Predicción de puntuación y capitalización en texto normalizado

Emiliano Torres - Franco Vanotti | Licenciatura en Ciencias de Datos

Introducción

Este tipo de sistema tiene aplicaciones reales en procesamiento de lenguaje natural. Un caso típico es el postprocesamiento de la salida de sistemas de reconocimiento automático del habla (ASR, por sus siglas en inglés). Estos sistemas suelen producir texto plano, en minúsculas y sin puntuación, lo cual dificulta su lectura y comprensión.
Nuestro objetivo es construir modelos que permitan "reconstruir" ese texto enriquecido con puntuación y mayúsculas, mejorando su utilidad en tareas posteriores como: Resumen automático, Análisis de sentimiento o Traducción automática. En particular vamos a concentrarnos en agregar capitalización de 3 tipos: palabras que se escriben completamente en mayúscula, palabras que tienen la primer letra en mayúscula y palabras con algunas mayúsculas. En el caso de la puntuacion vamos a concentrarnos en el agregado de puntos, comas y signos de preguntas.

Nuestro enfoque para resolver el problema: *Un modelo Machine Learning.*

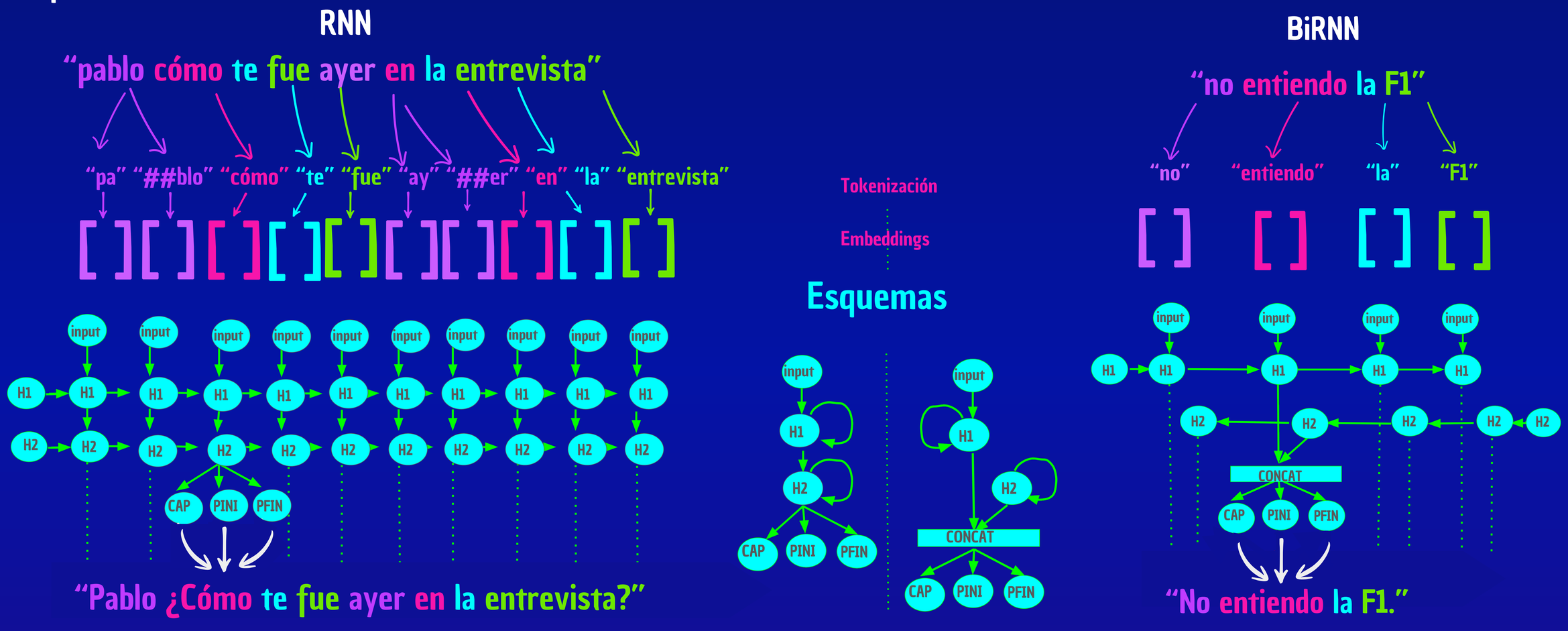
Problemas

- Contamos con representaciones estáticas de los tokens, pero las tareas de capitalización y puntuación necesitan el contexto en el que se encuentra cada token.
- Entrenar instancia a instancia era muy costoso. Además la longitud variable de las secuencia nos dificulto crear batches de instancias.

Soluciones

- Usar RNN como encoders para mezclar el contexto de los tokens creando una representación enriquecida de los mismos para facilitar la tarea de clasificación.
- Agregamos padding por batch para poder paralelizar el trabajo en GPU.

