

Práctica PLN – Modelos neuronales para etiquetado de secuencias

En esta práctica, implementaremos dos redes neuronales para resolver tareas de etiquetado de palabras en oraciones escritas en lenguaje natural, y las aplicaremos a dos problemas clásicos de PLN: etiquetación morfológica (*PoS tagging*) y reconocimiento de entidades nombradas (*named-entity recognition*). La Tabla 1 ilustra gráficamente en qué consisten estas dos tareas con un ejemplo.

Inputs	En	Can	Roca	el	menú	cuesta	280	€
POS	PREP	PROPN	PROPN	DET	NOUN	VERB	NUM	SYMBOL
NER	O	B-ORG	I-ORG	O	O	O	B-PRICE	I-PRICE

Tabla 1. Ejemplo de una oración de entrada y una posible salida para las dos tareas propuestas: etiquetación morfológica y reconocimiento de entidades.

Los datasets que usaremos para ambas tareas se incluyen con los materiales de esta práctica: (i) el dataset PartUT para etiquetación morfológica y (ii) los datasets MITRestaurant y MITMovie para el reconocimiento de entidades nombradas sobre dominios de restauración y cine, respectivamente. Deberá entrenarse una red distinta para cada dataset.

En concreto, **se pide implementar dos arquitecturas neuronales** para tareas de etiquetado de secuencias.

Modelos basados en snippets y redes feed-forward: El modelo deberá ser de tipo [Sequential](#) o [Functional API](#). Se propone una arquitectura por defecto para la red. Para clasificar cada palabra, la entrada a la red será un snippet, un vector compuesto por la palabra a etiquetar, así como las n palabras previas y las n siguientes a ella. La red debe ser configurable para este parámetro y deberán reportarse experimentos para $n=0, 1, 2$. Se propone una configuración por defecto para la red, con libertad para tocar aspectos como los hiperparámetros o el número de capas. En concreto, se propone que el vector de palabras de entrada sea convertido a un conjunto de representaciones vectoriales, cada una de tamaño 20, usando una capa [Embedding](#). Dichos vectores serán concatenados y enviados a una capa [Dense](#) intermedia, de dimensión 64, que será a su vez conectada a otra capa de salida Dense, que calcula la distribución de probabilidad para predecir la etiqueta de cada palabra través de una función softmax.

Modelo LSTM: El modelo deberá ser de tipo [Sequential](#) o [Functional API](#). Se propone una arquitectura por defecto para la red: (i) una capa [Embedding](#) que transforme las palabras en vectores de tamaño 20, (ii) una capa [LSTM](#) con dimensión de salida 64 que procese la secuencia de vectores y (iii) una capa de salida consistente en una capa [Dense](#), que de nuevo calcula la distribución de probabilidad para predecir la etiqueta de cada palabra través de una función softmax. Se deberá implementar una LSTM unidireccional y otra versión Bidireccional haciendo uso de la capa [Bidirectional](#).

Las funcionalidades que deberán soportar las redes serán:

1. **Entrenar los modelos** utilizando un conjunto de entrenamiento (train.txt) y un conjunto de desarrollo (dev.txt) para la evaluación interna.
2. **Testar los modelos entrenados.** Utilizar el modelo entrenado para calcular el rendimiento de las dos redes sobre un conjunto de oraciones test.txt. En el caso de etiquetación morfológica se pide reportar la accuracy. En el caso de los datasets para el reconocimiento de entidades, además de la accuracy se hará uso del repositorio [nervaluate](#) para reportar la F1-score para las variantes ent_type, partial, exact y strict para cada tipo de entidad.

Una posible opción de presentación de la práctica es mediante un programa Python que se ejecute con el siguiente comando:

```
python main.py --model [ff,lstm,bilstm] --task [ner,pos] --train PATH_TRAINING_SET --dev PATH_DEVELOPMENT_SET --test PATH_TEST_SET
```

Donde --model recibe unos de los valores FF, LSTM, BILSTM que hacen referencia al modelo feed-forward, al modelo con LSTMs o al modelo con LSTM bidireccionales; --task es la tarea para la que sea entrenado el modelo; y donde PATH_TRAINING_SET, PATH_DEVELOPMENT_SET y PATH_TEST_SETS son las rutas correspondientes a los ficheros de train.txt, dev.txt y test.txt de un determinado dataset. Podéis añadir otros parámetros que consideréis oportunos (por ejemplo, número de iteraciones a entrenar, tamaño de las capas si optáis por valores no por defecto, etc).

Dicho comando podría entrenar el modelo seleccionado, y seguidamente reportar las métricas de evaluación sobre el conjunto de test en función de la tarea seleccionada.

Entrega

La práctica debe incluir un breve manual de usuario, que indique cómo ejecutar la práctica para entrenar y evaluar los modelos en los distintos datasets. También debería incluirse un análisis de los resultados y una breve discusión de las diferencias que aprecia entre ambas implementaciones, así como sus ventajas e inconvenientes. La memoria no debe exceder las 3 páginas y deberá utilizar un tipo de letra Arial, Calibri o Times New Roman con un tamaño mínimo de 11 pt.

Las prácticas se realizarán en parejas, respetando los grupos formados en el primer módulo de la asignatura.

Seguimiento parcial: Durante la semana del 17 de abril, se hará un seguimiento durante las horas de clases sobre el avance de la práctica. Durante esa semana, el modelo basado en snippets y redes feed-forward deberá estar implementado y deberá poder ejecutarse (entrenamiento y evaluación) sobre los datasets propuestos. Los grupos que no aprueben este seguimiento parcial tendrán una penalización de 0.3 puntos sobre la nota final de 2 puntos de las prácticas de este módulo. Será posible seguir trabajando en el modelo feed-forward hasta la entrega final para corregir los errores detectados. Los alumnos que obtengan el total de la nota para el modelo feed-forward, solo serán evaluados del modelo LSTM en la entrega final.

Defensa: La práctica deberá subirse al campus virtual, en el apartado habilitado para ello, como muy tarde el 5 de mayo a las 22:00 horas. Únicamente un miembro del grupo debe subir la práctica. La defensa (obligatoria) tendrá lugar durante la última semana de prácticas (del 8 al 12 de mayo). Los estudiantes que no acudan a dicha defensa suspenderán la práctica con una nota de 0 puntos.