Arquitecturas profundas para el reconocimiento de entidades nombradas

Santiago Moreno
Facultad de ingeniería
Universidad de Antioquia
Medellín, Colombia
santiago.moreno3@udea.edu.co

Nestor Calvo
Facultad de ingeniería
Universidad de Antioquia
Medellín, Colombia
nestor.calvo@udea.edu.co

***Index Terms*—NER, NLP, Deep Learning**

I. Contexto

En el análisis de texto uno de los primeros problemas a resolver es la identificación de las entidades a las que se refieren las palabras en cada frase. Este reconocimiento da paso a un análisis con más detalle del texto, con técnicas como la identificación de relaciones entre entidades, o el reconocimiento de entidades más específicas agrupadas dentro de una super-entidad previamente reconocida.

El reconocimiento de entidades nombradas (NER por sus siglas en inglés) se enfoca en reconocer menciones de entidades en un texto dado, como, ubicaciones, personas, organizaciones, entre otras [1]. Por ejemplo, en la figura 1, se identifica que Jorge Robledo es una persona y la segunda aparición de la palabra Robledo se refiere a una ubicación.

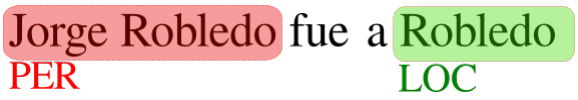

Figura 1. Ejemplo NER

Para desarrollar un modelo que realice el NER se debe tener en cuenta que exisitirán tantas clases como entidades haya, y aquellas palabras que no tengan una entidad asociada se etiquetarán con la entidad OTHERS (O).

El formato mas usado para este tipo de problema es el formato CONLL, este consiste en un archivo de texto plano en donde hay un documento de 2 columnas, la primera columna corresponde a la palabra y la segunda corresponde a su entidad (clase). Cada frase está compuesta por una serie de filas (palabras con etiqueta) cuyo fin de frase e inicio de la siguiente está representado por una que contiene solo un salto de línea.

Dado que una entidad puede estar compuesta por varias palabras las etiquetas pueden tener diversos formatos para reconocer esto. De los mas conocidos es el IOB2, el cual agrega una letra al inicio de cada entidad, en donde B representa el inicio de una entidad (begin) e I representa las palabras siguientes al inicio de la entidad (inside). Las palabras etiquetadas como OTHERS mantienen si etiqueta O y no tiene esta característica.

II. Objetivo Machine Learning

Ya que para reconocer las entidades en un texto se debe tener en cuenta el contexto de toda una frase, para ello es necesario implementar técnicas que modelen datos secuenciales.

Teniendo un dataset en el formato CONLL se busca implementar un modelo que etiquete cada palabra dentro de una frase, teniendo en cuenta el contexto que brinda la frase misma.

III. Dataset

El dataset utilizado es tomado de [4] y consiste en una serie de documentos legales en alemán previamente etiquetados y tomados de decisiones juridícas. La base de datos contiene en total 20 entidades, 19 específicas mas la entidad ”OTROS”.

En total el dataset tiene 66,723 frases con 2,157,048 palabras, cada palabra está etiquetada en el formato IOB2, esta se depura para eliminar este formato y facilitar la obtención de métricas de desempeño. La organización de la base de datos es conforme al formato CONLL.

IV. Notebooks

Para el desarrollo del trabajo se realizó por medio de un solo notebook el cual se encarga de realizar todo el proceso necesario para el entrenamiento del modelo, esto se realizó con el fin de garantizar el correcto funcionamiento de todas la partes. Para este notebook se siguió la estructura que se observa en la Figura 2

Primero se tiene una celda la cual se encarga de importar las librerias e instalar los paquetes necesarios para todo lo que se va a realizar posteriormente, luego se tiene un conjunto de celdas que se usa para todo lo que corresponde a la carga y analisis del dataset, para el proyecto se hace uso primero de un dataset que contiene las palabras y su respectiva etiqueta en formato CONLL. Segundo, de un archivo de embeddings. Ambos archivos se descargan por medio del comando !curl, el dataset se encuentra almacenado en el repositorio de GitHub, mientras que el archivo de embeddings (debido a su peso) se encuentra almacenado en una carpeta de OneDrive.

