Estudio del grupo de transformaciones del espacio de word embeddings

May 21, 2021

Objetivo

Este proyecto tiene como objetivo estudiar el espacio de word embeddings resultantes de algoritmos como word2vec. En lugar de estudiar directamente este espacio, enfocaremos nuestra atención en el grupo de transformaciones que actúan en este espacio de manera *interesante*.

1 Antecedentes

El algoritmo `word2vec` ha sido muy exitoso en producir representaciones incrustadas de palabras las cuales capturan similitudes semánticas y analogías con gran éxito. Las similitudes semánticas quedan descritas en función de la cercanía medida con la métrica angular. Por otro lado, las analogías quedan descritas en términos de traslaciones. Este proyecto busca estudiar el espacio de embeddings de `word2vec` por medio del estudio y descripción de la acción de estas traslaciones en el espacio de embbedings.

2 La acción de las traslaciones en el espacio de embeddings

Como se mencionó anteriormente, uno de los logros del algoritmo `word2vec` es el cálculo de analogías. Las analogías son relaciones entre palabras, de la forma:

$$\begin{aligned} \text{Rey} : \text{Reyna} &:: \text{Hombre} : \text{Mujer} \\ \text{España} : \text{Madrid} &:: \text{Noruega} : \text{Oslo} \end{aligned}$$

Consideraremos un corpus de word embeddings

$$\mathbf{W} = \{w_1, ..., w_N\},$$

de tal forma que $\|w_k\| = 1$ para $k = 1, ..., N$. Aquí, $\|z\|$ denota la norma euclidiana del vector z. Definimos la similitud coseno como

$$\text{sim}(a, b) = \frac{a \cdot b}{\|a\|\|b\|},$$

donde $a \cdot b$ denota al producto punto de a y b.

Usando word embeddings calculados por `word2vec`, una analogía de la forma $a : b :: c : d$ satisface la relación

$$w_a - w_b \approx w_c - w_d, \tag{2.1}$$

donde w_a es el word embedding de la palabra a y así sucesivamente. Si alguna de las palabras es desconocida, por ejemplo la palabra a, la relación (2.1) se convierte en

$$w_a \approx w_b + (w_c - w_d), \tag{2.2}$$

Es decir, la relación semántica entre las palabras c y d define la traslación por el vector $v_{c,d} = w_c - w_d$ que, al aplicarla al embedding de una palabra b, resuelve la analogía $x : b :: c : d$. Esto se representa graficamente en la figura 1.

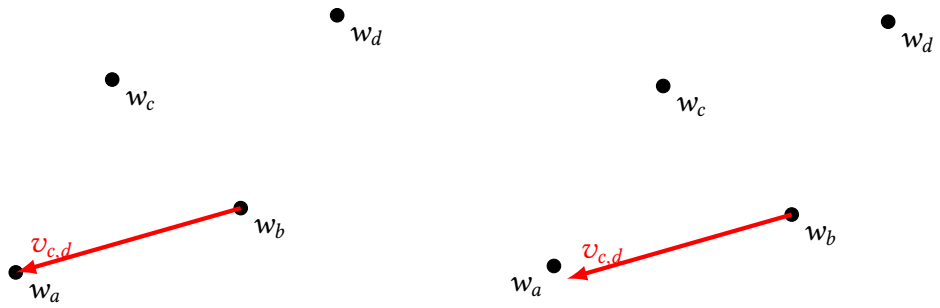

Figure 1: La acción de una traslación en un punto. A la izquierda se satisface estrictamente $w_a = w_b + (w_c - w_d)$. A la derecha, la relación es una aproximación, como sucede en realidad.

