INF335 - Tecnologías de Búsqueda Web

Tarea 1

Universidad Técnica Federico Santa María

Objetivos:

- Implementar y analizar la herramienta de python NLTK (Natural Language Tookit) para trabajar con procesamiento de texto y lenguaje natural.

- Estudiar e implementar las estructuras de datos adecuadas para representar un corpus, documentos y palabras con su categorización correspondiente.

Dataset : Amazon Fine Food Review

Para esta tarea se va a trabajar con el dataset de *"Amazon Fine Food Review"* el cual contiene más de 500.000 críticas de platos de comida y restaurants provenientes de Amazon. El archivo consiste en un .csv ("Comma Separate Values") el cual contiene la siguiente estructura:

1. **Id** - Id único de cada reseña
2. **ProductId** - Id único que identifica el producto a analizar
3. **UserId** - Id único que identifica al usuario
4. **ProfileName** - Nombre del usuario que realizó la reseña
5. **HelpfulnessNumerator** - número de usuarios que indicaron que encontraron esta crítica util
6. **HelpfulnessDenominator** número de usuarios que indicaron que encontraron esta crítica util -
7. **Score** - Rating, con valores entre 1 y 5 estrellas
8. **Time** - timestamp for the review
9. **Summary** - breve resumen de la reseña
10. **Text** - string que contiene la reseña

Link Descarga Dataset:

https://drive.google.com/open?id=0B1GNvIDVzwwLR2dwQVliRnBWMnM

Objetivo: Extraiga del documento el item "Text" y generé un corpus , almacenando en un string todas las reseñas del dataset . Usará esta variable para realizar las siguientes etapas de preprocesamiento de texto.

Preprocesamiento:

1. Si observa el corpus, se dará cuenta de que hay etiquetas *html* embebidas en algunas reseñas. Para eliminar estas etiquetas , use la libreria *Beautiful Soup* (link: https://www.crummy.com/software/BeautifulSoup/bs4/doc/).

2. Convierta el corpus , de modo que solo existan minúsculas (lowercase).

3. Usando la lista de stopwords ortorgada por nltk, elimine aquellas palabras que sean clasificadas como stopwords, es decir, aquellas palabras que poseen poco contexto léxico y no otorgan información relevante.

4. Elimine las palabras que aparezcan en el corpus con una frecuencia inferior a un umbral definido (ejemplo : inferior a 3) (para ello, es recomendable determinar previo la frecuencia de cada término usando un diccionario).

5. Usando nltk, determine los Top-30 collocations mas relevantes del corpus, usando Bigramas .Implemente la función BigramAssocMeasures() y BigramCollocationFinder.from_words(). Recuerde que para este punto el corpus debe estar tokenizado. (mirar documentación).

6. Usando la libreria incorporada en nltk, implemente Stanford POS tagger para categorizar y obtener los tags de cada token del corpus usando Part-Of-Speech Tagger (POSTagger).

7. Usando la libreria incorporada en nltk, implemente Named Entity Recognition (NER) con Stanford NER Tagger. Analice y describa sus resultados.

8. **Sentiment Analysis** : Implemente usando la libreria *Vader* incorporada en nltk para analizar la polaridad del corpus ,determinar cada documento (para ello es necesario re-estructurar el corpus como un array de documentos, o sentencias):

    1. Tokenizar el corpus a nivel de sentencia (recuerde incorporar el preprocesamiento previo).
    2. Para cada sentencias (reseña) , implemente Vader para determinar la polaridad.

9. Usando nltk, determine los Top-30 collocations mas relevantes del corpus, usando Bigramas .Implemente la función BigramAssocMeasures() y BigramCollocationFinder.from_words(). Recuerde que para este punto el corpus debe estar tokenizado. (mirar documentación).

10. Usando la libreria incorporada en nltk, implemente Stanford POS tagger para categorizar y obtener los tags de cada token del corpus usando Part-Of-Speech Tagger (POSTagger).

11. Usando la libreria incorporada en nltk, implemente Named Entity Recognition (NER) con Stanford NER Tagger. Analice y describa sus resultados.

12. **Sentiment Analysis** : Implemente usando la libreria *Vader* incorporada en nltk para analizar la polaridad del corpus ,determinar cada documento (para ello es necesario re-estructurar el corpus como un array de documentos, o sentencias):

    1. Tokenizar el corpus a nivel de sentencia (recuerde incorporar el preprocesamiento previo).
    2. Para cada sentencias (reseña) , implemente Vader para determinar la polaridad.

Notas

- Para varias etapas del preprocesamiento, usará diferentes librerias disponibles en Python. Se recomienda usar el instalador de paquetes *pip* (link: https://pypi.python.org/pypi/pip) .
- Algunos de estos pasos del preprocesamiento pueden demorar en compilar (en algunos casos sobre 45 min, dependiendo de la máquina), por lo que es recomendable ir guardando el estado del corpus su posterior uso. Para estos casos se recomienda usar la libreria pickle en python (link: https://docs.python.org/2/library/pickle.html)

Documentación y Ejemplos

- Beautiful Soup : https://www.crummy.com/software/BeautifulSoup/
- Bigrams and Collocations: http://www.nltk.org/howto/collocations.html
- Stanford PoS Tagger : http://www.nltk.org/api/nltk.tag.html#module-nltk.tag.stanford
- Stanford 'Tagger' Link Download : https://nlp.stanford.edu/software/tagger.shtml#Download
- Stanford Ner Tagger: https://pythonprogramming.net/named-entity-recognition-stanford-ner-tagger/
- Sentiment Analysis with Vader: http://www.nltk.org/howto/sentiment.html

Instrucciones

1. El informe debe entregarse en un archivo jupyter notebook (diferente a este) con el código implementado y los análisis correspondientes. El informe debe subirse en la plataforma oficial de moodle en formato comprimido (.zip) con el nombre*tarea1_rol.zip*

2. Todas las consultas serán atendidas por el canal de consultas de moodle.

3. La fecha de entrega es el dia **10 de Abril** . Pasada esa fecha se descontaran 20 puntos por dia.