Una vez descargados ambos archivos se deben leer y almacenar en variables para su posterioro uso, esto se realiza en las celdas correspondientes a Data Load. Previo al tratamiento y/o preprocesado de datos, se debe hacer un breve analisis de

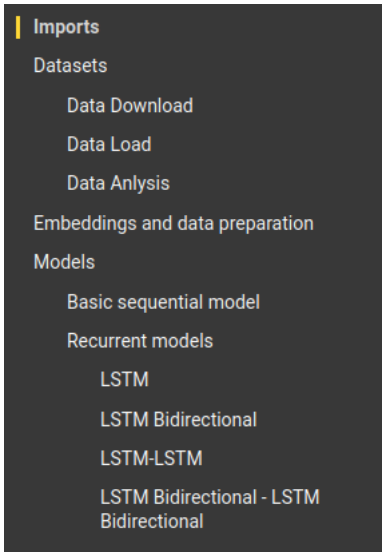

Figura 2. Estructura de notebook (imagen tomada direcatmente del indice de Google Colab)

estos para conocer como se encuentra el dataset, este proceso se realiza en Data Analysis.

Una vez que se ha observado la estructura de los datos se procede a hacer un preprocesado al dataset el cual se encuentra en .[Em]beddings and Data Preparation, en esta capa se acomoda el dataset que será usado durante los diferentes modelos. Posteriormente la capa de modelos se dividen en modelo secuencial y los modelos recurrentes.

V. Descripción de la solución

Como solución al problema del reconocimiento de entidades nombradas se plantea un sistema de etiquetado automático de palabras de palabras. Cada palabra tendrá asociada una etiqueta que representa la entidad a la que pertenece dicha palabra. Aquellas palabras que no tengan una etiqueta asociada se les asigna la etiqueta OTROS, representada por la letra "O", esto es propio del formato de entidades IOB2 con el cual cuenta el dataset usado.

Para la solución propuesta la entrada será una frase y su salida será la serie de etiquetas para la frase. Esta sistema onsiste en una capa de embedimiento de palabras, seguido de una capa de processado de información y por último una capa de salida la cual entregará la etiqueta asociada a cada palabra en la frase.

Las etapas de solución realizadas se describen a continuación:

V-A. Preparación de datos:

Ya que se cuenta con una base de datos en el formato CONLL no es necesario definir un estándar para la organización de los datos.

El primer paso en el procesado es definir el número de palabras maxima por frase. El largo de las frases del dataset utilizado cuenta con una distribución dada por la Figura 3 :

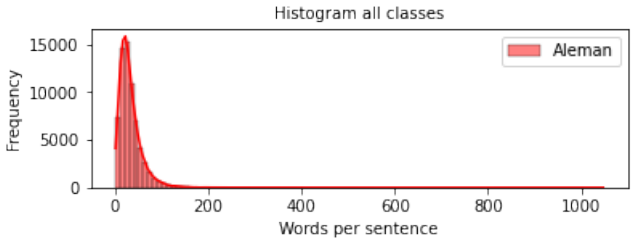

Figura 3. Histograma del largo de las frases sin truncar

Como se puede observar la mayor concentración de datos tiene menos de 150 palabras por frase, con algunos outliers que llegan a tener mas de 1000 palabras. Por esto se define una longitud de frase máxima dada por el percentil 99 de la distribución, esto es 120 palabras por frase.