Ahora, en lugar de considerar la traslación $v_{c,d}$ aplicada al punto w_b, aplicamos la traslación a todos los puntos del corpus, como en la figura 2. Al aplicar la traslación, cada embedding w se *mueve* al punto $w + v_{c,d}$, si la palabra que representa el vector w forma parte de alguna analogía, este resultado de la traslación debe quedar cerca de algún embedding. Este caso ocurre en la figura 2 con el punto w_b se mueve y queda muy cerca de w_a, es decir, con la pareja $w_b \mapsto w_a$. El mismo fenómeno ocurre con las parejas de puntos $w_d \mapsto w_c$, aunque, ya que $v_{c,d}$ está definido en términos de w_d y w_c, cada traslación siempre tiene una pareja de puntos que satisface la propiedad.

Definición 1 — Para $\alpha > 0$, si una traslación $v_{i,j} = w_i - w_j$ satisface que hay dos embeddings w, z, distintos de w_i y w_j, tales que

$$\text{sim}\big(z, w + v_{i,j}\big) > \alpha,$$

decimos que la traslación $v_{i,j}$ es α-compatible con el corpus $\mathbf{W}$.

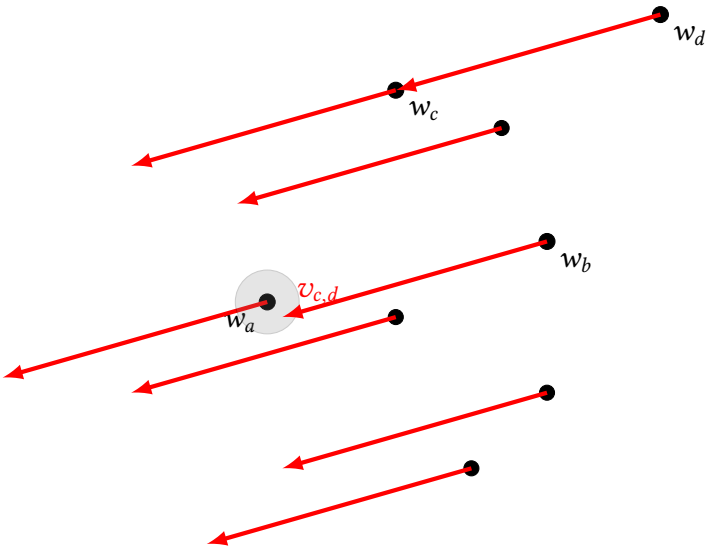

Figure 2: La acción de la traslación $v_{c,d}$ en el espacio de word embeddings. Los embeddings w_a y w_b hacen de esta traslación, una traslación compatible con el corpus.

El ejemplo de la figura 2 muestra que la traslación $v_{c,d}$ es α-compatible con el corpus con un α apropiado. Los puntos que satisfacen la condición son w_a y w_b.

Por otro lado, en la figura 3 vemos la situación donde la acción de una traslación no es compatible con el conjunto de puntos.

Como último ejemplo, en la figura 4, vemos el caso de un corpus que tiene dos traslaciones 1-compatibles con el grado más alto de compatibilidad. Es un ejemplo ideal que muestra un corpus con una estructura muy sencilla.

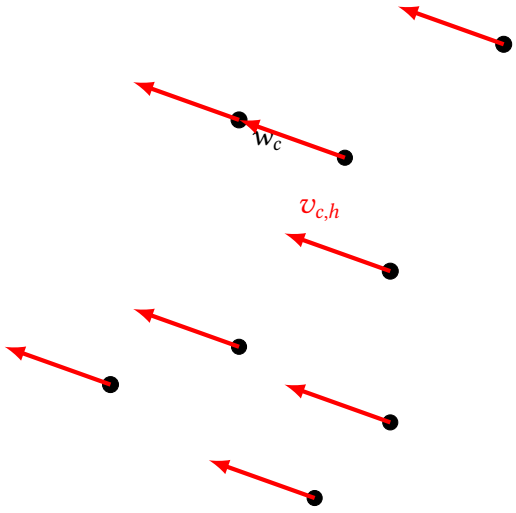

Figure 3: La acción de la traslación $v_{c,d}$ en el espacio de word embeddings.

