Extracción de Información

Daniel de la Osa Fernandez D.OSA@ESTUDIANTES.MATCOM.UH.CU
Grupo C-411

Jose Luis Alvarez de la Campa J.ALVAREZ@ESTUDIANTES.MATCOM.UH.CU
Grupo C-412

Resumen

En este proyecto se presentan dos modelos para etiquetar las palabras de un texto según la parte de la oración a la que pertenecen, por ejemplo decir si es un sustantivo un adjetivo etc. El segundo es para identificar entidades en un documento, por ejemplo organizaciones, personas etc. El objetivo es con estos modelos es preparar el documento para sentar las bases para la extracción de información de este.

1. Introducción

Existen diferentes frameworks para lograr clasificar las palabras según su la parte de la oración a la cual pertenecen (POS-TAG) así como el reconocimiento de entidades (NER). Entre ellos encontramos NLTK, Spacy , Freeling y Stanford-NLP. A partir de lo hecho en estos frameworks y apoyándonos en ellos crearemos unos modelos que permitan llevar a cabo estas tareas , particularmente usando Spacy.

2. Preprocesamiento

Para lograr la tarea se utilizaron dos corpus de datos para el entrenamiento del modelo a los cuales se dividieron por oraciones que se tokenizaron y se les agregó a cada token la parte de la oración a la cual pertenecía, así como su clasificación de entidad (Persona , Organización, locación ,otros). De esta manera con el corpus anotado se crearon json para cargar la información la modelo y así comenzar su entrenamiento.

3. Modelo para Identificar Partes de la Oración

Para lograr este modelo usamos la arquitectura de la red neuronal de Spacy, tomando las características siguientes para lograr la inferencia después de varios ciclos de entrenamiento:

1. la palabra en sí
2. la palabra delante
3. la palabra detrás
4. si es la ultima de la oración
5. si es la última
6. si empieza con mayúsculas
7. si es en mayúscula completamente
8. si es en minúsculas completamente
9. prefijo de 1 , 2 y 3 letras
10. sufijo de 1 , 2 y 3 letras
11. si es un número

Con estas características elegidas se promedio a realizar varios ciclos de entrenamiento sobre los datos preparados hasta alcanzar un modelo con un acierto considerable del 90 %. Con este modelo se podría ahora dado una oración (String), deducir la parte de la oración de cada palabra que la conformaba.

4. Modelo para NER

Para lograr esto se uso una arquitectura basada en redes neuronales llamada BLSTM-CRF. Para preparar el conjunto de entrenamiento, que se conforma por oraciones donde cada palabra se le asocia su clasificación de NER, se llevaron las palabras a números y se rellenaron las secuencias para que tuvieran la misma longitud y esto conforma el vector de ejemplos a predecir; esto junto con el vector de etiquetas que no es mas que un vector de ceros donde en la posición asociada a la clasificación que tiene la palabra en el corpus de entrenamiento tiene un uno, de esta forma indicando que la palabra a la que ese vector de etiquetas pertenece tiene como etiqueta de verdad la correspondiente según el indice que tiene marcado como uno.

De esta manera no es necesario desarrollar un sistema de características complejo ya que mediante el campo aleatoria condicional se le dan los pesos a cada clasificación para cada palabra y a partir la BLSTM le da una puntuación a esta disposición de los pesos para luego el CRF vuelva ajustar los pesos mejorando cada vez mas la predicción de las etiquetas correspondientes para cada palabra.

5. Como Usar Modelos

Para usar los modelos ejecutar la libreta de jupyter TestingModels. Aqui pueden probarse diferentes documentos cambiando la variable "archivo a procesar" por la dirección del documento que se quiere pasar por los dos modelos.