La distribución resultante está dada por la Figura 4 :

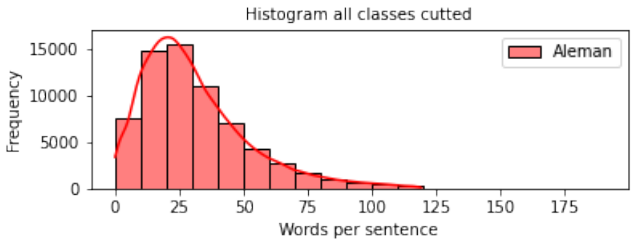

Figura 4. Histograma del largo de las frases truncado

Luego de definir la longitud máxima se debe estándarizar el largo de todas las frases para los modelos. Para esto se añade a cada frase la palabra "PADDING" con la etiqueta "O" hasta que su largo sea 120. De esta manera todas las frases tienen una longitud de 120. Por último, las etiquetas usadas son codificadas de manera categórica en números enteros.

V-B. Implementación de modelos

Se proponen diferentes modelos que cumplen con la tarea de reconocimeinto de entidades. Cada modelo cuenta con las siguientes capas.

Capa de embedimiento: Esta capa se encargará de transformar las palabras que se encuentran en texto bruto por una representación vectorial en un espacio de 100 dimensiones. Así cada palabra será ubicada en el espacio dependiendo de su contexto, aportando información semántica a la solución del problema. Los valores de embedimiento son estáticos se obtienen de un modelo Word2Vec basado en arquitectura Skip-Gram entrenada con 116 milliones de frases extraídas de diferentes bases de datos [5].

Capa de procesado: Una vez las palabras son cambiadas por su representación vectorial se procede a procesar los datos haciendo uso de arquitecturas de Deep Learning como capas densas, unidades de Long-Short Term Memory (LSTM) y LSTM bidireccionales (BiLSTM).

Capa de salida: Cuando los datos ya son procesados por la capa intermedia se hace necesario asignar una etiqueta a cada

Cuadro I
RESULTADO DE MODELOS

	LSTM	BiLSTM	LSTM-LSTM	BiLSTM-BiLSTM
Valor F1	0.37	0.42	0.25	0.48

palabra dentro de la frase ingresada. Esta última capa tendrá tantas salidas como etiquetas haya en el corpus y entregará la etiqueta que tiene una mayor probabilidad de pertenecer a la palabra de ínteres, para obtener esta probabilidad por etiqueta se usa la función de activación softmax.

Primeramente se plantea un sistema cuya arquitectura está compuesta de capas densas como capa de procesado y su entrada es solo una palabra sin información del contexto de la frase.

Ya que esta solución no tiene en cuenta la información sequencial propia de los datos, se proponen arquitecturas que haga uso de modelos de procesado de información secuencial. Para esta solución se prueban diferentes combinaciones de capas intermedias como LSTM, BiLSTM, LSTM-LSTM y BiLSTM-BiLSTM. Para todas las soluciones propuestas se usa la función de perdida "Sparse categorical crossentropy".

V-C. Evaluación de modelos

Para medir el desempeño de los modelos se obtienen metricas de desempeño para cada etiqueta. Las métricas usadas son sensitividad, especificidad y el valor F1.

VI. ITERACIONES

VI-A. Selección de modelo

Todos los modelos cuentan con la capa de embeddings inicial y la capa de salida con activación softmax, se probaron diferentes combinaciones en la capa de procesado. Para todos los modelos se usa la función de perdida "Sparse categorical crossentropy" debido a su forma de trabajar con etiquetas representadas por números enteros. Y el optimizador ADAM con una tasa de aprendizaje de 0,001.

Modelo denso: Para la selección del modelo se utilizó primero un modelo que consiste en 3 capas densas una 512, otra de 256 y otra de 128 neuronas respectivamente. A este modelo se le ingresa una palabra sin información del contexto de la frase y este entrega la etiqueta de dicha palabra

Con el fin de utilizar la información del contexto de la frase se implementan modelos basados en redes recurrentes:

Modelos basados en redes recurrentes: Para estos modelos se utilizan las celulas LSTM con los siguientes parámetros: 128 celulas, dropout de 20 % y una tasa de aprendizaje de 0,001. Se usaron las siguientes configuraciones: LSTM, BiLSTM, LSTM-LSTM y BiLSTM-BiLSTM. De estos se obtuvieron los resultados mostrados en la Tabla I.

