Supongamos `word2vec` para el caso Skip-gram, es decir, que queremos estimar la probabilidad del contexto o a partir de una palabra central c:

$$P(w_o|w_c) = \frac{e^{u_o^T v_c}}{\sum_{i=1}^{|V|} e^{u_i^T v_c}} \tag{1}$$

donde u_o es el word vector de la palabra del contexto (*outsider*) y v_c es el de la palabra central. Recordemos que la red de este modelo tiene una matriz $U \in \mathbb{R}^{|V| \times d}$ cuyas filas son los vectores $u_1, \ldots, u_{|V|} \in \mathbb{R}^d$, una matriz $V \in \mathbb{R}^{d \times |V|}$ cuyas columnas son los vectores $v_1, \ldots, v_{|V|} \in \mathbb{R}^d$ y una activación softmax. Si estimo esta probabilidad por ML o por Cross-Entropy (que es lo mismo), me queda una función objetivo

$$J_t(u_o, v_c) = \log\left(P(w_o|w_c)\right) = \log\left(u_o^T v_c\right) - \log\left(\sum_{i=1}^{|V|} e^{u_i^T v_c}\right) \tag{2}$$

Esto implicaría que para cada muestra t del corpus de entrenamiento se tengan que calcular todas las $e^{u_i^T v_c} \;\; \forall\, i = 1, \ldots |V|$, lo cual es muy costoso. Entonces lo que se hace es tirar esta función de costo a la basura y definir una nueva que me permita no tener que pasar por todo este cálculo, pero que me sirva para entrenar *word embeddings* igual. Antes de pasar a la función es importante hacer énfasis en esto último: ya no estoy usando la ecuación 2 como función objetivo, pero voy a seguir considerando que quiero estimar el contexto a partir de una palabra central.

Definimos una nueva forma de entrenar *word embeddings* a partir de la probabilidad $P(h|w_o, w_c)$ de que el conjunto (w_o, w_c) conformado por una palabra w_o del contexto y una palabra w_c central pertenezcan al corpus de entrenamiento D:

$P\left((w_o, w_c) \in D\right) = P(h = 1|w_o, w_c) =$ La palabra w_o apareció en el contexto de w_c en mi corpus de entrenamiento

$P\left((w_o, w_c) \notin D\right) = P(h = 0|w_o, w_c) =$ La palabra w_o no apareció en el contexto de w_c en mi corpus de entrenamiento

Por ejemplo, si yo entreno con un corpus $D = \{Here,\ comes,\ the,\ sun\}$ y considero una ventana de contexto de tamaño 1, el par $(Here, comes)$ pertenece al corpus, pero el par $(Here, the)$ no.

También definimos la forma de esta probabilidad como el producto de las matrices U y V definidas igual que antes, pero seguida de una activación sigmoidea:

$$P(h = 1|w_o, w_c) = \sigma(u_o^T v_c)$$
$$P(h = 0|w_o, w_c) = 1 - \sigma(u_o^T v_c)$$

Además, hago una variante extra. En lugar de entrenar sobre el conjunto D que entrenaba antes, voy a entrenar sobre el conjunto de todas las posibles combinaciones de pares ordenados (w_o, w_c) con $w_o, w_c \in V$. Este conjunto se encuentra conformado por la suma disjunta del conjunto D y $\tilde{D}$ (su complemento). De esta forma, voy a pedir que se maximice la probabilidad de que salgan los pares (w_o, w_c) que aparecieron en el corpus y de que, a su vez, se maximice la probabilidad de que no salgan los pares (w_o, w_c) que no aparecieron en el corpus (es decir, que aparecieron en $\tilde{D}$). Esto capaz se entiende mejor observando la definición de nuestra nueva función de costo:

$$\begin{aligned}\theta &= \underset{\theta}{\text{argmax}} \prod_{(w_o,w_c)\in D} P(h = 1|w_o, w_c) \prod_{(w_o,w_c)\in \tilde{D}} P(h = 0|w_o, w_c) \\ &= \underset{\theta}{\text{argmax}} \prod_{(w_o,w_c)\in D} \sigma(u_o^T v_c) \prod_{(w_o,w_c)\in \tilde{D}} \big(1 - \sigma(u_o^T v_c)\big)\end{aligned}$$

