Jerónimo Molina Molina

Procesamiento del Lenguaje Natural

4. Ejercicio de preprocesamiento de textos en PLN

Preprocesamiento de textos

En este tema exploraremos diferentes librerías Python para procesamiento del lenguaje natural. Algunas ya las hemos utilizado, si bien vamos a abordar de nuevo.

De nuevo quiero insistir en que no vas a encontrar una referencia o tutorial completo a ninguna librería como parte de esta asignatura (sí en referencias, por supuesto), pero, en un lenguaje de alto nivel como Python, con tantísimos millones de colaboradores a lo largo y ancho del planeta, es lógico que surjan constantemente diferentes librerías para procesamiento de textos, algunas de las cuales ya hemos empleado en termas anteriores (por ejemplo, gensim para tratamiento de embeddings), por lo que este tema pretende emplear, nuevamente, alguna de ellas, y que, a partir del mismo, el alumno profundice en las mismas en mayor o menor medida, en función de cuál le resulte más interesante o de las funcionalidades que en cada caso necesite. Lo que si es muy importante es habituarse a trabajar en un entorno abierto como Python, desarrollando para ello la habilidad de familiarizarse rápidamente con una librería nueva.

Probablemente a estas alturas ya eres muy consciente de que si algo importa al construir modelos de inteligencia artificial (y me da lo mismo que sean de aprendizaje automático tradicional, de aprendizaje profundo o de aprendizaje por refuerzo) son los datos -luego, elegir un algoritmo, ajustas sus parámetros y optimizar los hiperparámetros para generar el modelo más eficiente y preciso, es algo, totalmente automatizable y sistemático, por algo han surgido librerías de AutoML-. Pues bien, en procesamiento del lenguaje natural esto cobra, si cabe, algo más de relevancia.

Por ello quiero centrar los dos objetivos más importantes de este tema:

1. Aprender un método base para preprocesar datos en PLN. Este método no siempre tiene que aplicarse al pie de la letra, deberás adaptarlo a cada problema a partir del sentido común, deberás desarrollar la habilidad de ajustar cada paso del preprocesamiento de la data a tu caso de uso en cada momento.
2. Aprovechar este ejemplo y el ejercicio propuesto para familiarizarte más aún con NLTK y Spacy, dos de las librerías más famosas en NLP.

Por otro lado, es posible que no termines de encontrarle sentido a técnicas con NER o PoS. Probablemente de traslade a aquella época de tus años de colegio o instituto en la que el profesor o profesora de lengua te proponía ejercicios de análisis sintáctico, … y bueno, que tú, como humano, comprendas la estructura gramatical de una oración, pues bueno, está bien, pero una máquina, aparentemente, no necesita estos datos para la realización de otras tareas. Veamos, para disipar dudas, no uno, ni dos, sino tres casos de uso en los que sí resulta interesante:

1. Sumarización: Un modelo del lenguaje que asuma tareas de sumarización necesita, sin lugar a duda, comprender el contexto, y para ello le resultará esencial identificar si “rojo” es un sustantivo o un calificativo. Lo mismo ocurre con NER, resulta muy útil al modelo identificar lugares, personas etc. e incluso mediante NEL (Named Entity Link) proporcionar al humano información adicional. *Named Entity Linking permitiría, por ejemplo, al identificar una expresión como “el desierto del Sáhara”, “la bolsa de New York” o “Esclerosis Múltiple Amiotrófica”, proporcionar enlaces para ampliar información al respecto*.
2. Q-A: La tarea Question Answering consiste en que un modelo del lenguaje procesa (es decir, comprende, analiza, memoriza…) un texto y, después, si el modelo es interrogado, es capaz de responder sobre la base de lo aprendido (¿te recuerda a Chat-GPT?)…. Bien, pues igualmente, debe ser capaz de diferenciar entre diferentes estructuras y elementos gramaticales para ofrecer una precisión y comportamiento adecuados.

3. PoS y NER como apoyo en tareas IA, mejorando la explicabilidad de otros modelos. Imagina un sistema IA que (actualmente, entre otros, estoy involucrado en un proyecto de estas características, aunque de mayor complejidad respecto a lo expuesto aquí),a partir de un texto, deba indicar si el mensaje incumple ciertas normativas. Estas "sentencias" debería revisarlas un experto, por lo que un análisis PoS y NER resultan una ayuda inestimable para que el experto comprenda los motivos de la clasificación realizada por el sistema.

Ejercicio práctico

A continuación, abordaremos un ejercicio de preprocesamiento de un dataset. El objetivo es preparar el dataset para entrenar un modelo predictivo, y partiremos de un conjunto de datos muy conocido, llamado "review_filmaffinity.csv".

No vamos a generar un modelo, pues eso es algo que corresponde a asignaturas más enfocadas en el machine learning, pero si que vamos a llegar al punto de generar un dataset a partir del que comenzar el proceso de entrenamiento, centrándonos, eso sí, en el análisis de los tokens.

Las diferentes librerías que abordaremos a lo largo del ejercicio y del tema son

1. NLTK: Traducido como "Kit de Herramientas para el lenguaje natural" o Natural Language ToolKit. Una librería que se remonta, en su primera versión compatible con Python 2.4 al año 2005, probablemente la librería para PLN más antigua que sigue en activo.
2. Spacy: Librería para PLN desarrollada en 2015, escrita en Python y Cython

El uso de una librería o de otra en este tema es completamente aleatorio sin que utilizar una más que otra deba considerarse, en absoluto, como una preferencia o recomendación del profesor.