VI-B. Modelo final:

Al observar los resultados preeliminares se escoge el modelo conformado por dos capas LSTM bidireccionales. Su arquitectura se muestra en la Figura 5.

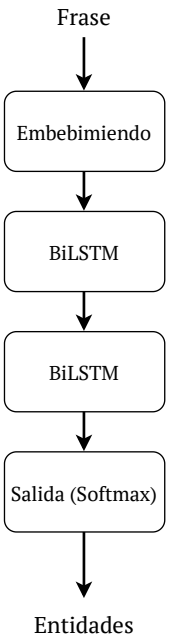

Figura 5. Arquitectura final

VII. RESULTADOS

Con el modelo final se ontienen los resultados para cada clase en la Figura 6

	precision	recall	f1-score	support
AN	0.00	0.00	0.00	26
EUN	0.00	0.01	0.01	308
GRT	0.24	0.05	0.08	631
GS	0.55	0.80	0.65	3725
INN	0.06	0.04	0.05	409
LD	0.00	0.00	0.00	278
LDS	0.00	0.00	0.00	35
LIT	0.31	0.55	0.40	640
MRK	0.00	0.00	0.00	46
ORG	0.00	0.00	0.00	248
PER	0.00	0.00	0.00	330
RR	0.00	0.00	0.00	348
RS	0.59	0.68	0.63	2471
ST	0.00	0.00	0.00	136
STR	0.00	0.00	0.00	16
UN	0.00	0.00	0.00	190
VO	0.00	0.00	0.00	152
VS	0.00	0.00	0.00	135
VT	0.00	0.00	0.00	578
micro avg	0.49	0.47	0.48	10702
macro avg	0.09	0.11	0.09	10702
weighted avg	0.36	0.47	0.40	10702

Figura 6. Resultados finales

VIII. REFERENCIAS Y RESULTADOS PREVIOS

Referente al problema del NER se han tratado diferentes enfoques. Las primeras propuestas de solución se realizaron

con un enfoque basado en reglas [2] [4], luego se trataron algoritmos supervisados con extracción de características [3] [4] y después con el aumento de la capacidad computacional y el número de datos se comenzó a utilizar un enfoque profundo [4].

En [4] se utilizó una base de datos etiquetada con el formato BIO2. Allí se implementó un modelo basado en campos aleatorios condicionales (CRF por sus siglas en inglés), con 19 entidades obtuvieron un F1-score de 93.05 %. En este mismo trabajo se implementó un enfoque profundo, generando word embeddings con Word2Vec [5] e implementando una BiLSTM con una capa de CRF, con esta arquitectura se obtuvo un F1-score de 93.75 %.

REFERENCIAS

[1] Nadeau, D., Sekine, S. (2007). A survey of named entity recognition and classification. Lingvisticae Investigationes, 30(1), 3-26.

[2] Kim, J. H., Woodland, P. C. (2000). A rule-based named entity recognition system for speech input. In Sixth International Conference on Spoken Language Processing.

[3] Krishnan, V., Manning, C. D. (2006, July). An effective two-stage model for exploiting non-local dependencies in named entity recognition. In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (pp. 1121-1128).

[4] Leitner, E., Rehm, G., Moreno-Schneider, J. (2019, September). Fine-grained named entity recognition in legal documents. In International Conference on Semantic Systems (pp. 272-287). Springer, Cham.

[5] Reimers, N., Eckle-Kohler, J., Schnober, C., Kim, J., Gurevych, I. (2014). Germeval-2014: Nested named entity recognition with neural networks.