.03	-0.02
Food	0.09	0.01	0.02	0.01	0.95	0.97

man \rightarrow woman as king \rightarrow ?
 $e_{\text{man}} - e_{\text{woman}} \approx e_{\text{king}} - e_{\text{?}}$

$$e_{\text{man}} - e_{\text{woman}} = \begin{bmatrix} -2 \\ -0.01 \\ 0.01 \\ 0.08 \end{bmatrix}$$

$$e_{\text{king}} - e_{\text{queen}} = \begin{bmatrix} 2.2 \\ 2.2 \\ 2.2 \\ 2.2 \\ 0 \end{bmatrix}$$

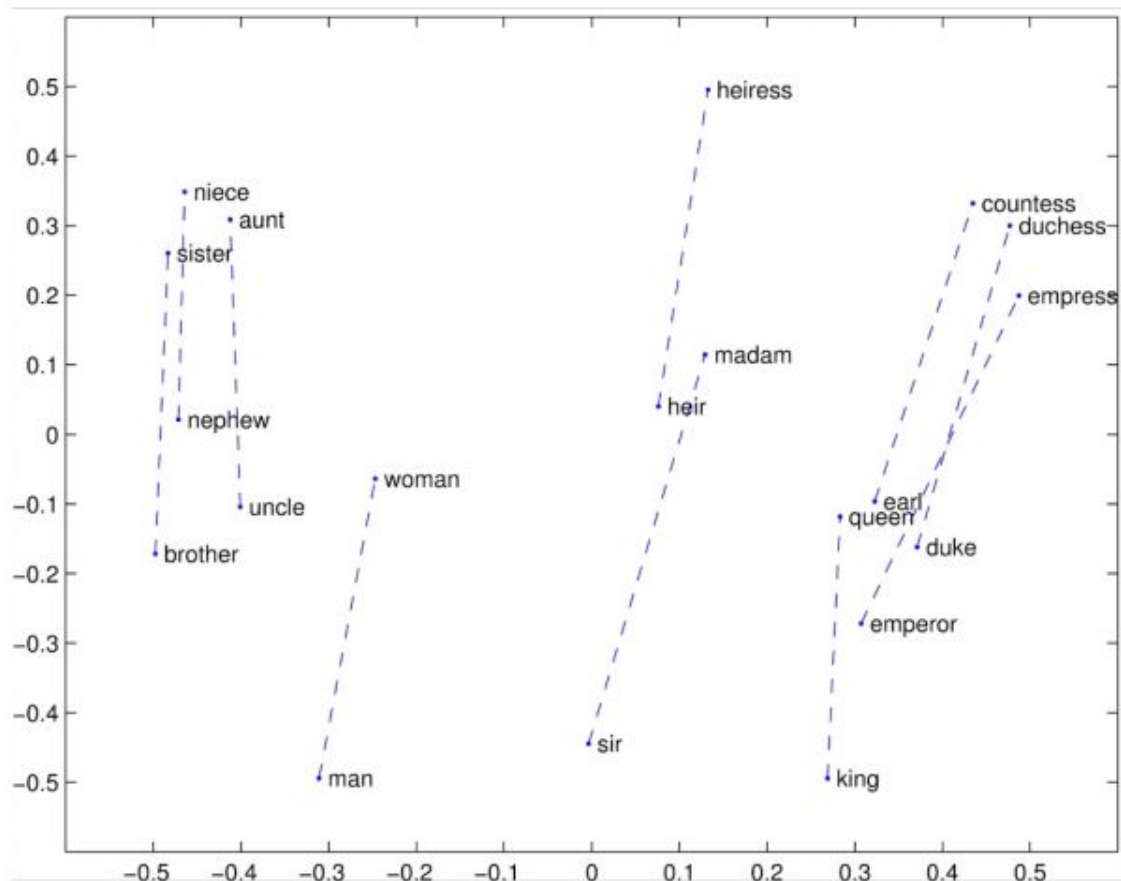
Resolvendo analogias com word embeddings



