Jerónimo Molina Molina

Procesamiento del Lenguaje Natural

3. Técnicas básicas en NLP

Técnicas básicas de procesamiento del Lenguaje Natural

Hasta el momento hemos ido trabajando sobre algunos ejercicios de procesamiento del lenguaje natural, aplicando procesos básicos pero sin identificarlos, sin hacer mención a cada uno, como parte de un proceso, de un preprocesamiento de datos de forma natural, pero permíteme decirte que hay una serie de pasos, o de tratamientos con la data, que están identificados y que, tal es su importancia, que debemos explicarlos llegados a este punto, por lo que vamos a dedicar este tema 3 a explicar algunas técnicas básicas de NLP, tales como:

- Tokenización
- Lematización
- Elminación de StopWords y Puntuación
- PoS o Part of Speech Tagging
- Shallow parsing / Chunks
- Named Entity Recognition

¡Vamos a por ello!

Introducción

Comencemos dando una descripción sencilla de cada una de las técnicas mencionadas antes, de tal modo que se pueda enfocar, a nivel teórico, cada una de ellas. Verás rápidamente que alguna ya la hemos aplicado.

1. Tokenizar: Dividir un texto en unidades de menor tamaño, tales como párrafos, oraciones o palabras, de modo que se puedan procesar o analizar de forma independiente.
2. Named Entity Recognition -NER-: Tarea de identificar y categorizar entidades -tokens- nombradas en un texto, tales como lugares, personas, organizaciones u otros. Esta técnica permite relacionar tokens identificando su equivalencia en posiciones diferentes de un mismo texto.
3. Part of Speech Tagging -PoS-: Proceso de identificación y etiquetado del rol sintáctico de cada token en una oración o texto.
4. Shallow Parsing / Chunks: El análisis superficial, o fragmentación, es el proceso de extraer frases de un texto no estructurado. Esto implica fragmentar grupos de tokens adyacentes en frases en función de sus etiquetas PoS. Algunos ejemplos de fragmentos conocidos son frases nominales, frases verbales y frases preposicionales. A partir de los nombres o los verbos puede extraerse información acerca del texto, e incluso podría dibujarse un árbol de análisis sintáctico del mismo. Es una técnica que tiene aplicación directa en las bases de datos vectoriales, ya que un documento -y me refiero a documento en el sentido más multimodal posible- puede almacenarse por trozos, optimizando así el rendimiento de acceso a la información, no solo la precisión de los vectores generados.
5. Eliminar Stop Words y puntuación: Se pueden definir las Stop Words como aquellos términos del vocabulario que no aportan valor añadido (o un valor añadido elevado) de cara a solucionar el problema al que nos enfrentamos. Estos tokens, así como los signos de puntuación, pueden eliminarse como parte de la preparación de la data, de modo que se optimice y se simplifique la solución.
6. Lematizar: Es un proceso lingüístico que consiste en reducir sus palabras a su raíz (lema). De este modo se puede reducir la cardinalidad (tamaño del conjunto de términos) del vocabulario. Como ejemplo se puede hablar de “singularizar”, eliminar género, encontrar el infinitivo de un verbo, etcétera. Encontrar la raíz y sustituir el token por su raíz.

Tokenizar un texto y aplicación de otras técnicas sobre la marcha

Ya se ha descrito la tokenización, que debería ser, las más de las veces, la primera de las técnicas a aplicar en un pre-procesamiento de un proyecto NLP, siempre que la arquitectura de IA a emplear. no incorpore ya una capa de tokenización.

En este ejercicio, no solo vamos a tokenizar un texto, primero en frases y después en palabras, sino que aplicaremos otras técnicas relacionadas y que son de gran utilidad. La idea es muy sencilla, como podrás observar, y simplemente se trata de ir trabajando el texto hasta obtener un resultado que resulte adecuado.

Pongamos por ejemplo que deseamos identificar un subconjunto de tokens para utilizarlos como variables independientes al entrenar un clasificador, digamos que, empleando OneHot Encodding. Bien, pues en este caso deberían aplicarse este tipo de técnicas, hasta determinar el subconjunto más adecuado de tokens a emplear.

Puedes acceder al código desde: https://github.com/jmolina010/ejemplo_tokenizar_y_mas/

De este modo, podríamos entrenar un modelo clasificador a partir de un vocabulario en formato OneHot, de modo que sobre el input se realice siempre un preprocesamiento, previo a la predicción.

Ejemplo práctico, posible caso real:

Imaginemos que disponemos de una serie de incidencias, que se reciben por mail y que, deseamos clasificarlas por su naturaleza, que podría ser software, comunicaciones o hardware, y disponemos de una muestra suficientemente amplia y balanceada de incidencias ya clasificadas a mano.

