Trabajo Práctico 1

Minería de Textos

Irene J. Ventura Farias

Descripción de la tarea conll2002 y los datos de evaluación.

Una vez leído el artículo: ***Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition*** , podemos comentar que la tarea conll2002 tiene como objetivo desarrollar un sistema de etiquetador de entidades (NER), implementando componentes de aprendizaje automático y que sea independiente del lenguaje.

Este etiquetador de entidades busca clasificar palabras con las siguientes entidades:

- **PER** -> Persona
- **ORG** -> Organización
- **LOC** -> Localización
- **MISC** -> Miscelánea (No pertenecen a ninguna de las entidades anteriores)

Exclusivamente para este sistema, se trabajó con textos en español y holandés. Sin embargo, los organizadores estaban también interesados en utilizar textos con palabras sin etiquetas para mejorar su rendimiento.

Para cada idioma, los datos consistían en 3 ficheros:

- **Datos de Entrenamiento** para los métodos de aprendizaje.
- **Datos de Desarrollo** para ajustar los hiperparámetros del modelo.
- **Datos de Prueba** para evaluar el modelo de aprendizaje

Todos los ficheros de datos mantienen la misma estructura: Cada dato esta representado en una línea. Cada línea está compuesta por una palabra, seguida de una etiqueta que indica si pertenece a una entidad o no. El esquema de etiquetado es una variante del formato IOB.

Concretamente, en el caso de los datos españoles son colecciones de artículos de noticias facilitados por la EFE, los cuales contienen únicamente palabras y etiquetas de entidad.

Para el evaluar el desempeño del etiquetador de entidades se tuvieron en cuenta las siguientes medidas:

- **Precisión** evalúa el porcentaje de entidades encontradas por el sistema y que son correctas

- **Cobertura (Recall)** evalúa la cantidad de entidades etiquetadas correctamente en función del total de palabras etiquetadas por entidades.

- **Medida-F** mide el porcentaje de equilibrio entre la precisión y la cobertura.

Doce sistemas de etiquetado se diseñaron para lograr esta tarea. Además, se calculó una tasa para cada conjunto de datos (español y Holandés) para evaluar las métricas.

Los resultados concluyentes sobre los doce sistemas han sacado mejores resultados que los valores base, pero el de Carreras et al., 2002 fue el que saco mejores resultados.

Objetivo de la Práctica

El objetivo de esta primera práctica es utilizar un etiquetador de entidades nombradas en español y evaluar los resultados obtenidos, para analizar los casos de errores de etiquetado y mejoras que se pueden implementar.

Herramientas utilizadas

Para el desarrollo de esta práctica he usado la librería Spacy de Python, diseñada para el procesamiento de lenguajes naturales. Concretamente con el etiquetador de entidades de esta librería, ya que Spacy posee modelos entrenados en español.

Estos tres modelos se basan en redes neuronales convolucionales entrenadas que se diferencian entre ellas principalmente por sus tamaños:

- ‘es_core_news_sm’
- ‘es_core_news_md’
- ‘es_core_news_lg’

En esta práctica he utilizado ***‘es_core_news_lg’*** y para los datos de prueba he trabajado con el texto de prueba del artículo esp.testb . Luego, para evaluar las métricas he hecho uso del fichero conlleval.py que han proporcionado.