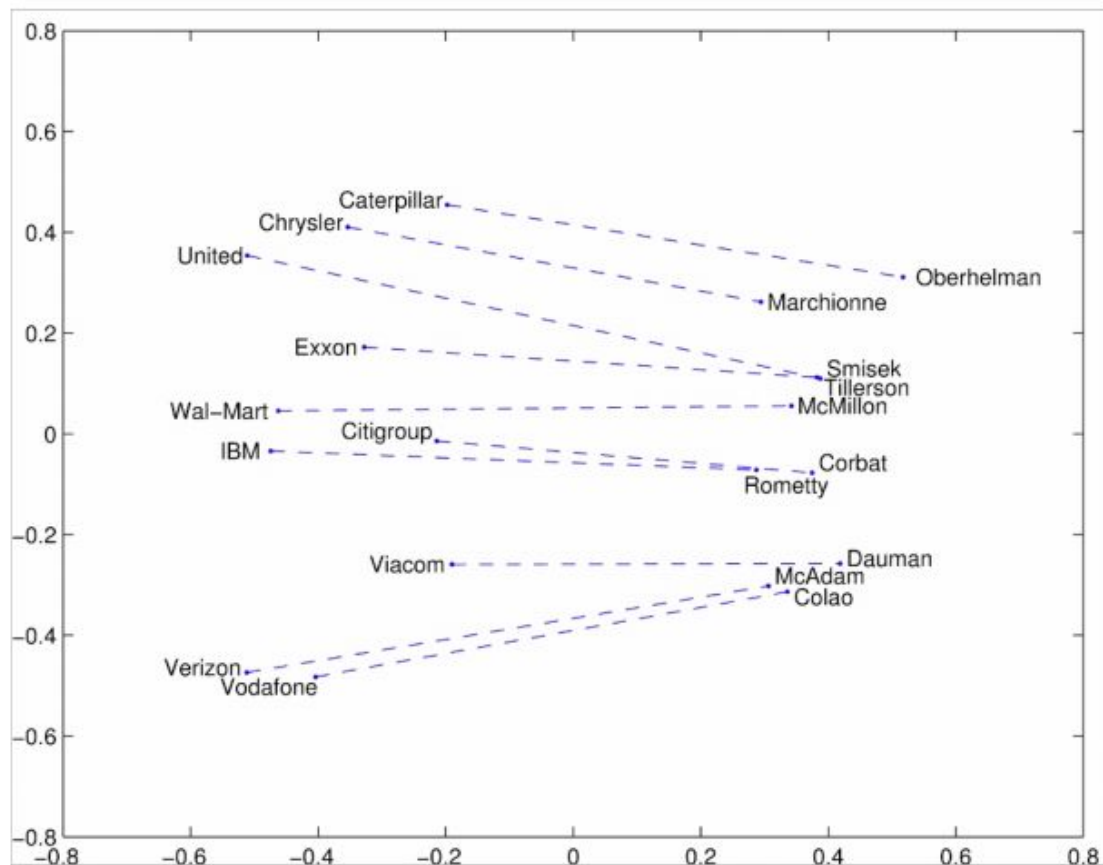
$$\mathbf{r}_{\text{man}} - \mathbf{r}_{\text{woman}} \approx \mathbf{r}_{\text{king}} - \mathbf{r}_{?}$$

Encontre a palavra w : $\text{argmax}_w \text{sim}(\mathbf{r}_w, \mathbf{r}_{\text{king}} - \mathbf{r}_{\text{man}} + \mathbf{r}_{\text{woman}})$

