Escuela Politécnica Superior

UAM
Universidad Autónoma de Madrid

Máster en Bioinformática y Biología Computacional

Minería de texto. Curso 2025-26

Práctica de laboratorio: Análisis de Emociones y Sentimiento

Entrega: 3 de noviembre de 2025 a través de Moodle
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 y sentimientos presentes en estas obras. Este ejercicio se enfoca en el uso de técnicas avanzadas de PLN, 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 emociones NRC (National Research Council), la base de datos léxica WordNet, y la biblioteca de Python Beautiful Soup.

Recursos:

1. **Natural Language Toolkit (NLTK)[1]:** Es una biblioteca de Python muy empleada en PLN. Permite realizar tareas como tokenización, POS-tagging, análisis sintáctico, entre otras.

2. **Léxico de Emociones:** Se trata de una colección de palabras y sus asociaciones con emociones. En este ejercicio, utilizaremos el Word-Emotion Association Lexicon del NRC (National Research Council)[2], también conocido como EmoLex. Se trata de un recurso ampliamente reconocido en el campo del PLN. 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[3]:** 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, se usará 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.

3. **Project Gutenberg[4]:** Es una biblioteca digital que ofrece miles de libros electrónicos gratuitos. Se trata de 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.

---

[1] https://www.nltk.org/
[2] https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm
[3] https://wordnet.princeton.edu/
[4] https://www.gutenberg.org/

Tareas:

1. **(1.5 puntos)** Cargar en una estructura de datos Python el Word-Emotion Association Lexicon del NRC[5]. Asegúrate de entender cómo se estructura el léxico y cómo se mapean las palabras a las emociones. Hay que tener en cuenta que existen varios ficheros con la misma información: un fichero con toda la información, un fichero por emoción, etc. Se puede elegir la opción que se estime oportuna. Se deberá considerar cómo organizar el léxico en memoria para un acceso rápido durante el análisis.

2. **(3.0 puntos)** Extender EmoLex utilizando WordNet desde NLTK[6] 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
```

[5] https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm
[6] https://www.nltk.org/howto/wordnet.html

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

```
books = {
    'Crime and Punishment ': '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',
    'Frankenstein': 'https://www.gutenberg.org/cache/epub/84/pg84.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': 'https://www.gutenberg.org/cache/epub/1727/pg1727.txt',
    'Moby Dick': 'http://www.gutenberg.org/files/15/15-0.txt',
    'The Divine Comedy': 'https://www.gutenberg.org/cache/epub/8800/pg8800.txt',
    'Critias': 'https://www.gutenberg.org/cache/epub/1571/pg1571.txt'
}
```

Se puede usar el siguiente fragmento de código para descargar el texto de una novela:

```
import requests

def download_text(url):
    """Descarga el texto de una novela en formato txt."""
    try:
        response = requests.get(url)
        response.raise_for_status()  # Lanza excepción para códigos HTTP 4xx/5xx
        return response.text
    except requests.exceptions.RequestException as e:
        print(f"Error al descargar el texto: {e}")
        return None
```

4. **(3.0 puntos)** Implementar una función para analizar el texto de las novelas de forma básica, contando 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.
    - Lematizar las palabras para reducirlas a su forma base (por ejemplo, "*running*" a "*run*").
    - Comparar cada dupla `<lema, 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 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 estas tareas de laboratorio consistirá en un archivo Jupiter Notebook con el código Python desarrollado:

- El código debe estar bien comentado para facilitar su comprensión.
- Se deberá incluir una breve explicación en cada paso aclarando las tareas realizadas, discutiendo las principales cuestiones de interés y (si procede), informar de los resultados.

Deberá enviase el archivo nombrado como mintex2526-lab1-XX_YY.ipynb a través de Moodle, donde XX e YY deben sustituirse por el apellido y nombre de cada integrante de la pareja, por ejemplo: mintex2526-lab1-RodriguezAntonio_VelazquezPedro.ipynb