Descripción de la Solución

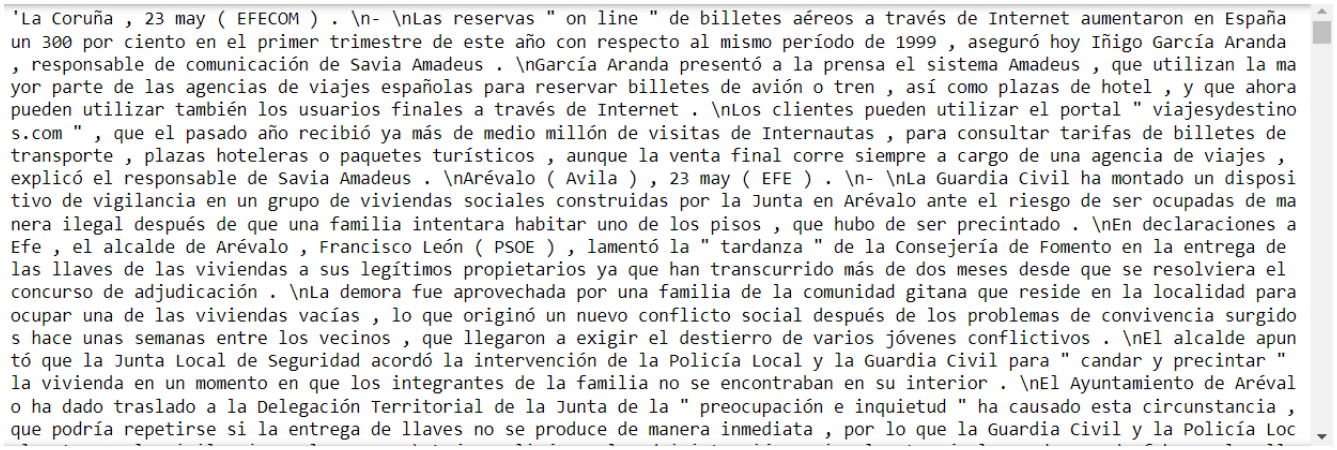
El código desarrollado consta de tres primeros pasos fundamentales para finalmente poder evaluar y analizar los resultados que obtenidos por etiquetador Spacy:

1. **Leer los datos del fichero** *esp.testb* **y transformar la información en estructura de datos.**

```
def get_raw_text(filename):
    # Leemos el fichero de prueba
    raw_text = open(filename, 'r')

    plain_text = '' #Texto Puro
    all_words_tags = [] # Lista de linea (palabra, categoría)

    for i,line in enumerate(raw_text):

        if not (line == '\n'):
            # Separamos el contenido de la linea y nos aseguramos de los espacios
            word, tag = line.strip().split()

            plain_text += word + ' '
            all_words_tags.append([word, tag])
        else:
            # Omitiremos en el diccionario los casos que son '\n'
            plain_text += line

    #Nos aseguramos que el texto completo no tenga espacios demás
    plain_text= plain_text

    # Cerramos el fichero
    raw_text.close()

    return plain_text, all_words_tags
```

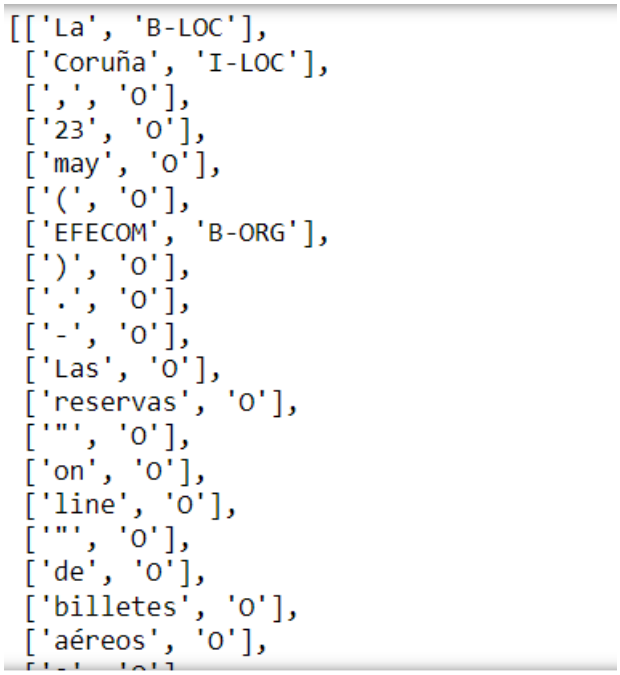
   He creado una función que recibe el nombre del fichero de prueba, lo abre y por cada línea evaluamos:
   a. Si es un salto de línea solo lo recolectamos en la cadena de caracteres que almacena el texto puro.
   b. Si no es un salto de línea separamos el contenido por espacios y obtenemos palabra y etiqueta. La palabra también la recolectamos en la cadena de caracteres, y también guardamos una lista (palabra, etiqueta) en una Lista