Bien, como primer paso, podría hacerse una selección de tokens de las incidencias de cada tipo, aquellos que, por los mails de que disponemos, son los tokens propios de cada tipo de incidencia, de lo que podría resultar la siguiente conclusión:

Tipo	Tokens
Software	Windows, SAP, Informe, Contraseña, Configuracion
Comunicaciones	Red, Wifi, Servidor, Unidad, Remoto, Internet
Hardware	Pantalla, Ratón Teclado, Disco, USB

Con todos estos tokens se puede preparar un OneHot de entrenamiento en el proceso de elaboración del dataset, y luego, en tiempo de ejecución, antes de predecir, puede construirse con la data de entrada un One-Hot en tiempo real que sea con el que se solicite predicción al modelo.

Procesamiento NER y PoS. Tagging con Spacy

Es un buen momento para presentar, de forma muy general, la librería de código abierto Spacy, desarrollada con el objetivo de facilitar las tareas de procesamiento de lenguaje natural. De hecho, Spacy es una librería open source para NLP que, en comparación con NLTK (por Natural Language ToolKit) está mucho más orientada a la productividad y a la creación de pipelines. No obstante, en este ejemplo, vamos a emplearla de forma básica. La web oficial es https://spacy.io/, donde puedes encontrar numerosos recursos para profundizar en ella.

En este ejemplo, que puedes localizar en https://github.com/jmolina010/ejemplo_ner_spacy/, encontrarás la forma de aplicar NER y PoS con esta librería.

Ejercicio propuesto. Análisis del sentimiento

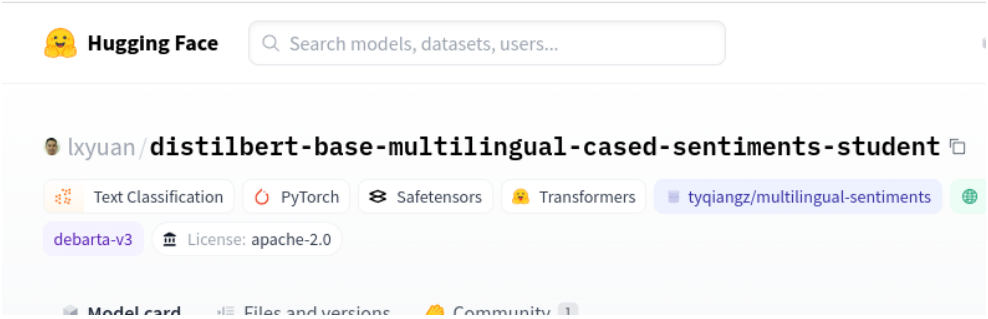
El análisis del sentimiento en un texto consiste, precisamente, en procesar ese texto y detectar si el sentimiento es negativo, neutro o positivo (por ejemplo, valorando ese texto en un rango [-1,1], siendo -1 negativo, 0 neutro y 1 positivo). Así pues, daremos una cierta holgura a cada valoración, estableciendo, por ejemplo, los siguientes criterios:

Sentimiento positivo	[1, 0.3]
Sentimiento neutro	[0.3, -0.3]
Sentimiento negativo	[-0.3, -1]

Para analizar el sentimiento podríamos entrenar un modelo de regresión propio, pero tendríamos que obtener un dataset lo suficientemente grande ya valorado, aplicar alguna de las técnicas básicas de preprocesamiento de texto que ya hemos aprendido y finalmente, entrenarlo.

Afortunadamente existen numerosas librerías que ya ofrecen esta funcionalidad. En esta ocasión vamos a abordar una solución con arquitectura transformer, que ya has estudiado, aunque más adelante la repasaremos por encima.

Si no conoces HuggingFace, te recomiendo que lo explores a fondo. De momento vamos a emplear un modelo preentrenado que analiza el sentimiento en diferentes idiomas, entre ellos, el español:



En la misma página del modelo encontrarás ejemplos de cómo explotarlo. Puedes encontrar el ejemplo completo, ya resuelto en:

https://github.com/jmolina010/sentimiento_transformer_ejemplo

Puedes ver que es un código realmente simple.

Te recomiendo, no obstante, que estudies a fondo la librería transformers (https://pypi.org/project/transformers/) así como las posibilidades que ofrece HuggingFace (https://huggingface.co/), donde, incluso, podrás hospedar tus propios modelos.

Ejercicio

A partir del ejemplo de transformers, explora el universo de modelos que ofrece hugging face y experimenta con ellos.

Bibliografía para completar el aprendizaje

1. Natural Language Processing with Python. Ed. O'Reilly. Aut: Steven Bird et al
2. https://spacy.io/
3. https://pypi.org/project/transformers/
4. https://huggingface.co/