Taller 1: Preprocesamiento de textos & exploración y extracción de elementos de lenguaje

Técnicas de Procesamiento de Lenguaje Natural
Doctorado en Ingeniería - Maestría en Ingeniería Eléctrica
Maestría en Ingeniería de Sistemas - Maestría en Investigación Operativa y Estadística
Universidad Tecnológica de Pereira
Profesor: Julián David Echeverry Correa
Contacto (email): jde@utp.edu.co
Primer semestre de 2025

1 Objetivo

Este taller tiene como propósito consolidar los fundamentos teóricos abordados en el curso de Técnicas de Procesamiento de Lenguaje Natural y desarrollar competencias prácticas en el uso de herramientas computacionales para el análisis textual. Los objetivos específicos de este taller son:

- **Análisis de corpus textuales:** Examinar las características fundamentales de un corpus (formato, dimensión, estructura y codificación) para seleccionar metodologías óptimas de procesamiento.
- **Preparación de datos textuales:** Implementar técnicas de preprocesamiento y depuración de corpus textuales para garantizar la calidad de los análisis subsecuentes.
- **Procesamiento lingüístico básico:** Aplicar operaciones esenciales como tokenización, *stemming*, lematización y filtrado de palabras vacías (*stop words*) en documentos.
- **Análisis morfosintáctico:** Ejecutar etiquetado gramatical (*POS tagging*) y análisis sintáctico para identificar estructuras lingüísticas relevantes para el análisis textual.
- **Extracción de patrones léxicos:** Desarrollar e implementar métodos para la identificación y extracción de colocaciones léxicas en español, aplicando diversas métricas estadísticas que evalúen la fuerza asociativa entre palabras.
- **Modelado estadístico del lenguaje:** Construir modelos de lenguaje mediante el cálculo de frecuencias y probabilidades de aparición de palabras y secuencias (n-grams), explorando sus aplicaciones prácticas en diversos contextos.

2 Instrucciones para la selección del corpus

Este taller debe ser desarrollado en grupos de 2 personas.

Para iniciar el desarrollo de este taller, cada grupo deberá disponer de un conjunto de documentos que conformarán su corpus de análisis. La selección de estos documentos es flexible y queda a criterio de cada grupo. Pueden emplear bases de datos de tareas de Procesamiento de Lenguaje Natural, o pueden construir ustedes mismo el corpus a partir de los documentos de su elección.

Tengan en cuenta varios factores a la hora de elegir los documentos:

- Que estén en el idioma en el que ustedes van a trabajar.
- Que tengan una codificación adecuada (UTF-8, ANSI, etc.).
- Que estén en un formato de archivos que ustedes puedan abrir y procesar.

Nota: Tengan también en cuenta que los documentos en formato PDF exigen unos pasos adicionales de depuración y extracción de información.

Recuerden que pueden aprovechar las funciones y herramientas disponibles en los cuadernos de Google Colab compartidos durante el curso, las cuales facilitarán significativamente el trabajo en este taller.

Pueden emplear, no obstante, la herramienta computacional de su eleccción (pueden trabajar en python desde cuadernos de Colab, o pueden trabajar en Orange o en otros lenguajes de programación). Pueden mezclar e intercalar el uso de las herramientas. Se espera de este taller que ustedes estén en la capacidad de resolver los problemas propuestos, independientemente de los medios empleados.

3 Actividades

Una vez hayan seleccionado el corpus, realicen las siguientes actividades:

1. Normalicen el formato de sus textos: i) pasen todos los textos a letras minúsculas; ii) eliminen o reemplacen caracteres especiales, símbolos y emojis.
2. Segmenten sus documentos a nivel de frase y palabra. Guarden el resultado en archivos separados.
3. Eliminen signos de puntuación y caracteres no alfabéticos. Calculen el número de tokens y palabras distintas en sus textos.
4. Seleccionen un listado de posibles *stop-words* y eliminen esas palabras de sus textos. Recalculen el número de tokens y el vocabulario. Comparen estos números con los obtenidos en el paso anterior. Pueden utilizar los listados de *stop-words* predefinidas que vienen con muchas librerías (p.ej. NLTK trae su propio listado de *stop-words*, Orange también tiene su propio listado).
5. Realicen tareas de *stemming* y lematizado en sus textos. De nuevo, recalculen el número de tokens y el vocabulario. Comparen estos números con los obtenidos en los pasos anteriores. Recuerden que pueden usar para las tareas de *stemming* y lematizado de los textos cualquiera de las herramientas que hemos visto y comentado en el curso, por ejemplo: Spacy, Stanza, NLTK, CORE NLP, etc.
6. Etiquetado gramatical (POS Tagging). Empleen una librería de su elección (por ejemplo, spaCy, Stanza, NLTK o CORE NLP) para etiquetar gramaticalmente cada token de sus textos. Guarden las etiquetas asignadas en un archivo de salida, de modo que cada token aparezca junto con su etiqueta POS correspondiente. Comenten y analicen brevemente los resultados, haciendo énfasis en posibles casos de ambigüedad o inconsistencias en la asignación de etiquetas.
7. Análisis de la distribución de etiquetas. Examinen las etiquetas gramaticales obtenidas en el punto anterior y analicen la distribución de las categorías más frecuentes (sustantivos, verbos, adjetivos, etc.). Presenten esta información en una tabla o gráfica. Discutan qué patrones se observan y cómo podrían relacionarse con la naturaleza del corpus.

