NLP: Generación de textos

Recurso: Sebastián Peralta

Conceptos NLP:

- Documentos: Se refiere a cada ejemplar de texto con el que se pretende trabajar. Similar a lo que sería una fila o registro en un dataset común.

- Corpus: conjunto de documentos.

- Tokenización: etapa en la cual se procesan los documentos para filtrarlos o modelarlos a un formato deseado. El ejemplo más básico de tokenización es separar una oración por las palabras que la conforman.

- Encoding: asignación o reemplazo de caracteres o palabras según un diccionario de códigos. De tal manera las oraciones o palabras son interpretables por la computadora.

  Ejemplo:
  Oración: “Tengo dos perros y los dos son muy inquietos”
  Encoding: {”Tengo”: 0, “dos”: 1, “perros”: 2, ”y”: 3, ”los”: 4, ”son”: 5, ”muy”: 6, ”inquietos”: 7}
  Resultado: [0, 1, 2, 3, 4, 1, 5, 6, 7]

- Padding: proceso que procura que las oraciones o secuencias de códigos a utilizar como datos de entrenamiento sean de la misma longitud. De tal manera que, no cause conflictos o errores para el input de una red neuronal.

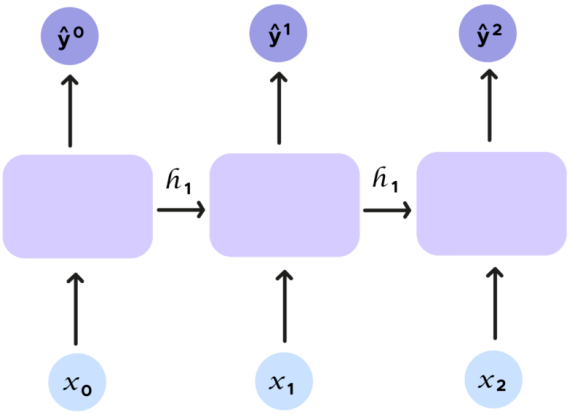
- Embedding: capa de redes neuronales recurrentes utilizada para NLP. Su función principal es determinar las características de los datos en función a los outputs deseados.

- LSTM (Long Short-Term Memory): tipo de capa para redes recurrente que permite que el contexto e impacto de los primeros elementos de una secuencia no se pierdan a través del procesamiento.

Modelos:

Redes Neuronales Recurrentes (RNN - Recurrent Neural Network):

Este tipo de red neuronal se caracteriza por su capacidad de trabajar con data secuencial. La estructura de estas redes simula una dinámica similar a la de la memoria humana, de tal manera que, cada elemento de la secuencia afecta a la siguiente y produce un resultado influenciado por todos los elementos.



La imagen previa describe la estructura de las RNN. Interpretando la imagen de izquierda a derecha, se puede observar que la primera neurona recibe un input x_0 y produce un output y_0, sin embargo, también produce el valor h. Este valor h es utilizado por la siguiente neurona y en conjunto con x_1 produce el output para esa neurona, y_1. Ese proceso continúa a través de toda la secuencia de elementos de entrada y procesa un resultado. En nuestro caso dicha secuencia sería una secuencia de tokens. Esta propiedad de este tipo de red nos permitirá obtener una generación de texto que tenga en cuenta el impacto de cada palabra en la oración y produzca un resultado más coherente.

Generative Pre-Trained Transformer (GPT):

Modelos Transformer creados por OpenAI. Son modelos sofisticados, especializados en la generación de textos. Asimismo, estos modelos han sido entrenados en base a miles de millones de parámetros por lo que es una arquitectura bastante robusta y competente. Las desventajas de los modelos transformer GPT es que la velocidad computacional suele ser lenta. Es decir, suele demorar un tiempo considerable para generar un resultado. Además, por ser modelos “Pre-entrenados” no presentan la capacidad de ser re entrenados, de ser necesario, para que se adapten de mejor manera a un problema específico.

Nota: para algunos modelos GPT es posible importarlos con una menor cantidad de parámetros lo que podría aminorar la velocidad computacional. Además, hay algunas fuentes que indican la existencia de versiones optimizadas de GPT. Sin embargo, implementarlas parece que representa una curva de aprendizaje considerable y no podría afirmar con seguridad que funcionen, por ahora. También existe documentación sobre formas de realizar transferencia de aprendizaje a modelos transformer. Sin embargo solo he encontrado guías extremadamente técnicas para lograrlo para generación de texto. Dicho eso es algo en lo que nos podemos fijar eventualmente.

Conclusiones

En estas instancias de mi investigación favorezco la implementación de una red neuronal recurrente para lograr la generación de texto. Ya que seremos capaces de entrenar el modelo y adaptarlo a los datos que deseemos. También será posible optimizar la arquitectura del modelo para aspirar por los mejores resultados posibles. Asimismo, me parece favorable explorar en una instancia posterior un estudio comparativo, especialmente en términos de velocidad computacional, entre distintas configuraciones de modelos GPT-2

o GPT-neo. La gran desventaja de estos modelos transformer viene a ser su incapacidad para ser entrenados, sin embargo, dependiendo de los resultados de la generación de texto se podrá evaluar si es algo aceptable para ustedes o ver alternativas.

Referencias:

- NLP: Zero to Hero Tensorflow Guide: https://www.youtube.com/watch?v=fNxaJsNG3-s&list=PLQY2H8rRoyvzDbLUZkbudP-MFQZwNmU4S
- ANN vs CNN vs RNN: https://www.youtube.com/watch?v=u7obuspdQu4
- How to use GPT-Neo to Generate AI-based Blog Content: https://www.section.io/engineering-education/leveraging-gptneo-to-generate-ai-based-blog-content/#how-to-leverage-gpt-neo-to-generate-ai-based-blog-content
- What is GPT-3?: https://www.techtarget.com/searchenterpriseai/definition/GPT-3
- Transfer learning with Transformers trainer and pipeline for NLP: https://medium.com/mlearning-ai/transfer-learning-with-transformers-trainer-and-pipeline-for-nlp-8b1d2c1a8c3d