21/04/2023

Gilbert Márquez Aldana

B94560

Reporte #4

Estuve investigando para comprender mejor sobre word2vec. Primero investigué sobre la arquitectura skip-gram, la cual predice el contexto de una palabra.

Encontré algunas decisiones de diseño que debemos tomar

- La arquitectura que utilizaremos: Skip-gram o CBOW. Si bien CBOW es la que parece ajustarse más a nuestra idea inicial.
- Necesitamos indicar un tamaño de ventana, el cual indicará la cantidad de palabras que conforman un contexto. Por ejemplo, una ventana de tamaño 2, significa que el contexto son las 2 palabras que están antes y las dos palabras que están después.

Word2Vec on source code: Semantic meaning of code and it's beautiful implications

En este blog se realiza un experimento que busca comprobar si se puede realizar un análisis semántico aplicado a código, por ejemplo, *int* debería ser similar a *float* o un algoritmo *bubble sort* debería ser similar a un *selection sort*. Para el experimento se utilizó un repositorio de código de Django, se extrajeron los tokens de todos los archivos y se crearon code sentences. Con estas últimas se entrenó el modelo de Word2Vec.

Este probablemente nos sea de mucha utilidad, ya que es una aplicación de NLP a código, justo lo que nosotros vamos a hacer. Ya Daniel está trabajando en la parte de tokenization. Una vez tengamos los tokens, deberíamos crear los code sentences y, posteriormente, entrenar el modelo de Word2Vec.

Word2Vec – Programming with Text

Me vi esta lista de reproducción que explica Word2Vec y word embeddings. Sin embargo, no creo que me sea de gran utilidad para la investigación.

Tensorflow – Word2Vec tutorial

Realicé una parte de este tutorial, para aprender sobre la implementación del modelo Skip-gram con TensorFlow. En este caso, a partir de una sola oración. Me pude familiarizar un poco con el proceso de entrenamiento, aunque no terminé el tutorial.

La próxima semana voy a investigar sobre la biblioteca de Python llamada Gensim y su implementación de Word2Vec. También voy a terminar el tutorial de TensorFlow

e investigar sobre la implementación de CBOW que se puede realizar. De este modo, podremos decidir cuál biblioteca utilizar.

Además, voy a leer el artículo “Efficient Estimation of Word Representations in Vector Space”, el cual compara distintas arquitecturas de word embeddings, entre ellas a Skip-gram y CBOW.

Referencias

Word2Vec on source code: Semantic meaning of code and it’s beautiful implications
https://medium.com/@amarbudhiraja/word2vec-on-source-code-semantic-meaning-of-code-and-its-beautiful-implications-cb34cafc4a58

Tensorflow – Word2Vec tutorial
https://www.tensorflow.org/tutorials/text/word2vec

Word2Vec – Programming with Text
https://www.youtube.com/playlist?list=PLRqwX-V7Uu6aQ0oh9nH8c6U1j9gCg-GdF