8. Análisis sintáctico: generación de árboles de dependencia. Elijan un subconjunto representativo de oraciones de su corpus y realicen un análisis de dependencia sintáctica utilizando la misma herramienta (o alguna otra) empleada en el etiquetado POS (spaCy, Stanza, etc.). Guarden la representación de los árboles de dependencia (por ejemplo, en un formato textual o visual). Seleccionen dos o tres ejemplos llamativos para ilustrar la estructura sintáctica de oraciones complejas.

9. Evaluación del análisis sintáctico. Revisen manualmente los árboles de dependencia generados en el punto anterior y describan posibles errores o inconsistencias que puedan encontrar (por ejemplo, asignaciones erróneas de la raíz, dependencias ambiguas, etc.).

10. Identificación de entidades nombradas (NER). Implementen un módulo de reconocimiento de entidades nombradas con una librería adecuada (spaCy, Stanza, NLTK, etc.). Describan qué tipos de entidades se pueden identificar (por ejemplo, personas, organizaciones, ubicaciones, etc.). Comenten brevemente los resultados y dificultades encontradas, especialmente en el caso de entidades ambiguas o poco frecuentes.

11. Evaluación básica de NER. Seleccionen manualmente un subconjunto de sus documentos y comparen las entidades reconocidas por su sistema con las que realmente aparecen en el texto (gold standard o revisión manual). Calculen métricas como precisión, exhaustividad (recall) y F1.

12. Modifiquen todas las frases del archivo (en el que guardaron las frases segmentadas) para que tengan una marca de inicio de frase <s> y una marca de fin de frase </s> . Esto se hace con el fin de que se puedan identificar, en los modelos de lenguaje, las palabras más frecuentes con las que se inician frases y las más frecuentes al final de las mismas. A partir de estos textos calcule los *n-grams* (hasta n = 3).

13. Empleando el método de información mutua y los bigramas obtenidos en el paso anterior, determinen los pares de palabras que podrían ser *colocaciones* en sus textos.

14. Implementen un generador de frases aleatorias. Utilice los *n-grams* generados en el ítem 12 y sus modelos de lenguaje (sus probabilidades). Diseñen una estrategia que les parezca adecuada, por ejemplo, podrían determinar de forma aleatoria la palabra inicial de cada frase y completar desde ahí. Además, consideren diseñar estrategias de selección de *n-grams* para que el generador no caiga en bucles o ciclos cerrados.

15. Repitan el paso 12 y calculen los modelos de lenguaje a nivel de bigrama (probabilidades de los bigramas), pero esta vez dejen un documento por fuera del corpus de entrada. Utilicen este documento para evaluar la calidad de su modelo de lenguaje empleando la medida de perplejidad. Repitan varias veces este paso, pero cada vez con distintos documentos de evaluación. Comparen las medidas de perplejidad. Concluyan al respecto.
Nota: Tengan en cuenta que el texto de evaluación puede contener palabras OOV - *Out Of Vocabulary*; en ese caso convendría emplear estrategias de suavizado, p.ej: Laplace Smoothing o método de Backoff. Argumenten qué harían en este caso.

16. Implementen un módulo de corrección ortográfica para su sistema. Pueden utilizar sistemas ya diseñados como pyspellchecker (`https://pypi.org/project/pyspellchecker/`), hunspell (`https://pypi.org/project/hunspell/`), SymSpell (`https://symspellpy.readthedocs.io/en/latest/`) o adaptar la función de distancia de edición mínima. El sistema puede requerir interacción con el usuario si así lo desean, o diseñar un módulo de decisión sobre la sustitución (por ejemplo, sustituir siempre por la palabra más probable).

4 Importante

Se debe presentar un informe del taller, junto con los scripts que se usaron para generar los resultados que aparezcan en el informe. El informe debe incluir los resultados solicitados, con una discusión sobre los mismos.

El informe no tiene que incluir marco teórico. Pueden presentar el informe en distintos formatos:

- Formato IEEE.
- PDF exportado de cuadernos de trabajo (cuadernos de google colab, live scripts de matlab, notebooks de jupyter)
- Directamente el cuaderno de trabajo. Verifique además que entrega todo lo necesario (dependencias) para que los cuadernos puedan ser ejecutados.