Procesamiento de Lenguaje Natural o Minería de textos
Tema 1: Introducción al procesamiento de textos.

Objetivo: El participante identificará el procesamiento de textos a partir de los conceptos relacionados con el Procesamiento del Lenguaje Natural y la Minería de Textos, a través del lenguaje de programación Python, para el descubrimiento, extracción y almacenamiento de la información.

Temario:

1. ¿Qué es PLN?
2. Problemas de ambigüedad
3. Construcciones primitivas en texto
4. Funciones de cadenas y de comparación de textos
5. Manejo de archivos de texto
6. Internacionalización y problemas con caracteres no ASCII

Lecturas:

AMPLN. (2019). CICLing: International Conference on Computational Linguistics and Intelligent Text Processing. Obtenido de AMPLN Asociación Mexicana para el Procesamiento del Lenguaje: https://www.cicling.org/ampln/

Justicia de la T., M. d. (2017). Nuevas Técnicas de Minería de Textos: Aplicaciones. Universidad de Granada. Tesis Doctorales. Obtenido de http://hdl.handle.net/10481/46975

Gomez-Adorno, H., Bel-Enguix, G., Sierra, G., Sánchez, O., & Quezada, D. (2018). A machine learning approach for detecting aggressive tweets in spanish. In Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018), CEUR WS Proceeding.

Sidorov, G., Markov, I., Kolesnikova, O., & Chanona-Hernández, L. (2019). Human interaction with shopping assistant robot in natural language. Journal of Intelligent & Fuzzy Systems, 36(5), 4889-4899.

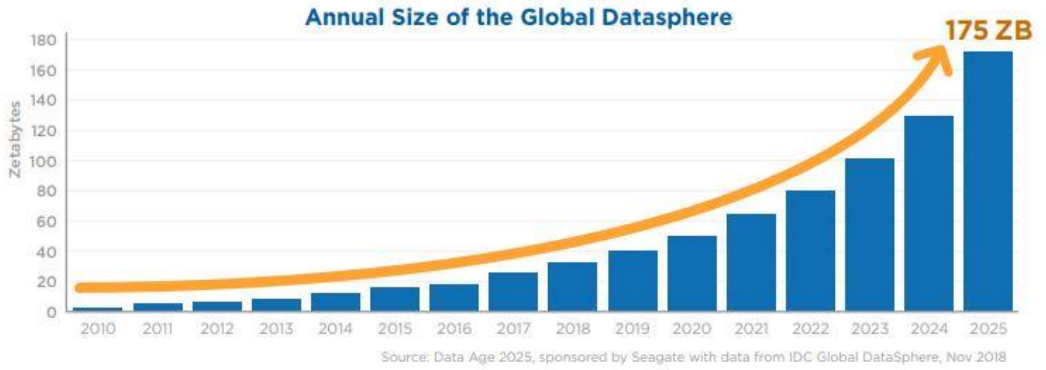
Introducción

Hoy en día, la propagación del uso de dispositivos computacionales y de comunicación para la producción de información digital, y en particular en la producción de documentos textuales estructurados, semi - estructurados y no estructurados ha generado la necesidad de desarrollar métodos, algoritmos y sistemas capaces de realizar el procesamiento automatizado de éstos para la recopilación, exploración y aprovechamiento de toda la información.

¡El texto está en todos lados!

Incremento de datos textuales

- Los datos textuales continúan creciendo exponencialmente
- En 2025 el volumen de datos en el mundo será 175 veces más que en 2011[1]
- El volumen de datos llegará a 175 zettabytes en 2025, según un informe de la consultora IDC[2][3], lo que significa el equivalente a 175 veces la información generada en 2011.



Fuente: The Digitization of the World From Edge to Core

- *Más de la mitad de los datos permanecerá guardado en la nube*[4]. Aproximadamente, el 80% de los datos de una organización se encuentra en formato no estructurado[5]: Llamadas de servicio al cliente, Correos electrónicos, blogs y redes sociales.

Datos escondidos a plena vista

[1] https://www.fundacionbankinter.org/blog/noticia/en-2025-el-volumen-de-datos-en-el-mundo-sera-175-veces-mas-que-en-2011

[2] International Data Corporation (**IDC**) Es la principal firma mundial de inteligencia de mercado, servicios de consultoría, y eventos para los mercados de Tecnologías de la Información, Telecomunicaciones y Tecnología de Consumo

[3] https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf

[4] https://www.fundacionbankinter.org/blog/noticia/en-2025-el-volumen-de-datos-en-el-mundo-sera-175-veces-mas-que-en-2011

[5] https://support.sas.com/resources/papers/proceedings14/1288-2014.pdf

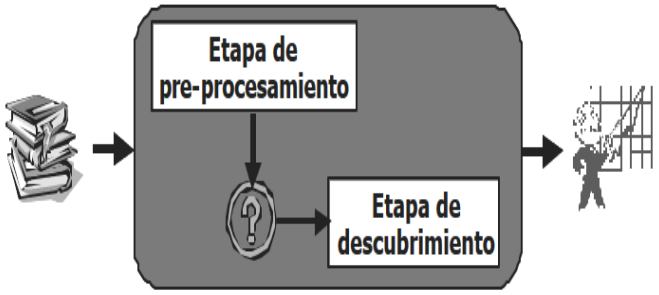
Visualización de video

- What is Natural Language https://youtu.be/Ian4sk4VcnA
- Tendencias IA: el procesamiento del lenguaje natural https://youtu.be/o3C2P-Wio5U
- **¿Qué es el Procesamiento de Lenguaje Natural y cómo aplicarlo? https://youtu.be/5c0qIh54uqE**
- La Siguiente Gran Revolución: NLP (Procesamiento del Lenguaje Natural) https://youtu.be/cTQiN9dewIg

El procesamiento del lenguaje natural (abreviado PLN, o NLP del inglés*, Natural Language Processing*) surge a partir de la interrelación que existe entre la lingüística, que es la ciencia que estudia el lenguaje humano, y la computación. A esta disciplina se le asignan diferentes nombres como: procesamiento de lenguaje natural, procesamiento de texto, tecnologías de lenguaje y lingüística computacional, en cualquiera de los casos se trata de procesar el texto por su sentido y no como un archivo binario. El PLN, es un área encargada del desarrollo eficiente de algoritmos para procesar textos y hacer la información accesible a las aplicaciones informáticas.

