Equipo 1.
Práctica 1: Creación de Diccionario para aplicación de word2vec.

Resumen de “Distributed Representations of Words and Phrases and their Compositionality”.

En el texto sobre **Distributed Representations of Words and Phrases and their Compositionality”,** no es otra cosa más que la explicación de lo que conocemos como word2vec, obviamente mucho más detallado y con algunos detalles como el softmax jerárquico, detalles de los modelos log-lineales, entre otros ejemplos dados.

Basándose en la introducción del texto podemos notar que las representaciones distribuidas de palabras en un espacio vectorial pueden ayudar a tener mejor rendimiento en un algoritmo de aprendizaje de NLP para agrupar mejor las palabras.

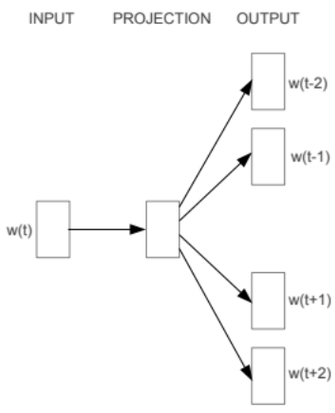
Para que una máquina aprenda del texto sin procesar, necesitamos transformar estos datos en un formato vectorial que luego puedan ser procesados fácilmente por nuestras computadoras. Esta transformación de texto sin procesar en un formato vectorial se conoce como representación de palabras.

Esta representación muestra la palabra en el espacio vectorial, de modo que si los vectores de palabras están cerca unos de otros, significa que están relacionadas entre sí.

Concepto previo. Word Embeddings. Podemos ver una incrustación de palabras (**Word Embeddings**) como una representación vectorial continua de una palabra.

SKIP-GRAM MODEL. El modelo Skip-gram (llamado "word2vec") es uno de los conceptos más importantes en la NLP moderna, sin embargo, muchas personas simplemente usan su implementación y/o incrustaciones previamente capacitadas, y pocas personas comprenden completamente cómo se construye el modelo.

Skip-gram se utiliza para predecir la palabra de contexto para una palabra de destino determinada. Aquí, se ingresa la palabra de destino mientras que se generan las palabras de contexto. Como hay más de una palabra de contexto para predecir, lo que dificulta este problema.



Skip-gram

Como podemos ver, $w(t)$ es la palabra de destino o la entrada dada. Hay una capa oculta **(projection)** que realiza el producto escalar entre la matriz y el vector de entrada $w(t)$. Después el resultado del producto escalar en la capa oculta se pasa a la capa de salida **(output)**.

La capa de salida calcula el producto escalar entre el vector de salida de la capa oculta y la matriz de peso de la capa de salida. Luego aplicamos la función de activación softmax para calcular la probabilidad de que las palabras aparezcan en el contexto de $w(t)$ en una ubicación de contexto dada.

El artículo explica más a detalle algunas de las extensiones de este modelo, para mejorar el rendimiento.

También nos habla de cómo las representaciones de palabras están limitadas por su incapacidad para representar frases idiomáticas que no son composiciones de palabras individuales.

Por ejemplo: "Boston Globe" es un periódico, por lo que no es una combinación natural de los significados de "Boston" y "Globe".

Por lo tanto, el uso de vectores para representar las frases completas hace que el modelo Skip-gram sea considerablemente más expresivo.

La adición de vectores a menudo puede producir resultados significativos.

Por ejemplo, $vec("Russia") + vec("river")$ es cerca de $vec("olga\ River")$, y $vec("Germany") + vec("capital")$está cerca de $vec("Berlín")$.

La composicionalidad sugiere que se puede obtener un grado no obvio de comprensión del lenguaje usando operaciones matemáticas básicas en las representaciones de vectores de palabras.

Objetivo del modelo Skip-gram. El objetivo de entrenamiento del modelo Skip-gram es encontrar representaciones de palabras que sean útiles para predecir las palabras circundantes en una oración o un documento.

Más formalmente, dada una secuencia de palabras de entrenamiento $w1, w2, w3, \ldots, wT$, el objetivo del modelo Skip-gram es maximizar el promedio de probabilidad.