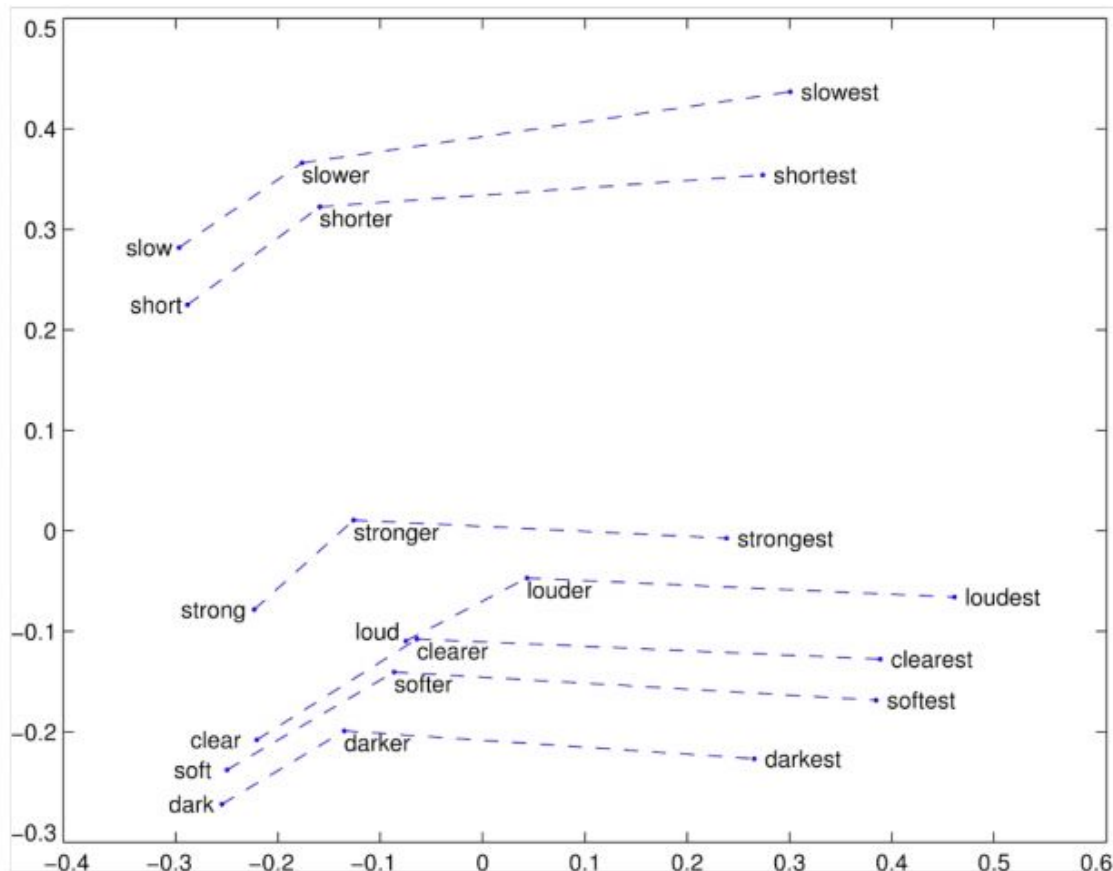
Visualizações do GloVe



Visualizações do GloVe



Visualizações do GloVe



Word Vector Analogies: exemplos semânticos

: city-in-state

Chicago Illinois Houston Texas

Chicago Illinois Philadelphia Pennsylvania

Chicago Illinois Phoenix Arizona

Chicago Illinois Dallas Texas

Chicago Illinois Jacksonville Florida

Chicago Illinois Indianapolis Indiana

Chicago Illinois Austin Texas

Chicago Illinois Detroit Michigan

Chicago Illinois Memphis Tennessee

Chicago Illinois Boston Massachusetts

<http://code.google.com/p/word2vec/source/browse/trunk/questions-words.txt>

Word Vector Analogies: exemplos sintáticos

: gram4-superlative

bad worst big biggest

bad worst bright brightest

bad worst cold coldest

bad worst cool coolest

bad worst dark darkest

bad worst easy easiest

bad worst fast fastest

bad worst good best

bad worst great greatest

<http://code.google.com/p/word2vec/source/browse/trunk/questions-words.txt>

Avaliação em analogias

- Dimensão do embedding:
100, 300, 1000
- Tamanho do conj de treinamento:
1B, 1.5B, 1.6B, 6B, 42B
- Avaliação:
Semântica, Sintática, Média

