Escuela Politécnica Superior

UAM
Universidad Autónoma de Madrid

Máster Universitario en Ciencia de Datos

Procesamiento del Lenguaje Natural 2023-24

Práctica de laboratorio 1: Análisis de Sentimientos en Novelas Clásicas

Entrega: 17 de marzo de 2024
Modelo de trabajo: la práctica se abordará y entregará por parejas

---

En este ejercicio, trabajaremos con el Procesamiento del Lenguaje Natural (PLN) para analizar las emociones expresadas en textos literarios disponibles en Project Gutenberg. El objetivo es construir un sistema que pueda identificar y contar las emociones presentes en estas obras. Este ejercicio se enfoca en el uso de técnicas avanzadas de PLN, incluyendo el análisis de sentimientos, la extracción de información de texto y el procesamiento de lenguaje natural en general.

Para llevar a cabo este ejercicio, se te proporcionará acceso a una serie de recursos y herramientas, incluyendo el léxico de sentimientos NRC (National Research Council), la base de datos léxica WordNet, y la biblioteca de Python Beautiful Soup.

Recursos:

1. **Léxico de Emociones:** Se trata de una colección de palabras y sus asociaciones con emociones o sentimientos. En este ejercicio, utilizaremos el léxico de sentimientos NRC (National Research Council)[1]. Se trata de un recurso ampliamente reconocido en el campo del PLN para el análisis de sentimientos. Cuenta con 14,182 unigramas (palabras) con las categorías de sentimientos positivo y negativo, así como las emociones de enfado, anticipación, disgusto, miedo, alegría, tristeza, sorpresa y confianza. Está disponible en más de cien idiomas mediante traducción automática.

2. **WordNet:** Es una base de datos léxica en inglés que agrupa palabras en conjuntos de significados relacionados llamados sinónimos léxicos conocidos como *synsets*. En el contexto de esta práctica, puede usarse para extender el léxico NRC mediante la inclusión de sinónimos, hipónimos e hiperónimos de las palabras ya presentes en el léxico. La intención es permitir capturar una gama más amplia de expresiones emocionales y mejorar la precisión del análisis de sentimientos.

3. **Project Gutenberg:** Es una biblioteca digital que ofrece miles de libros electrónicos gratuitos. Es una excelente fuente de textos literarios gratuitos debido a su amplia variedad de obras y a la facilidad con la que se pueden descargar y utilizar para fines educativos y de investigación.

4. **Beautiful Soup:** Es una biblioteca de Python para extraer datos de archivos HTML y XML. Se utiliza en este ejercicio para descargar el texto de las novelas de Project Gutenberg y limpiarlo para el análisis. Las páginas web de Project Gutenberg contienen el texto dentro de etiquetas HTML, y Beautiful Soup permite extraer ese texto de manera eficiente y prepararlo para el análisis de sentimientos.

[1] https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm

Tareas:

1. **(1.5 puntos)** Cargar en una estructura de datos Python el léxico de sentimientos NRC (National Research Council). Se puede descargar el léxico de sentimientos NRC desde el siguiente enlace: https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm. Asegurarse de entender cómo se estructura el léxico y cómo se mapean las palabras a las emociones. Tener en cuenta que hay varios ficheros con la misma información: un fichero con toda la información, un fichero por emoción, etc. Elegir la opción que se estime oportuna. Considerar cómo organizar el léxico en memoria para un acceso rápido durante el análisis.

2. **(3.0 puntos)** Extender el léxico NRC utilizando WordNet (https://www.nltk.org/howto/wordnet.html) para incluir sinónimos, hipónimos, hiperónimos de las palabras ya presentes en el léxico. También puedes usar la función *derivationally_related_forms()* de WordNet de modo que el léxico pueda extenderse más aún. Esta función devuelve una lista de formas derivadas de una palabra, como plurales, participios pasados, etc. Esto puede ser útil para encontrar variaciones de una palabra que puedan estar asociadas con la misma emoción. El léxico deberá implementarse como un diccionario Python que tenga como clave una dupla `<lemma, POS-tag>`, y como valor la lista de emociones con las que dicha dupla se podría asociar.

   Para poder usar NLTK y WordNet deberás instalar NLTK con `pip` y luego importarlas y cargarlas del siguiente modo:

```
from nltk.corpus import wordnet as wn
import nltk
nltk.download('wordnet')
```