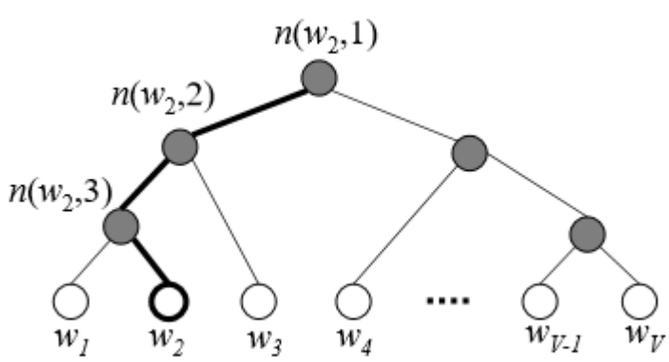
$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0} log\ p(w_{t+j}|w_t)$$

Esto es la probabilidad logarítmica promedio de las palabras de contexto dada la palabra central. Aquí T es el tamaño del corpus, wt son las palabras y c es (la mitad) el tamaño del contexto de entrenamiento. La probabilidad p es la salida de un softmax (jerárquico) sobre similitudes de palabras.

Si c es más grande da como resultado más ejemplos de entrenamiento y, por lo tanto, puede conducir a una mayor precisión, a expensas del tiempo de entrenamiento.

SOFTMAX JERÉRQUICO. La función Softmax genera un vector que representa las distribuciones de probabilidad de una lista de resultados potenciales, mientras que el Softmax jerárquico es una alternativa a softmax que es más rápida de evaluar.

Entonces podemos decir que El softmax jerárquico es una alternativa al softmax en el que la probabilidad de cualquier resultado depende de un número de parámetros del modelo que es solo logarítmico en el número total de resultados.



La principal ventaja es que, en lugar de evaluar los nodos de salida de una red neuronal para obtener la distribución de probabilidad, es necesario evaluar los nodos sobre $log_2(w)$.

El softmax jerárquico utiliza una representación de árbol binario de la capa de salida con las palabras w como sus hojas y para cada nodo representa las probabilidades de sus nodos secundarios.

MUESTREO NEGATIVO. Esta es una simplificación de algo llamado Estimación de contraste de ruido (NCE, Gutmann y Hyvärinen 2012), la idea es que debería poder distinguir ejemplos positivos de ejemplos negativos utilizando regresión logística (también conocida como clasificación binaria) en lugar de elegir la clase correcta de todo el vocabulario.

NCE utiliza las probabilidades numéricas de la distribución del ruido para algunas garantías estadísticas, pero el método de muestreo negativo más simple resultó ser suficiente.

También nos permite modificar solo un pequeño porcentaje de los pesos, en lugar de todos ellos para cada muestra de entrenamiento y esto lo hacemos modificando ligeramente nuestro problema. En lugar de intentar predecir la probabilidad de ser una palabra cercana para todas las palabras del vocabulario, intentamos predecir la probabilidad de que nuestras palabras de muestra de entrenamiento sean vecinas o no.

SUBMUESTREO DE FRECUENCIA DE PALABRAS. En corporaciones muy grandes, las palabras más frecuentes pueden aparecer fácilmente cientos de millones de veces (por ejemplo, "en", "el" y "a"). Estas palabras suelen proporcionar menos valor informativo que las palabras raras.

Por ejemplo, mientras que el modelo Skip-gram se beneficia de observar las coocurrencias de "Francia" y "París", se beneficia mucho menos de observar las frecuentes coocurrencias de "Francia" y "el", ya que casi todas las palabras ocurren juntas frecuentemente dentro de una oración con "el".

Esta idea también se puede aplicar en la dirección opuesta; Las representaciones vectoriales de palabras frecuentes no cambian significativamente después del entrenamiento en varios millones de ejemplos. Para contrarrestar el desequilibrio entre las palabras raras y frecuentes, usamos un enfoque de submuestreo simple.

LEARNING PHRASES. Es curioso saber que, al parecer, hay toda una rama sobre este tema. Los autores eligen un esquema rápido parece funcionar bastante bien, básicamente, hacen un pase sobre los datos de entrenamiento puntuando bigramas por la razón del recuento de bigramas con el producto de los recuentos de unigramo, con un coeficiente de descuento sintonizable .