Semántica Distribucional

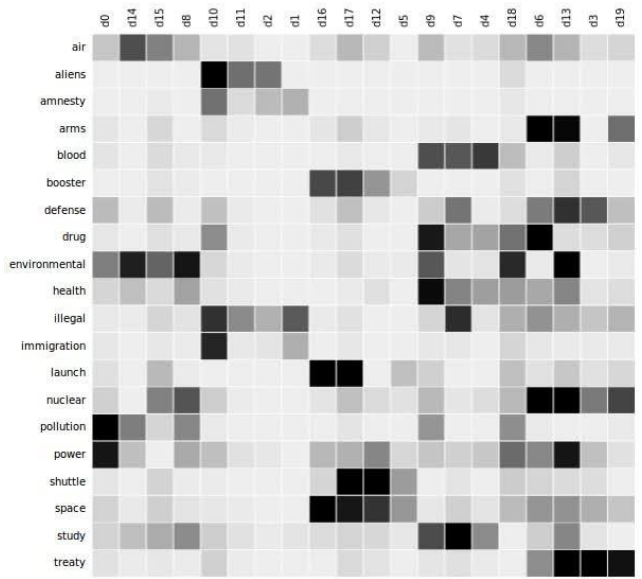
Word Embeddings

Semántica Distribucional

- Busca cuantificar y categorizar la similitud entre elementos linguisticos: palabras, frases, oraciones, textos.
- Se basa en la llamada hipótesis distribucional: items linguisticos con distribuciones similares, tienen significados similares.
- Para el caso de palabras: palabras que ocurren dentro del mismo contexto tienen el mismo significado. “A word is characterized by the company it keeps".

Latent Semantical Analysis

- Se construye una matriz término-documento
- Cada fila representa un documento y cada columna un término dentro del vocabulario.
- La intersección fila columna es una medida de la presencia de la palabra en el documento.
- Dicha medida puede ser la cantidad de veces que ocurre, o el tf-idf:
-



$$\mathrm{tf}(t,d) = \frac{\mathrm{f}(t,d)}{\max\{\mathrm{f}(t,d) : t \in d\}} \qquad \mathrm{idf}(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

$$\mathrm{tfidf}(t,d,D) = \mathrm{tf}(t,d) \times \mathrm{idf}(t,D)$$

Latent Semantical Analysis

- Una vez construída la matriz, cada fila representa a cada documento, con un número de dimensiones constantes.
- Para bajar la dimensionalidad se aplica TruncatedSVD.

  De esta manera se generan representaciones de baja dimensionalidad de documentos.

Matriz de coocurrencia palabra-palabra

- Se define la coocurrencia según:
  - Ventana de Contexto
  - Oración
  - Párrafo
  - Documento
- Luego se aplica Truncated SVD.

De esta manera tenemos representaciones vectoriales de palabra de baja dimensionalidad.

	I	love	Programming	Math	tolerate	Biology	.
I	0	2	0	0	1	0	2
love	2	0	1	1	0	0	0
Programming	0	1	0	0	0	0	1
Math	0	1	0	0	0	0	1
tolerate	1	0	0	0	0	1	0
Biology	0	0	0	0	1	0	1
.	1	0	1	1	0	1	0

Matriz de PMI (Pointwise Mutual Information)

$$\operatorname{pmi}(x;y) \equiv \log \frac{p(x,y)}{p(x)p(y)} = \log \frac{p(x|y)}{p(x)} = \log \frac{p(y|x)}{p(y)}$$

- Es otra forma de medir coocurrencia entre palabras.
- En teoría de la información, la PMI es la información que me dá sobre la ocurrencia de una palabra, que haya ocurrido otra, con respecto a palabras que se presentan independientemente.
- los valores negativos de PMI en general no aportan mucho y suelen haber mucha varianza en su estimación, por lo que los valores de PMI menores que cero suelen ser desestimados. A esta matriz se la llama PPMI (Positive PMI).
- Sobre la representación obtenida, se suelen cumplir los test de analogía.
- Lo normal es aplicar SVD a la matriz de PPMI para generar una representación de palabra de baja dimensionalidad.