Model	Dim.	Size	Sem.	Syn.	Tot.
ivLBL	100	1.5B	55.9	50.1	53.2
HPCA	100	1.6B	4.2	16.4	10.8
GloVe	100	1.6B	<u>67.5</u>	<u>54.3</u>	<u>60.3</u>
SG	300	1B	61	61	61
CBOW	300	1.6B	16.1	52.6	36.1
vLBL	300	1.5B	54.2	<u>64.8</u>	60.0
ivLBL	300	1.5B	65.2	63.0	64.0
GloVe	300	1.6B	<u>80.8</u>	61.5	<u>70.3</u>
SVD	300	6B	6.3	8.1	7.3
SVD-S	300	6B	36.7	46.6	42.1
SVD-L	300	6B	56.6	63.0	60.1
CBOW [†]	300	6B	63.6	<u>67.4</u>	65.7
SG [†]	300	6B	73.0	66.0	69.1
GloVe	300	6B	<u>77.4</u>	67.0	<u>71.7</u>
CBOW	1000	6B	57.3	68.9	63.7
SG	1000	6B	66.1	65.1	65.6
SVD-L	300	42B	38.4	58.2	49.2
GloVe	300	42B	<u>81.9</u>	<u>69.3</u>	<u>75.0</u>

GloVe (opcional)

Global **Ve**ctors for
word representation

GloVe (global vectors for word representation)

- Defina $q_{ij} = \frac{\exp(\mathbf{u}_j^\top \mathbf{v}_i)}{\sum_{k \in \mathcal{V}} \exp(\mathbf{u}_k^\top \mathbf{v}_i)}$
- Reescreva a função negative log-likelihood do skip-gram

$$-\log \left[\prod_{t=1}^T \prod_{-m \leq j \leq m, j \neq 0} \mathbb{P}(w^{(t+j)} \mid w^{(t)}) \right]$$

como

$$- \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} x_{ij} \log q_{ij}$$

com os contadores x_{ij} definidos de forma apropriada.

GloVe (global vectors for word representation)

I want a glass of orange juice to go along with my cereal.

$$x_{ij} = \# \text{ vezes que } w_i \text{ aparece no contexto de } w_j$$

(contexto)
(target)

- Contexto pode ser definido de várias formas.
 - Quando contexto e target são definidos simetricamente, temos:

$$X_{ij} = X_{ji}$$

- Outro ex.: contexto definido como **qualquer palavra** dentro de uma janela de tamanho $[-5,+5]$ ou $[-10,+10]$

GloVe (global vectors for word representation)

- Em seguida, faça

$$-\sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} x_{ij} \log q_{ij} = -\sum_{i \in \mathcal{V}} x_i \sum_{j \in \mathcal{V}} p_{ij} \log q_{ij}$$

Com $x_i = \sum_{j \in \mathcal{V}} x_{ij}$, $p_{ij} = x_{ij}/x_i$

- Isso nos dá uma soma ponderada da cross-entropy para palavras no contexto de w_i , cujo peso x_i é o número de vezes em que w_i é target word
 - Embora muito usada, cross-entropy nem sempre é uma boa escolha
 - Custo para fazer com que q_{ij} seja uma probabilidade válida é alto
 - Previsões feitas a partir da distribuição condicional envolvendo palavras incomuns pode ser muito ruim

GloVe (global vectors for word representation)

- Em seguida, faça

$$-\sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} x_{ij} \log q_{ij} = -\sum_{i \in \mathcal{V}} x_i \sum_{j \in \mathcal{V}} p_{ij} \log q_{ij}$$

Com $x_i = \sum_{j \in \mathcal{V}} x_{ij}$, $p_{ij} = x_{ij} / x_i$

- Substitua a cross-entropy por log square loss

$$\sum_{j \in \mathcal{V}} p_{ij} \log q_{ij} \rightarrow \sum_{j \in \mathcal{V}} (\log p_{ij} - \log q'_{ij})^2$$