Finalmente obtenemos el texto puro:



```
'La Coruña , 23 may ( EFECOM ) . \n- \nLas reservas " on line " de billetes aéreos a través de Internet aumentaron en España
un 300 por ciento en el primer trimestre de este año con respecto al mismo período de 1999 , aseguró hoy Iñigo García Aranda
, responsable de comunicación de Savia Amadeus . \nGarcía Aranda presentó a la prensa el sistema Amadeus , que utilizan la ma
yor parte de las agencias de viajes españolas para reservar billetes de avión o tren , así como plazas de hotel , y que ahora
pueden utilizar también los usuarios finales a través de Internet . \nLos clientes pueden utilizar el portal " viajesydestino
s.com " , que el pasado año recibió ya más de medio millón de visitas de Internautas , para consultar tarifas de billetes de
transporte , plazas hoteleras o paquetes turísticos , aunque la venta final corre siempre a cargo de una agencia de viajes ,
explicó el responsable de Savia Amadeus . \nArévalo ( Avila ) , 23 may ( EFE ) . \n- \nLa Guardia Civil ha montado un disposi
tivo de vigilancia en un grupo de viviendas sociales construidas por la Junta en Arévalo ante el riesgo de ser ocupadas de ma
nera ilegal después de que una familia intentara habitar uno de los pisos , que hubo de ser precintado . \nEn declaraciones a
Efe , el alcalde de Arévalo , Francisco León ( PSOE ) , lamentó la " tardanza " de la Consejería de Fomento en la entrega de
las llaves de las viviendas a sus legítimos propietarios ya que han transcurrido más de dos meses desde que se resolviera el
concurso de adjudicación . \nLa demora fue aprovechada por una familia de la comunidad gitana que reside en la localidad para
ocupar una de las viviendas vacías , lo que originó un nuevo conflicto social después de los problemas de convivencia surgido
s hace unas semanas entre los vecinos , que llegaron a exigir el destierro de varios jóvenes conflictivos . \nEl alcalde apun
tó que la Junta Local de Seguridad acordó la intervención de la Policía Local y la Guardia Civil para " candar y precintar "
la vivienda en un momento en que los integrantes de la familia no se encontraban en su interior . \nEl Ayuntamiento de Aréval
o ha dado traslado a la Delegación Territorial de la Junta de la " preocupación e inquietud " ha causado esta circunstancia ,
que podría repetirse si la entrega de llaves no se produce de manera inmediata , por lo que la Guardia Civil y la Policía Loc
```

y una lista de valores:



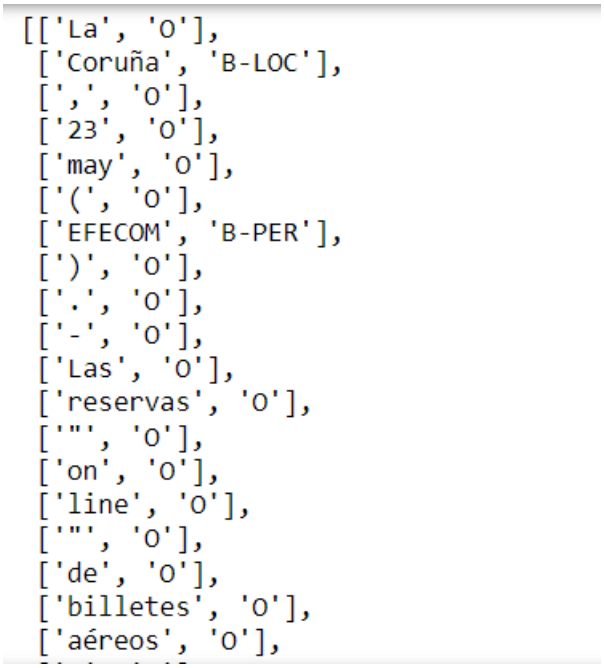
```
[['La', 'B-LOC'],
 ['Coruña', 'I-LOC'],
 [',', 'O'],
 ['23', 'O'],
 ['may', 'O'],
 ['(', 'O'],
 ['EFECOM', 'B-ORG'],
 [')', 'O'],
 ['.', 'O'],
 ['-', 'O'],
 ['Las', 'O'],
 ['reservas', 'O'],
 ['"', 'O'],
 ['on', 'O'],
 ['line', 'O'],
 ['"', 'O'],
 ['de', 'O'],
 ['billetes', 'O'],
 ['aéreos', 'O'],
```