Tests de Analogía

- Son tests que miden qué tan estructurada la representación de las palabras.
- Rey (w1) -> Reina(w2), Príncipe (w3) -> Princesa (w4)
- cos_sim(w2 - w1 + w3, wi) es máximo cuando wi = w4?
- Mencionar la mentira de los tests de analogía
- Semántico / Morfológico

Glove

- Involucran la información que brinda el cociente de la coocurrencia de dos palabras:

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

Glove

La función de costo incorpora esta información de manera de mejorar los tests de analogía. Para ver el detalle se recomienda ver el paper o el siguiente blog:

https://towardsdatascience.com/light-on-math-ml-intuitive-guide-to-understanding-glove-embeddings-b13b4f19c010

Paper:

https://nlp.stanford.edu/pubs/glove.pdf

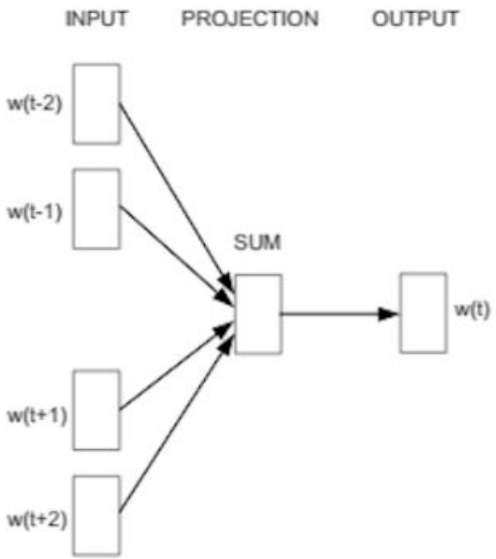
Glove

- Función de costo:

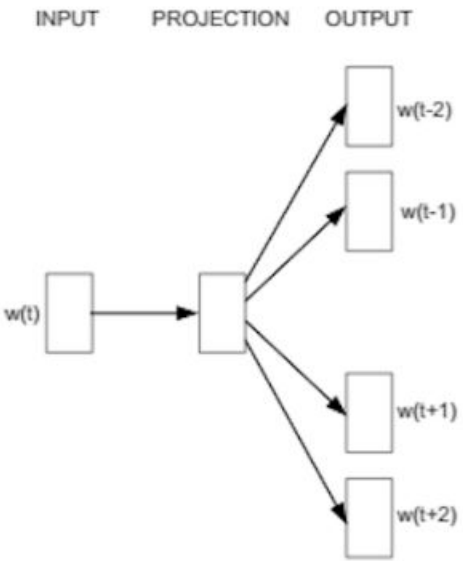
$$J(\theta) = \frac{1}{2} \sum_{i,j=1}^{W} f(P_{ij})(u_i^T v_j - \log P_{ij})^2$$

- la función f es una medida de cuán vinculados están los términos ij, según otras palabras que actúen como elementos en común. Se calcula a partir de los cocientes de probabilidad vistos anteriormente.

Word2Vec



CBOW



Skip-gram

- CBOW: a partir de las palabras de contexto, predice la probabilidad de la palabra central (target).
- SKIP-GRAM: a partir de la palabra central predice la probabilidad de las palabras de contexto.

Word2Vec

- CBOW loss:

$$J_\theta = \frac{1}{T}\sum_{t=1}^{T} \log p(w_t \mid w_{t-n}, \cdots, w_{t-1}, w_{t+1}, \cdots, w_{t+n})$$

- SKIP-GRAM Loss:

$$J_\theta = \frac{1}{T}\sum_{t=1}^{T} \sum_{-n \le j \le n, \neq 0} \log p(w_{t+j} \mid w_t)$$

Word2vec

- Skipgram model:

$$p(w_{t+j} \mid w_t) = \frac{\exp(h^\top v'_{w_{t+j}})}{\sum_{w_i \in V} \exp(h^\top v'_{w_i})}$$