   Por otra parte, dado que la codificación de POS-*tagging* que emplea WordNet no es la del PennTreeBank, para hacer traducciones entre una y otra nomenclatura, se puede emplear los siguientes diccionarios.

```
wordnet_to_penn = {
    'n': 'NN', # sustantivo
    'v': 'VB', # verbo
    'a': 'JJ', # adjetivo
    's': 'JJ', # adjetivo superlativo
    'r': 'RB', # adverbio
    'c': 'CC' # conjunción
}

penn_to_wordnet = {
    'CC': 'c',   # Coordinating conjunction
    'CD': 'c',   # Cardinal number
    'DT': 'c',   # Determiner
    'EX': 'c',   # Existential there
    'FW': 'x',   # Foreign word
    'IN': 'c',   # Preposition or subordinating conjunction
    'JJ': 'a',   # Adjective
    'JJR': 'a', # Adjective, comparative
    'JJS': 'a', # Adjective, superlative
    'LS': 'c',   # List item marker
    'MD': 'v',   # Modal
    'NN': 'n',   # Noun, singular or mass
    'NNS': 'n', # Noun, plural
    'NNP': 'n', # Proper noun, singular
    'NNPS': 'n', # Proper noun, plural
```

```
    'PDT': 'c', # Predeterminer
    'POS': 'c', # Possessive ending
    'PRP': 'n', # Personal pronoun
    'PRP$': 'n', # Possessive pronoun
    'RB': 'r',   # Adverb
    'RBR': 'r', # Adverb, comparative
    'RBS': 'r', # Adverb, superlative
    'RP': 'r',   # Particle
    'SYM': 'x', # Symbol
    'TO': 'c',   # to
    'UH': 'x',   # Interjection
    'VB': 'v',   # Verb, base form
    'VBD': 'v', # Verb, past tense
    'VBG': 'v', # Verb, gerund or present participle
    'VBN': 'v', # Verb, past participle
    'VBP': 'v', # Verb, non-3rd person singular present
    'VBZ': 'v', # Verb, 3rd person singular present
    'WDT': 'c', # Wh-determiner
    'WP': 'n',   # Wh-pronoun
    'WP$': 'n', # Possessive wh-pronoun
    'WRB': 'r', # Wh-adverb
    'X': 'x'     # Any word not categorized by the other tags
    }
```

3. **(1.5 puntos)** Cargar el texto de novelas clásicas disponibles en Project Gutenberg. Utilizar la biblioteca Beautiful Soup para extraer el texto de las páginas HTML y prepararlo para el análisis.

   El siguiente es un diccionario con 10 novelas conocidas que están accesibles en Project Gutenberg que puedes usar en tu código.

```
books = {
    'Moby Dick': 'http://www.gutenberg.org/files/2554/2554-0.txt',
    'War and Peace': 'http://www.gutenberg.org/files/2600/2600-0.txt',
    'Pride and Prejudice': 'http://www.gutenberg.org/files/1342/1342-0.txt',
    'Crime and Punishment': 'http://www.gutenberg.org/files/2556/2556-0.txt',
    'The Adventures of Sherlock Holmes': 'http://www.gutenberg.org/files/1661/1661-
0.txt',
    'Ulysses': 'http://www.gutenberg.org/files/4300/4300-0.txt',
    'The Odyssey': 'http://www.gutenberg.org/files/16133/16133-0.txt',
    'The Divine Comedy': 'http://www.gutenberg.org/files/15/15-0.txt',
    'Fortunata y Jacinta': 'https://www.gutenberg.org/files/1342/1342-h/1342-
h.htm',
    'Critias': 'https://www.gutenberg.org/files/1571/1571-h/1571-h.htm'
}
```

   Para utilizar Beautiful Soup en el contexto del ejercicio, primero hay que instalar la biblioteca con `pip install beautifulsoup4`. Luego, se puede usar el siguiente fragmento de código para descargar y limpiar el texto de una novela:

```
import requests
from bs4 import BeautifulSoup

def download_text(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.text, 'html.parser')
    text = soup.get_text()
    return text```
```

4. **(3.0 puntos)** Implementar una función para analizar el texto y contar las ocurrencias de palabras vinculadas con emociones en el texto. Esta función debe:

    - Leer el texto y dividirlo en palabras individuales (tokenización).
    - Asignar a cada palabra su correspondiente etiqueta de parte del discurso (POS tagging) para diferenciar entre verbos, sustantivos, adjetivos, etc.
    - Lemmatizar las palabras para reducirlas a su forma base (por ejemplo, "*running*" a "*run*").
    - Comparar cada dupla `<lemma, POS-tag>` con las entradas en el léxico extendido para determinar la emoción asociada.
    - Contar las ocurrencias de cada emoción en el texto y generar un informe detallado.

    Solo por si se necesita a modo de soporte, es posible que la implementación a realizar deba hacer los siguientes `import` y deba descargar (`download`) los siguientes recursos de NLTK:

```
from nltk.corpus import wordnet as wn
from nltk import pos_tag
from nltk.tokenize import word_tokenize
from nltk.stem.wordnet import WordNetLemmatizer
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
```

5. **(1.0 puntos)** Presentar los resultados del análisis de sentimientos en las novelas clásicas. Incluir estadísticas sobre las emociones más comunes y cualquier patrón interesante que hayas observado. Considerar cómo visualizar los datos y cómo explicar las conclusiones del análisis solicitado.

Entregable.

El entregable de las tareas de laboratorio consiste en los siguientes archivos:

- nlp2324-lab1-XX.ipynb: un jupyter notebook con el código Python desarrollado. Este código debe estar bien comentado para facilitar su comprensión
- nlp2324-lab1-XX.pdf: un breve informe en el que se enumeran las tareas realizadas, se discuten las principales cuestiones de interés y (si procede) se informan los resultados.

Enviar estos archivos a través de Moodle en un archivo zip llamado nlp2324-lab1-XX.zip, donde XX debe sustituirse por el identificador del equipo, por ejemplo, 01, 02, ...