TEMA 3 – HOJA DE EJERCICIOS VII

Se plantean diferentes ejercicios basados en modelos preentrenados de embeddings contextualizados de palabras.

Ejercicio 1. Embeddings Contextualizados - ELMo

A diferencia de los embeddings estáticos, que otorgan un único vector a cada palabra independientemente del contexto en el que se encuentra, los embeddings contextualizados generan representaciones dinámicas que varían según el significado de la palabra dentro de una frase concreta. Esta característica permite capturar mejor la ambigüedad y polisemia presentes en las palabras del lenguaje natural.

Uno de los primeros modelos en proporcionar estos embeddings contextualizados de palabras fue ELMo. Basado en el empleo de redes neuronales recurrentes bidireccionales (biLSTM), ELMo genera vectores dinámicos de palabras en función del contexto en el que aparecen.

De este modo, ELMo proporciona embeddings contextualizados a nivel de palabra, adaptando su representación según el contexto en el que se encuentra dentro de un determinado texto. Por medio de TensorFlow Hub, podemos cargar este modelo mediante el siguiente código de ejemplo:

```
import numpy as np
import tensorflow as tf
import tensorflow_hub as hub
from sklearn.metrics.pairwise import cosine_similarity
elmo = hub.load("https://tfhub.dev/google/elmo/3")
```

Una vez cargado este modelo, es posible obtener los embeddings de las palabras que conforman una frase mediante la siguiente función:

```
elmo.signatures['default'](tf.constant([sentence]))['elmo'].numpy()
```

Dada la frase de consulta "Dogs are domestic animals." y el conjunto de frases ["Dogs are pets.", "This is a dog.", "They are free today."] se pide lo siguiente:

- Con ELMo, obtén el vector promedio de la frase de consulta y también el de cada frase del conjunto de frases.
- Con ELMo, obtén el vector máximo de la frase de consulta y también el de cada frase del conjunto de frases.
- Para las anteriores vectorizaciones, calcula la similitud coseno entre la frase de consulta y el conjunto de frases. Reflexiona sobre qué frases son consideradas más similares por estos modelos y qué vectores funcionan mejor (media o máximo).

  Nota: en este ejercicio no es necesario realizar una etapa previa de tokenización.

  - https://numpy.org/doc/2.2/reference/generated/numpy.mean.html
  - https://numpy.org/doc/2.2/reference/generated/numpy.max.html

Ejercicio 2. Embeddings Contextualizados – BERT

Uno de los modelos más representativos dentro de esta categoría de embeddings es BERT. A diferencia de ELMo, BERT utiliza una arquitectura basada en mecanismos de atención y transformers, lo que le permite considerar el contexto completo en el que se encuentra una determinada palabra al generar su representación vectorial.

De igual forma, BERT proporciona embeddings contextualizados a nivel de palabra en la salida de su última capa. Una vez obtenidos los outputs generados por BERT, es posible acceder a las representaciones vectoriales de las palabras procesadas mediante el atributo `last_hidden_state`:

```
tokens = tokenizer(text, padding=True, truncation=True,
return_tensors="pt")

with torch.no_grad():

   outputs = model(**tokens)

outputs.last_hidden_state.numpy()[:, 1:-1, :]
```

Dada nuevamente la frase de consulta "Dogs are domestic animals." y el conjunto de frases ["Dogs are pets.", "This is a dog.", "They are free today."] se pide lo siguiente:

- Con el modelo y tokenizador de BERT 'bert-base-uncased', obtén el vector promedio de la frase de consulta y el de cada frase del conjunto de frases.
- Con el modelo y tokenizador de BERT 'bert-base-uncased', obtén el vector máximo de la frase de consulta y el de cada frase del conjunto de frases.
- Para las anteriores vectorizaciones, calcula la similitud coseno entre la frase de consulta y cada una del conjunto de frases. Reflexiona sobre qué frases son consideradas más similares por estos modelos y qué vectores funcionan mejor (media o máximo).

Nota: https://huggingface.co/google-bert/bert-base-uncased

Ejercicio 3. Embeddings Contextualizados – SBERT

Recientemente, han surgido diversos modelos basados en BERT para el cálculo de embeddings a nivel de frase. Uno de estos modelos es SBERT, versión optimizada de BERT diseñada específicamente para tareas de recuperación de la información y comparación de frases.

Dada una vez más la frase de consulta "Dogs are domestic animals." y el conjunto de frases ["Dogs are pets.", "This is a dog.", "They are free today."] se pide:

- Con el modelo de SBERT 'all-MiniLM-L6-v2', obtén los embeddings de la frase de consulta y de cada frase del conjunto de frases.
- Con el modelo de SBERT 'all-mpnet-base-v2', obtén los embeddings de la frase de consulta y de cada frase del conjunto de frases.
- Para los anteriores embeddings, calcula la similitud coseno entre la frase de consulta y cada una del conjunto de frases. Reflexiona sobre qué frases son consideradas más similares por estos modelos.

  Nota:

  - https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2
  - https://huggingface.co/sentence-transformers/all-mpnet-base-v2