Desarrollando esta ecuación, se llega a lo siguiente:

$$\theta = \underset{\theta}{\text{argmin}} \; - \sum_{(w_o,w_c)\in D} \log \frac{1}{1 + e^{-u_o^T v_c}} \; - \sum_{(w_o,w_c)\in \tilde{D}} \log \frac{1}{1 + e^{u_o^T v_c}} \tag{3}$$

Y acá viene el truco. El conjunto $\tilde{D}$ en realidad es un conjunto de las muestras que no aparecieron en el corpus (es decir, "muestras negativas"), por lo que debería contener pares de palabras que no vienen juntas frecuentemente. La idea es que si yo puedo encontrar una probabilidad $P_n(u)$ tal que

$$\sum_{(w_o,w_c)\in \tilde{D}} \log \frac{1}{1 + e^{u_o^T v_c}} \approx \sum_{(w_o,w_c)\in D} \mathbb{E}_{k\sim P_n(k)} \left[\log \frac{1}{1 + e^{u_k^T v_c}}\right] \tag{4}$$

entonces la expresión del costo puede reescribirse como

$$\theta = \underset{\theta}{\text{argmin}} \; - \sum_{(w_o,w_c)\in D} \left(\log \frac{1}{1 + e^{-u_o^T v_c}} \; - \mathbb{E}_{k\sim P_n(k)} \left[\log \frac{1}{1 + e^{u_k^T v_c}}\right]\right) \tag{5}$$

resultando en una función objetivo

$$J_t(u_o, v_c) = -\log \frac{1}{1 + e^{-u_o^T v_c}} - \mathbb{E}_{k \sim P_n(k)} \left[\log \frac{1}{1 + e^{u_k^T v_c}} \right] \tag{6}$$

que es la del paper de Mikolov. En realidad, la del paper de Mikolov es un poco distinta, pero se refiere a lo que acabo de escribir. En el paper se habla de tomar como objetivo

$$J_t(u_o, v_c) = -\log \frac{1}{1 + e^{-u_o^T v_c}} - \sum_{k=1}^{K} \mathbb{E}_{k \sim P_n(k)} \left[\log \frac{1}{1 + e^{u_k^T v_c}} \right] \tag{7}$$

pero lo único que quiere decir es que las muestras $u_1, \ldots, u_K$ se tomen siguiendo una distribución $P_n(k)$. Fijate que el número K representa el número de muestras negativas que hay que tomar para estimar esta probabilidad y que también es un hiperparámetro. El paper muestra los resultados para algunos valores, que en general son mucho más bajos que el tamaño del vocabulario.

Ahora, ¿cómo defino la probabilidad $P_n(k)$? Bueno, esto (por lo que vi) es absolutamente empírico. Ellos lo que dicen es que la puedo definir en función de la probabilidad de unigramas $U(k)$ como

$$P_n(k) = \frac{U(k)^{3/4}}{\alpha} \tag{8}$$

con α un factor de normalización. Yo entiendo que el sentido de este paso es que vos podés definir una secuencia de palabras que no tienen mucha correlación entre sí muestreando prácticamente al azar (o digamos, en función de la probabilidad de unigramas, pero sin entrenar un modelo de lenguaje) las palabras de tu vocabulario V. Lo que tiene de muy positivo este método es que conseguís un costo en función de la palabra central y el contexto, en lugar de la palabra central y todo el vocabulario. El valor "3/4" es simplemente un número que probaron y que les funcionó, pero que ayuda a muestrear palabras del vocabulario con poca frecuencia. Sería, creo yo, una forma de suavizar la probabilidad de unigramas.