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

Com algum q_{ij} fácil de ser calculado

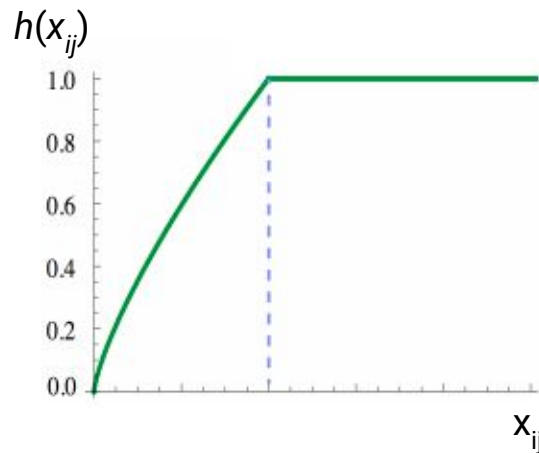
- Adicione um termo de bias para a palavra central e palavras do contexto
- Substitua os pesos x_i por uma função $h(x_i)$ monótona crescente em $[0,1]$

Modelo

Minimizar

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

- $h(x_{ij})$ tem como papel:
 - Eliminar termos do somatório para pares (i, j) que não aparecem juntos, i.e. $x_{ij} = 0 \Rightarrow f(x_{ij}) = 0$. Assume-se que $0 \log 0 = 0$.
 - Compensar termos que aparecem com muita/pouca frequência, i.e., $f(x_{ij})$ assume valores menores para palavras extremamente frequentes



Note que u_j e v_i são simétricos. Representação final de w é a média

$$e_w^{(\text{final})} = \frac{\mathbf{u}_w + \mathbf{v}_w}{2}$$

Entendendo GloVe a partir de razões entre probabilidades condicionais

Intuição: razão entre probabilidades de co-ocorrências pode codificar componentes do significado

	$x = \text{solid}$	$x = \text{gas}$	$x = \text{water}$	$x = \text{random}$
$P(x \text{ice})$	large	small	large	small
$P(x \text{steam})$	small	large	large	small
$\frac{P(x \text{ice})}{P(x \text{steam})}$	large	small	~ 1	~ 1

Entendendo GloVe a partir de razões entre probabilidades condicionais

Intuição: razão entre probabilidades de co-ocorrências pode codificar componentes do significado

	$x = \text{solid}$	$x = \text{gas}$	$x = \text{water}$	$x = \text{fashion}$
$P(x \text{ice})$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(x \text{steam})$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$\frac{P(x \text{ice})}{P(x \text{steam})}$	8.9	8.5×10^{-2}	1.36	0.96

Codificando significado com diferenças entre vetores

Q: Como capturar a razão entre probabilidades de co-ocorrência como componentes lineares em um espaço vetorial?

R: Queremos calcular

$$\log \frac{P(w_i|w_j)}{P(w_i|w_k)}$$

No modelo log-bilinear, temos $\log P(w_i|w_j) = \mathbf{u}_j^\top \mathbf{v}_i$

Portanto,
$$\log \frac{P(w_i|w_j)}{P(w_i|w_k)} = \mathbf{u}_j^\top \mathbf{v}_i - \mathbf{u}_k^\top \mathbf{v}_i = (\mathbf{u}_j - \mathbf{u}_k)^\top \mathbf{v}_i$$

Escolhendo uma função

$$f(\mathbf{u}_j, \mathbf{u}_k, \mathbf{v}_i) = f((\mathbf{u}_j - \mathbf{u}_k)^\top \mathbf{v}_i) \approx \frac{p_{ij}}{p_{ik}}$$


Codificando significado com diferenças entre vetores

Q: Como capturar a razão entre probabilidades de co-ocorrência como componentes lineares em um espaço vetorial?