Además de eso he comprobado que el total de líneas de nuestro fichero es de 51533

2. **Cargar el modelo Spacy, aplicarle el texto plano y almacenar los resultados en una estructura de datos.**

Creamos una instancia del modelo de spacy "*es_core_news_lg", que directamente evaluara el texto que hemos generado en el paso anterior.*

Una vez procesado el texto sobre el modelo he generado una lista de listas, es decir cada posición de la lista almacena una lista (palabra, etiqueta), la misma estructura que creamos para los datos de prueba. Para los casos en los que Spacy a etiquetado la palabra como Miscelánea, debemos omitir el parámetro *ent_type para obtener la etiqueta. La lista obtenida de Spacy, contiene* 51761 tokens lo que quiere decir que tiene mayores elementos que la original.



```
[['La', 'O'],
 ['Coruña', 'B-LOC'],
 [',', 'O'],
 ['23', 'O'],
 ['may', 'O'],
 ['(', 'O'],
 ['EFECOM', 'B-PER'],
 [')', 'O'],
 ['.', 'O'],
 ['-', 'O'],
 ['Las', 'O'],
 ['reservas', 'O'],
 ['"', 'O'],
 ['on', 'O'],
 ['line', 'O'],
 ['"', 'O'],
 ['de', 'O'],
 ['billetes', 'O'],
 ['aéreos', 'O'],
```

3. **Evaluar los datos originales con los obtenidos por el modelo y sacar las métricas.**

   Comparé la lista generada en base a los datos de Spacy con la lista del del Test.
   a. Si coinciden almacenamos la palabra, la etiqueta y el valor predicho en el fichero de salida
   b. Si no coinciden almacenamos la palabra que esperábamos por la que recibimos

   Luego nuestros valores de salida los interpreta con el fichero **conlleval**.**py** para obtener las medidas (precisión, recall, medida-f)

```python
def write_output_error_files(test_dict, spacy_dict):
    i = 0
    output_file = open('output'+number, 'w')
    error_file = open('error'+number, 'w')

    for token in test_dict:
        word, tag = token
        word_, pred = spacy_dict[i]

        if word.strip() == word_.strip():
            output_file.write(word + ' ' + tag + ' ' + pred + ' \n')
        else:
            error_file.write('Expected: '+ word +' Received:'+ word_ + '\n')
            i += 1

        i += 1
    error_file.close()
    output_file.close()
```

```
processed 51305 tokens with 3534 phrases; found: 4013 phrases; correct: 2318.
accuracy:  65.94%; (non-O)
accuracy:  90.78%; precision:  57.76%; recall:  65.59%; FB1:  61.43
              LOC: precision:  61.35%; recall:  75.65%; FB1:  67.75  1322
             MISC: precision:   9.72%; recall:  24.62%; FB1:  13.93  844
              ORG: precision:  74.19%; recall:  57.75%; FB1:  64.95  1085
              PER: precision:  81.36%; recall:  84.35%; FB1:  82.83  762
```

Conclusiones

Asumiendo que los datos de prueba extraídos del *esp.testb* presentaban 53049 líneas y al omitir los saltos de línea quedaba el conjunto en 51533 de los cuales coincidieron con lo obtenido en el modelo 51305 , por lo que los token omitidos no llegan a más del 0,44%

En líneas generales el modelo tiene una precisión alta y un medida-f por encima del 50%. Sin embargo, cuando evaluamos las etiquetas propiamente podemos c onsiderar que tiene mayor precisión etiquetar personas, y que tiene muy poca p recisión para las misceláneas, por lo que afecta el porcentaje precisión del mod elo .