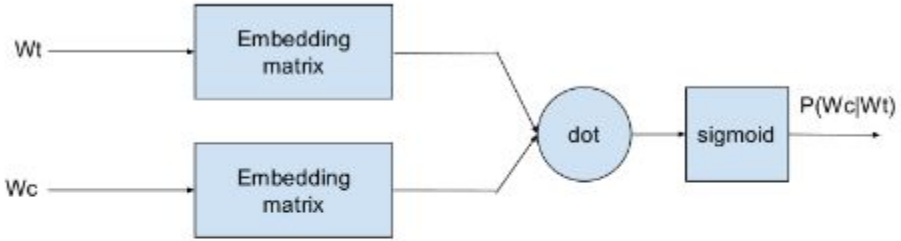
- ¿Qué pasa cuando se calcula el gradiente con el denominador?
- Hay una batería de soluciones a este problema:

http://ruder.io/word-embeddings-softmax/index.html

Optimizaciones de la Softmax

APPROACH	SPEED-UP FACTOR	DURING TRAINING?	DURING TESTING?	PERFORMANCE (SMALL VOCAB)	PERFORMANCE (LARGE VOCAB)	PROPORTION OF PARAMETERS
Softmax	1x	-	-	very good	very poor	100%
Hierarchical Softmax	25x (50-100x)	X	-	very poor	very good	100%
Differentiated Softmax	2x	X	X	very good	very good	< 100%
CNN-Softmax	-	X	-	-	bad - good	30%
Importance Sampling	(19x)	X	-	-	-	100%
Adaptive Importance Sampling	(100x)	X	-	-	-	100%
Target Sampling	2x	X	-	good	bad	100%
Noise Contrastive Estimation	8x (45x)	X	-	very bad	very bad	100%
Negative Sampling	(50-100x)	X	-	-	-	100%
Self-Normalisation	(15x)	X	-	-	-	100%
Infrequent Normalisation	6x (10x)	X	-	very good	good	100%

Word2Vec (Negative Sampling)



Cost Function:

$$J'_\theta = -\sum_{w_i \in V} [\log \frac{1}{1+\exp(-v_{w_t} v'_{w_{t+j}})} + \sum_{i=1}^{k} \log \frac{1}{1+\exp(-v_{w_t} v'_{w_{ij}})}]$$

- El contexto y la palabra central se samplean con la distribución de ocurrencia de cada palabra, mas smoothing de parámetro alfa.

$$P_\alpha(w_i) \propto P(w_i)^\alpha$$

Particularidades de Word2Vec

- Subsampleo de palabras frecuentes:

$$p = 1 - \sqrt{\frac{t}{f}}$$

- Levy and Goldberg (2014) demostraron que W2V converge a una factorización de la matriz de PMI, con las siguiente salvedades:
  - Shifted PPMI (SPPMI):
  -
$$SPPMI(w, c) = \max(PMI(w, c) - \log k, 0)$$
  -
  - Context distribution sampling:

$$PMI(w, c) = \log \frac{p(w, c)}{p(w)p_\alpha(c)} \text{ where } p_\alpha(c) = \frac{f(c)^\alpha}{\sum_c f(c)^\alpha} \text{ and } f(x) \text{ is the frequency of word } x$$

Particularidades de Word2Vec

- Dynamical Context Window:
  - Se suelen considerar palabras mas cercanas al target mas importantes que las mas lejanas.
  - Para hacer esto, habría que asignar un coeficiente a cada una de las palabras.
  - Otra opción es samplear el tamaño de ventana entre 1 y k para cada palabra.

Blog super recomendado:

http://ruder.io/word-embeddings-1/index.html

FastText

Paper: https://arxiv.org/abs/1607.04606

- Es muy similar a Word2Vec pero trabaja no solo a nivel de palabra, sino a nivel de ngrams de letra.
- De esta forma se pueden obtener los word vectors de palabras desconocidas. Si n=3:
  - <matter> = ”<ma”, “mat”,”att”,”tte”,”ter”>
- El embedding propuesto para la palabra se obtiene de sumar el embedding para cada uno de los ngramas.
- En la implementación original se utiliza n desde 3 a 6.

FastText

Suppose that you are given a dictionary of n-grams of size G. Given a word w, let us denote by $\mathcal{G}_w \subset \{1, \ldots, G\}$ the set of n-grams appearing in w. We associate a vector representation $\mathbf{z}_g$ to each n-gram g. We represent a word by the sum of the vector representations of its n-grams. We thus obtain the scoring function:

$$s(w, c) = \sum_{g \in \mathcal{G}_w} \mathbf{z}_g^\top \mathbf{v}_c.$$

Contextual Embeddings

- Son embeddings que tienen en cuenta no solo el significado de la palabra de manera general, sino su acepción dentro de un contexto.
- Consideran el problema de desambiguación para la representación de palabras.