Arquitecturas



Dataset

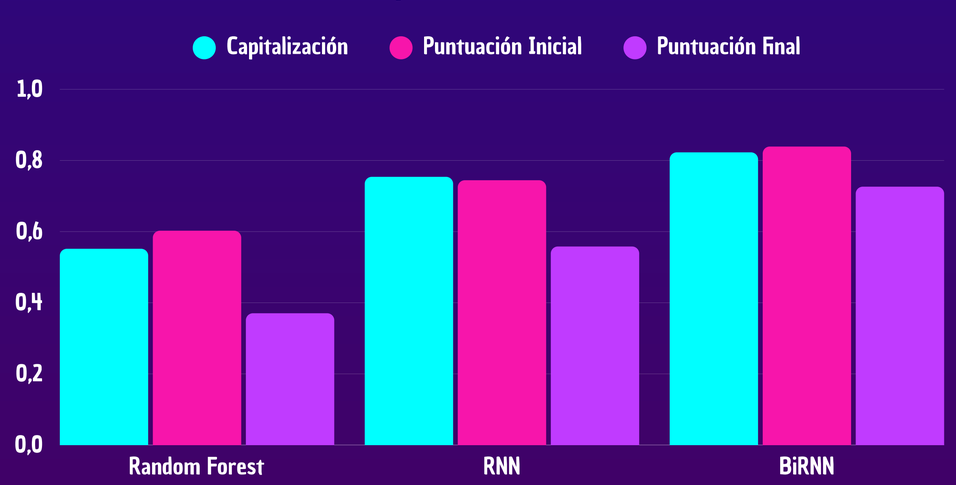
El dataset que utilizamos para entrenar a nuestros modelos cuenta con dialogos de peliculas en distintos idiomas. Tomamos un 10% del dataset en español para limpiarlo y construir los datos que alimentarán a nuestra redes neuronales y al random forest. En la tabla podemos observar las proporciones de cada clase. Entrenamos las redes con 1 millón 2 mil datos usando el 80% para entrenamiento y el 20% para test. Dejamos cien mil datos como held out para reportar las métricas

DATOS REDES NEURONALES	
Cantidad de instancias	1.200.000
Porcentaje por categoría capitalización	
Tokens que arrancan en mayúscula %	70.56 %
Tokens todos en minúsculas %	22.92 %
Tokens con mayúsculas y minúsculas alternadas %	5.81 %
Tokens todos en mayúscula %	0.71 %
Porcentaje por categoría puntuación inicial	
Tokens sin puntuación inicial %	97.89 %
Tokens que abren preguntas %	2.11 %
Porcentaje por categoría puntuación final	
Tokens sin puntuación final %	83.31 %
Tokens que cierran preguntas %	2.52 %
Tokens que terminan en punto %	10.34 %
Tokens que terminan en coma %	3.83 %

DATOS RANDOM FOREST	
Cantidad de instancias	60.000
Porcentaje por categoría capitalización	
Tokens que arrancan en mayúscula %	70.65 %
Tokens todos en minúsculas %	23.11 %
Tokens con mayúsculas y minúsculas alternadas %	5.53 %
Tokens todos en mayúscula %	0.71 %
Porcentaje por categoría puntuación inicial	
Tokens sin puntuación inicial %	97.88 %
Tokens que abren preguntas %	2.12 %
Porcentaje por categoría puntuación final	
Tokens sin puntuación final %	83.27 %
Tokens que cierran preguntas %	2.52 %
Tokens que terminan en punto %	10.40 %
Tokens que terminan en coma %	3.81 %

Resultados

Tabla comparativa (F1 macro) Held Out			
Modelo	Capitalización	Puntuación Inicial	Puntuación Final
Random Forest	0.5517	0.6025	0.3704
RNN	0.7540	0.7444	0.5581
BiRNN	0.8228	0.8391	0.7263



Fortalezas y debilidades de las redes neuronales

Consideramos que la mejor forma de entender que tan bien están funcionando nuestros modelos es leyendo el texto que enriquecen de una manera crítica. Tomamos un conjunto de 20 instancias nunca antes vistas por el modelo e hicimos inferencia de su puntuación y capitalización, con las que llegamos a las siguientes conclusiones:

Fortalezas:

- La RNN con celdas LSTM predijo con alta precisión las preguntas cortas y las puntuaciones en frases medianas.
- La BiRNN tiene una mejor compresión de estructuras de texto más complejas.
- La BiRNN tiene una mejor capacidad de identificar palabras de minusculas y mayusculas alternadas. Por ejemplo: iPhone.
- El contexto bidireccional le permite usar información futura y pasada, mejorando la colocación de puntuación.

Debilidades:

- La RNN tuvo problemas en oraciones más largas sobre todo las que mezclaban preguntas con frases, entendemos que porque "saturaron" sus estados ocultos.
- La RNN tuvo dificultades para puntualizar preguntas consecutivas entendemos que se debe a la falta de contexto futuro.