Descripción del problema

Disponemos de un dataset que contiene críticas de 119.003 películas del periodo de 1900 a 2020. Los atributos del dataset son:

- Título: film_name
- Género: genre
- Puntuación promedio: film_avg_rate
- Puntuación asociada a la crítica: review_rate
- Título de la crítica: review_title
- Crítica: review_text

Objetivo: Generar un modelo predictivo que genere una puntuación provisional (sugerida) asociada a la crítica, a partir del texto de la misma. Nota: Nos vamos a limitar al preprocesamiento de la data.
Se parte de un dataset ya limpio y válido, que podrás encontrar en el repositorio git, junto al cuaderno del ejercicio. Debido a esto, las tareas básicas de investigación y limpieza de la data, vamos a saltarlas, centrándonos en los pasos que realmente tienen interés desde el punto de vista de la PNL.

Planteamiento de la solución

El objetivo es, como sabemos, preparar el dataset para que se puedas utilizar en el entrenamiento de un modelo, por lo que a continuación vamos a abordar los siguientes puntos:

1. Lectura del fichero, identificación y selección de columnas
2. Lectura de las críticas, tokenización, eliminación de signos de puntuación y stop words, generación de columna nueva y guardado de nueva versión del fichero
3. Determinación de frecuencia de aparición de tokens, y posible eliminación de tokens menos frecuentes. *Nota: Esta tarea, dependiendo del problema, puede abordarse en este momento o, quizá como parte el punto 5, pero debido a la naturaleza didáctica de este caso, y al elevado volumen de tokens se va a abordar, inicialmente, en este punto.*
4. Sustitución de sinónimos de tokens relevantes
5. Lematización de verbos
6. Planteamiento de franjas para selección de tokens y selección de tokens por franja
7. Generación de codificación OneHot

En cada paso, generaremos un fichero csv nuevo con el resultado de las tareas realizadas.

Puedes descargar el código y el dataset inicial del ejercicio en: https://github.com/jmolina010/ejemplo_preprocesamiento_textos

Conclusiones del ejercicio.

Con el trabajo realizado se han ilustrado diferentes técnicas o pasos para preprocesar textos de modo que se pueda generar un modelo de aprendizaje automático.

Debe tenerse en cuenta que, en producción, cuando se reciba una crítica, se deberá realizar el mismo preprocesamiento del texto y generar una lista de ceros y de unos indicando si en el texto está presente cada uno de los tokens que se han seleccionado.

Al tratarse de un ejemplo didáctico se han omitido algunos pasos, así como, y esto se ha explicado en el mismo ejercicio, algunos otros se han simplificado.

A continuación, se proponen algunos ejercicios basados en el explicado.

Laboratorio propuesto

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, supongamos que deseamos valorar 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:

Ejercicios propuestos

1. Celda 12
   - Revisa la celda 12 del cuaderno. Repasa detalladamente la lista de tokens cortos y elige todos aquellos que no deban eliminarse. Modifica el código y vuelve a ejecutar el cuaderno hasta esa celda.
   - Revisa ahora dict_tokens, manualmente y de forma detallada. Elimina aquellos tokens que son irrelevantes para la valoración de una crítica.

- Revisa nuevamente dict_tokens y reagrupa por sinónimos. *Ayuda: Revisa los tokens y determina cuáles son "equivalentes". Ten mucho cuidado con ello. A continuación, reagrupa esos tokens en el diccionario, sumando las apariciones de todos y poniendo el resultado bajo uno de los tokens, elimina las otras entradas (tokens)*

✔ ¿Cómo influyen estos cambios al dataset generado?

✔ ¿Qué consecuencias crees que pueden tener estos cambios en la generación del modelo?

2. Celdas 16 y 17
   - La reducción por número de apariciones es muy subjetiva. Se ha realizado pensando en un modelo de regresión. ¿Por qué piensas que se han fijado los límites superiores e inferiores de este modo pensando en regresión?
   - Si fueras a generar un modelo clasificador, ¿qué tokens eliminarías en función de su frecuencia de aparición? Justifica tu respuesta
3. Celda 18
   - Revisa los tokens de cada grupo en detalle. Elimina de la lista aquellos que, en cada grupo, por su puntuación, son significativos.
4. Generación de un modelo
   - Con todos los cambios realizados hasta ahora, genera de nuevo el dataset de entrenamiento.
   - Entrena un modelo de regresión, o, si lo prefieres, de clasificación.
   - Ahora aplica antes de entrenar el modelo un algoritmo de reducción de características y genéralo de nuevo. ¿Observas diferencias en el resultado de los modelos? ¿Cuáles?
5. Ahora a la predicción de un modelo de regresión, vamos a hacerle algún cambio:
   - Emplea un modelo de análisis del sentimiento con las críticas, y utiliza el resultado para ponderar la predicción de tu modelo. ¿Afecta en algo? ¿Qué conclusiones extraes?
   - Finalmente, olvida el modelo generado por un momento, y trata de extrapolar el análisis del sentimiento a una valoración entre 0 y 10. compara resultados entre esta extrapolación, tu modelo y tu modelo tras haberlo ponderado con análisis del sentimiento. ¿Qué diferencias observas? ¿Qué conclusiones extraes?

Bibliografía para completar el aprendizaje

1. Libro: Natural Language Processing with Python. Ed. O'Reilly. Aut: Steven Bird et al.
2. https://www.nltk.org/
3. https://radimrehurek.com/gensim/