

Práctica 3: Modelos Neuronales de Etiquetado de Secuencias

Fundamentos de Procesamiento de Lenguaje Natural

Pablo Chantada Saborido

Lucía Bardanca Rojo

Marcelo Ferreiro Sánchez

1. Manual de Usuario

Para ejecutar los modelos, se utiliza el siguiente comando:

```
1 python main.py --embedding [random, word2vec] --task [ner, pos] \  
2 --train PATH_TRAINING_SET --dev PATH_DEVELOPMENT_SET \  
3 --test PATH_TEST_SET
```

Donde:

- **--embedding**: Tipo de embedding (aleatorio o pre-entrenado)
- **--task**: Tarea a realizar (pos o ner)
- **--train, --dev, --test**: Rutas a los conjuntos de datos
- **--bidirectional**: Usar LSTM bidireccional (predeterminado)
- **--no_bidirectional**: Usar LSTM simple en lugar de bidireccional

2. Arquitectura del Sistema

2.1. Modelo

- **Capa de Embedding**
 - Dimensión: 200 (random) o 300 (word2vec)
 - Inicialización: Aleatoria o pre-entrenada (Word2Vec)
- **Capa LSTM**
 - Unidades: 128
 - Bidireccional: Sí (configurable)
 - Dropout: 0.3 para prevenir sobreajuste
- **Capa de Salida**
 - TimeDistributed(Dense) con activación softmax
 - Dimensión: Número de etiquetas únicas
- **Hiperparámetros**:
 - Epochs: 30-50 (con early stopping)
 - Batch size: 32
 - Optimizer: Adam (learning rate=0.001)
 - Loss: Categorical Crossentropy

3. Experimentos y Resultados

3.1. Resultados POS Tagging

Modelo	Precisión	Pérdida	Tiempo (s)
Bidireccional + Embeddings Aleatorias	0.9387	0.1093	120.11
Bidireccional + Embeddings Word2Vec	0.9174	0.1486	292.65
LSTM Simple + Embeddings Aleatorias	0.9135	0.1347	72.13
LSTM Simple + Embeddings Word2Vec	0.8917	0.1586	166.68

Cuadro 1: Resultados de los modelos de POS tagging

3.2. Resultados NER

Modelo	Precisión	Pérdida	F1 (strict)	Tiempo (s)
Word2Vec + BiLSTM	0.8732	0.3341	0.6900	446.21
Random + BiLSTM	0.8487	0.3357	0.6744	191.67
Word2Vec + LSTM Simple	0.8411	0.3848	0.6084	455.85
Random + LSTM Simple	0.8299	0.3962	0.5545	207.29

Cuadro 2: Resultados globales de los modelos NER

Entidad	F1-Score (modo strict) por Modelo			
	W2V+Bi	Random+Bi	W2V+Simple	Random+Simple
Location	0.7108	0.6792	0.6424	0.6512
Cuisine	0.7557	0.6917	0.7170	0.7080
Amenity	0.5743	0.4992	0.4773	0.4244
Dish	0.6710	0.6751	0.6645	0.6525
Price	0.7427	0.6445	0.5979	0.5854
Restaurant_Name	0.7079	0.6134	0.6243	0.5833
Hours	0.6166	0.5465	0.5257	0.4753
Rating	0.7059	0.6581	0.6510	0.6039