Escolhendo $f(x) = \exp(x)$, temos

$$\exp(\mathbf{u}_j^\top \mathbf{v}_i) \approx \alpha p_{ij}$$

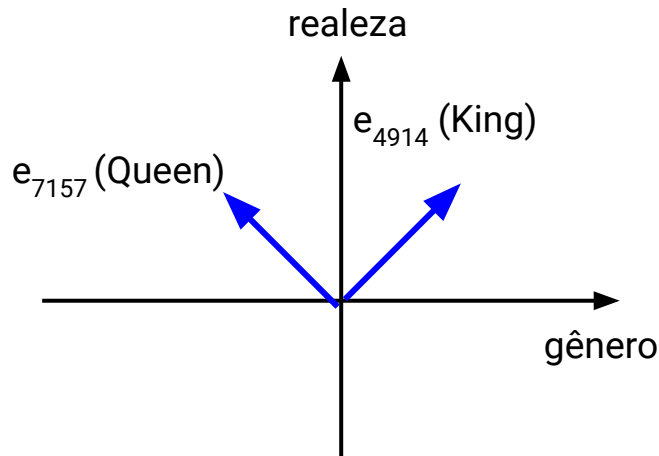
$$\mathbf{u}_j^\top \mathbf{v}_i \approx \log \alpha + \log x_{ij} - \log x_i$$


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

Visão featurizada dos word embeddings

Como gostaríamos que os embeddings fossem

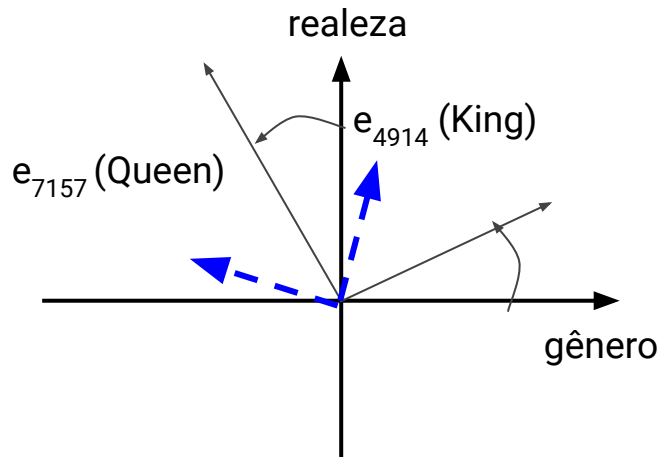
Possível interpretação	Man (5391)	Woman (9853)	King (4914)	Queen (7157)
Gender	-1	1	-0.95	0.97
Royal	0.01	0.02	0.93	0.95
Age	0.03	0.02	0.70	0.69
Food	0.09	0.01	0.02	0.01



Visão featurizada dos word embeddings

Como gostaríamos que os embeddings fossem

Possível interpretação	Man (5391)	Woman (9853)	King (4914)	Queen (7157)
Gender	-1	1	-0.95	0.97
Royal	0.01	0.02	0.93	0.95
Age	0.03	0.02	0.70	0.69
Food	0.09	0.01	0.02	0.01



Visão featurizada dos word embeddings

Na prática, não há como garantir que as dimensões tenham interpretação clara. Qualquer rotação dá origem ao mesmo mínimo:

$$\text{minimizar } \sum_{i=1}^{10000} \sum_{j=1}^{10000} h(x_{ij}) (\mathbf{u}_j^\top \mathbf{v}_i + b_i + c_j - \log x_{ij})^2$$
$$\underbrace{(\mathbf{A}\mathbf{u}_j)^\top (\mathbf{A}\mathbf{v}_i)} = \mathbf{u}_j^\top \mathbf{A}^\top \mathbf{A} \mathbf{v}_i = \mathbf{u}_j^\top \mathbf{v}_i$$

Apesar disso a matemática do paralelogramo ainda funciona!

Videoaulas

<https://youtu.be/QuELiw8tbx8>

<https://www.youtube.com/watch?v=BwmddtPFWtA>

<https://www.youtube.com/watch?v=LHXXI4-IEns>

https://www.youtube.com/watch?v=Keqep_PKrY8&t=553s

<https://www.youtube.com/watch?v=8HyCNIVRbSU>

<https://www.youtube.com/watch?v=94hG00EJFNo>