El primer objetivo es identificar para cuántas y cuáles traslaciones son α-compatibles con el corpus para algún valor de α relativamente alto.

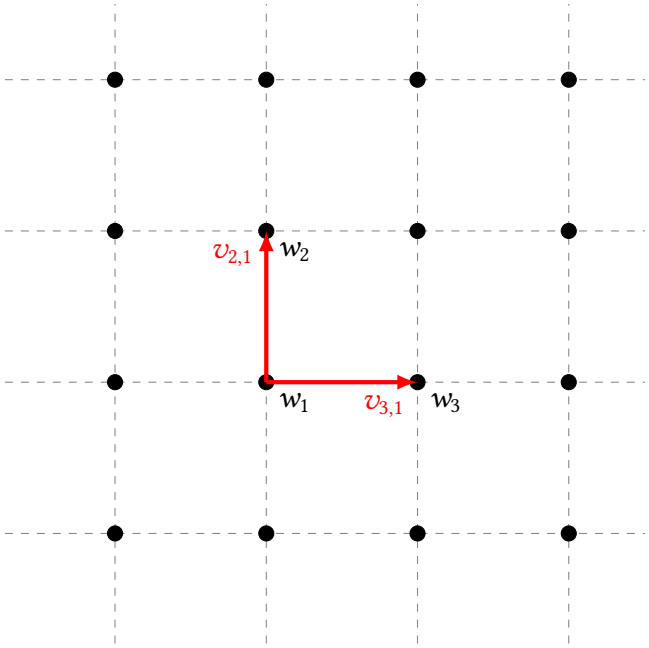

Figure 4: Un espacio de word embeddings *sencillo*, es generado por dos traslaciones $v_{3,1}$ y $v_{2,1}$.

3 Tareas a realizar

3.1 Tarea 1: Pasos a seguir

1. Comenzar con el elemento w_i (empezamos con $i = 0$).
2. Tomar un elemento w_j, con $j \neq i$.
3. Definimos y tomamos $v_{i,j} = w_i - w_j$.
4. Contar cuántos elementos $w \in \mathbf{W}$ están *cerca* de $z + v_{i,j}$ para cada $z \in \mathbf{W}$. Es decir, tomamos $\alpha = 0.8$ y definimos el conjunto $\Psi_{w,v_{i,j}}$ como la lista que regresa[1] la función

$$\texttt{most\_similar}(w + v_{i,j},\ \text{topn} = 10),$$

esta función está en la clase `Word2Vec` de `gensim`. Ahora contamos cuántas de estas palabras $z \in \Psi_{w,v_{i,j}}$ satisfacen $\text{sim}\big(z, w + v_{i,j}\big) \geq \alpha$. Llamamos a este número $n_{i,j}^{(w)}$ y definimos

$$n_{i,j} = \sum_{w \in \mathbf{W}} n_{i,j}^{(w)} \text{sim}\big(z, w + v_{i,j}\big)$$

Repetir lo anterior para todos los i y $j \neq i$ y reportar los resultados en una matriz $X = (n_{i,j})$, es decir

$$X = \begin{pmatrix} n_{1,1} & \cdots & n_{1,N} \\ \cdots & \cdots & \cdots \\ n_{N,1} & \cdots & n_{N,N} \end{pmatrix}$$

Guardar esta matriz en formato *npy* con numpy.

[1] En realidad, la función regresa una lista ordenada de tuplas. Cada tupla contiene la palabra y la similaridad con el vector dado.

3.2 Tarea 2: Pasos a seguir

Esta tarea consiste en determinar el rango (o forma de Jordan) de la matriz $A^T A$ donde

$$A = \begin{pmatrix} v_{1,1} \\ \cdots \\ v_{1,N} \\ v_{2,3} \\ \cdots \\ v_{2,N} \\ \cdots \\ v_{N-1,N} \end{pmatrix}$$