Cuadro 3: F1-Score por tipo de entidad para cada modelo NER

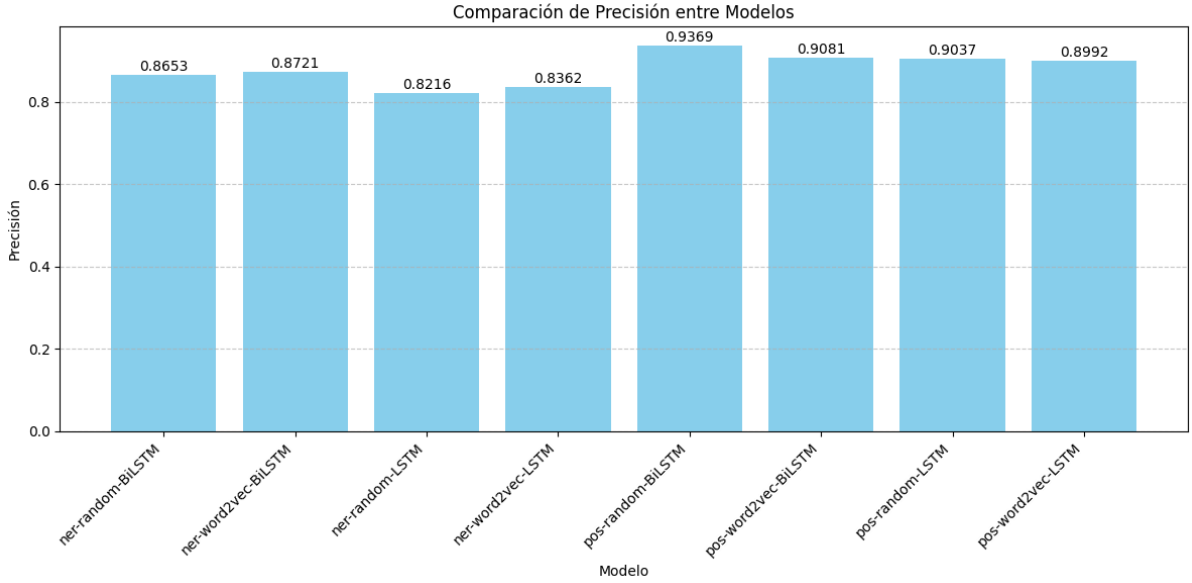


Figura 1: Comparación de precisión entre modelos de POS y NER

4. Análisis de Resultados

4.1. Comparativa de Embeddings

Para POS tagging, los embeddings aleatorios funcionaron mejor que Word2Vec (precisión de 0.9387 vs. 0.9174 con BiLSTM), lo que sugiere que en tareas con categorías bien definidas, las representaciones aleatorias pueden ser efectivas. En cambio, en NER, Word2Vec + BiLSTM ofreció mejor F1-score (0.6900 strict), especialmente en entidades como “Restaurant_Name”. Sin embargo, estos modelos requieren más tiempo de entrenamiento, debido a la mayor dimensionalidad (300D) y carga computacional.

Este comportamiento puede explicarse porque POS dependen más de patrones sintácticos que de semántica, permitiendo que embeddings aleatorios aprendan eficientemente durante el entrenamiento.

En contraste, NER se beneficia de la información semántica pre-entrenada de Word2Vec, que ya contiene conocimiento sobre entidades y relaciones del mundo real.

4.2. Análisis por Tipo de Entidad

Entidades como “Cuisine” y “Location” fueron mejor reconocidas ($F1 > 0.67$), mientras que “Amenity” y “Hours” presentaron mayores dificultades ($F1 < 0.55$); seguramente por la naturaleza gastronómica del Dataset (MITRestaurant). Word2Vec + BiLSTM fue especialmente útil para entidades con semántica específica. Los modelos son competentes detectando el tipo de entidad (modo `ent_type`), pero menos precisos con sus límites exactos (modo `strict`). La detección de límites es más difícil que la clasificación porque requiere identificar precisamente dónde empieza y termina una entidad, especialmente en casos de entidades multipalabra o con estructura variable.

4.3. Comparativa POS vs NER

POS tagging obtuvo mejores métricas (Accuracy = 0.9387) y menor pérdida (0.1093-0.1586 vs. 0.3341-0.3962), además de converger más rápido. Por el contrario, NER implica más complejidad debido al desbalance y diversidad de entidades. Esto se refleja en mayores tiempos de entrenamiento (191-455s vs. 72-292s) y pérdidas más altas.

Esta diferencia de rendimiento puede explicarse principalmente porque POS asigna etiquetas gramaticales a todas las palabras siguiendo patrones bien definidos, mientras que NER debe primero detectar si una palabra forma parte de una entidad y luego clasificarla correctamente. Además, en NER la misma palabra puede ser o no una entidad según el contexto, y las entidades pueden abarcar múltiples tokens consecutivos, aumentando la complejidad de detectar dónde comienzan y terminan.

4.4. Impacto de la Arquitectura LSTM

Los modelos BiLSTM superan a los LSTM simples en todas las métricas: +2-10 % de mejora en F1 y precisión. El coste es un aumento de 40–60 % en tiempo de entrenamiento. Su ventaja radica en la capacidad de considerar contexto anterior y posterior, permitiendo al modelo aprender el contexto completo de la palabra.

5. Conclusiones

Los modelos BiLSTM, especialmente combinados con Word2Vec, ofrecen el mejor rendimiento en NER, aunque a mayor coste computacional. Para POS, embeddings aleatorios con BiLSTM son una opción eficiente. La elección final del embedding depende de la tarea: semántica rica y mayor importancia al contexto (NER) vs. estructura gramatical (POS).