Práctica 1.
Técnicas de procesamiento del lenguaje natural.
Máster IA - Universidad de Alicante

Borja Navarro Colorado

diciembre 2023

1. Tarea

1.1. Análisis formal (categorial y sintáctico)

Crea un *script* en Python que procese un texto amplio con las aplicaciones comunes del PLN clásico: lematización, *PoS_tagger*, *parser* de dependencias y sentidos de las palabras.

Para ello, utiliza el *pipeline* básico de la herramienta de PLN SpaCy[1], completado con NLTK[2] para acceder a WordNet.

Por pasos:

1. Crear un cuaderno COLAB vacío.
2. Importar SpaCy, descargar el módulo de idioma elegido e importar.
3. Abrir una única novela de la carpeta *level1* del corpus ELTeC (ver luego).
4. Procesar el XML y extraer los párrafos:
   - Los párrafos están marcados con la etiqueta "p".
   - Para procesar XML en python, se recomienda por su sencillez *Beautiful Soup*.[3]
   - Si la novela es muy larga, se puede procesar solo una parte (unos cuantos capítulos, por ejemplo).
5. Analizar el texto con el *pipeline* básico de SpaCy y extraer un CSV con

   "palabra, lema, categoria_gramatical, tipo de dependencia sintáctica, palabra de quien depende"

6. Crear un gráfico en COLAB donde se muestren la cantidad de nombres, adjetivos, verbos y adverbios de la novela o fragmento analizado.

[1] https://spacy.io/
[2] www.nltk.org
[3] https://beautiful-soup-4.readthedocs.io/en/latest/

1.2. Análisis léxico-semántico (WordNet con NLTK)

Extraer el *synset* más frecuente de cada palabra. WordNet ordena los sentidos de las palabras por frecuencia, por lo que el primero será siempre el más frecuente.

Por pasos:

1. En el CSV creado anteriormente, incorporar una nueva columna con el *synset* más frecuente de la palabra.
2. Extraer los 10 sentidos más frecuentes del corpus (el lema).
3. **Entrega**: enviar enlace del cuaderno COLAB al profesor mediante la opción "Entrega de prácticas" de *UA-Cloud*.

2. Herramientas

- Cuaderno COLAB para crear el código[4] y cuenta gCloud.
- Python (ya instalado en COLAB)
- Para análisis formal:
  - SpaCy,[5]. Ya instalado en COLAB, solo hay que importarlo.
- Para análisis semántico:
  - El *synset* se puede obtener desde el *Open Multilingual WordNet* mediante la herramienta NLTK (*Natural Language ToolKit*). Ver https://www.nltk.org/howto/wordnet.html
  - Para acceder a NLTK desde SpaCy, ver: https://spacy.io/universe/project/spacy-wordnet

3. Corpus

Para darle algo de interés a la práctica, vamos a utilizar el tipo de texto más complejo que existe: el texto literario. En concreto debéis analizar una novela del corpus ELTeC: https://github.com/COST-ELTeC.

ELTeC es un corpus multilingüe de novelas publicadas en Europa durante los siglos XIX y XX. Cada novela está anotada a tres niveles: básico, estándar y avanzado. **Debéis seleccionar una novela del nivel estándar (level 1).** A este nivel, cada novela está marcada en XML siguiendo el estándar TEI[6]. Podéis utilizar el idioma que queráis (siempre que se encuentre en SpaCy).[7] Por ejemplo:

[4] https://colab.research.google.com
[5] https://spacy.io/
[6] https://tei-c.org/
[7] Ver los modelos disponibles aquí: https://spacy.io/models.

- novelas en español: https://github.com/COST-ELTeC/ELTeC-spa/tree/master/level1
- novelas en inglés: https://github.com/COST-ELTeC/ELTeC-eng/tree/master/level1
- novelas en francés: https://github.com/COST-ELTeC/ELTeC-fra/tree/master/level1
- novelas en portugués: https://github.com/COST-ELTeC/ELTeC-por/tree/master/level1
- etc.

Dado que el corpus está en GitHub, se pude descargar el *raw* de la novela en COLAB con WGET, acceder mediante REQUESTS, o clonar directamente el repo y abrir el fichero que se quiera analizar.

4. Documentación

Documentación básica SpaCy. Aquí está explicado todo lo necesario sobre PLN para realizar el análisis formal:

https://spacy.io/usage/spacy-101

Más información sobre SpaCy:

- Curso avanzado: https://course.spacy.io/es/]
- Documentación oficial: https://spacy.io/usage

Sobre NLTK: ver https://www.nltk.org/ y lo comentado antes.

5. Amplicación (opcional)

Realizar análisis de dependencias del corpus con SpaCy y extraer en formato CONLL. Desde SpaCy se puede utilizar STANZA y UD-Pipe.
Documentación:

- https://spacy.io/universe/project/spacy-conll
- https://github.com/BramVanroy/spacy_conll