La **minería de textos** se define como el proceso de descubrimiento de patrones interesantes y nuevos conocimientos en una colección de textos, es decir, es el proceso encargado del descubrimiento de conocimientos que no existían explícitamente en ningún texto de la colección, pero que surgen de relacionar el contenido de varios de ellos (Hearst, 1999) (Kodratoff, 1999).

La metodología empleada para realizar la minería de textos puede ser general o específica. Una **metodología general** como la propuesta por (Tan, 1999), en el que define dos etapas principales: una **etapa de pre - procesamiento** y **una etapa de descubrimiento**. En la primera etapa, los textos se transforman a algún tipo de representación estructurada o semi - estructurada que facilite su posterior análisis, mientras que en la segunda etapa las representaciones intermedias se analizan con el objetivo de descubrir en ellas algunos patrones interesantes o nuevos conocimientos.



Proceso de minería de textos

Fuente: Montes y Gómez, M. (2001). Minería de texto: Un nuevo reto computacional. Obtenido de https://ccc.inaoep.mx/~mmontesg/publicaciones/2001/MineriaTexto-md01.pdf

¿Qué se puede hacer con datos textuales?

- Encontrar, identificar y extraer información relevante
- Clasificar documentos
- Búsqueda de documentos relevantes
- Análisis de sentimientos, Clasificación de Opiniones
- Agrupamiento de documentos
- Identificación de tópicos
- Resúmenes

Python para minería de textos

Python es un lenguaje de programación mutiparadigma, es decir, que soporta orientación a objetos, programación imperativa y programación funcional. Sirve para programar código y **desarrollar aplicaciones que permitan el análisis de datos**.

En los últimos años el lenguaje se ha hecho muy popular por razones como la cantidad de librerías que contiene, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar un realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero. También por la **sencillez y velocidad con la que se crean los programas**.

Instalador e instrucciones Python: http://python.org/

Python permite realizar tareas de pre - procesamiento de textos con la utilización de algunas de sus herramientas

- El Ecosistema SciPy
- NLTK (bibliotecas PLN más conocidas, potente y más utilizadas en el ecosistema de Python, útil para todo tipo de tareas, desde tokenización, hasta derivación, etiquetado de parte del habla y más)
- NumPy (extensión de Python que le agrega mayor soporte para vectores y matrices, constituyendo una biblioteca de funciones matemáticas de alto nivel para operar con esos vectores o matrices) y Pandas (biblioteca de software escrita como extensión de NumPy para manipulación y análisis de datos para el lenguaje de programación Python)
- Spacy (es un marco en el entorno de PLN moderno y confiable que rápidamente se convirtió en el estándar para hacer PLN con Python)
- Scikit-learn (biblioteca de aprendizaje automático, de software gratuito y código abierto, herramienta simple y eficiente para la minería de datos y el análisis de datos, proporciona utilidades para extraer las características del texto

Preparando el Entorno de trabajo

- **Python 3. *** https://www.python.org/downloads/
  (Introducción a Python: https://datacarpentry.org/python-ecology-lesson-es/01-short-introduction-to-Python/index.html**)**
- **Entorno** de trabajo interactivo: **Jupyter Notebooks** (permite desarrollar código en Python de manera dinámica e integrar en un mismo documento tanto bloques de código como texto, gráficas o imágenes) - https://jupyter.org/install. Documentación: https://jupyter-notebook.readthedocs.io/en/stable/. Aunque se pueden instalar otros IDE de desarrollo como: PyCharm, Visual Studio Code (en las que se debe de generar el archivo .py y ejecutarlo en consola)
- **Colaboratory**, también llamado **Colab** (https://colab.research.google.com/notebooks/intro.ipynb), te permite escribir y ejecutar código de Python en un navegador, con las siguientes particularidades: (https://www.datahack.es/blog/big-data/google-colab-para-data-science/)

- Sin configuración requerida
- Acceso gratuito a GPU
- Facilidad para compartir

Verificando que se encuentren instaladas las librerías: ***pip freeze*** – ***conda list***

Construcciones primitivas en texto

- Oraciones / cadenas de entrada
- Palabras o tokens
- Caracteres
- Documentos, archivos más grandes

Ejercicio1(es)-Introducción al procesamiento de textos.ipynb

Internacionalización:

La codificación de caracteres es el método que permite convertir un carácter de un lenguaje natural (como el de un alfabeto o silabario) en un símbolo de otro sistema de representación, como un número o una secuencia de pulsos electrónicos en un sistema electrónico aplicando normas o reglas de codificación.

Definen la forma en la que se codifica un carácter dado en un símbolo en otro sistema de representación. Ejemplos de esto son el código Morse, la norma ASCII o la UTF-8, entre otros.

Ejemplos:

- ASCII
- IBM EBCDIC
- Latin-1
- JIS: Estándar industrial Japonés
- CCCII: Código de caracteres chinos para el intercambio de información
- EUC: Código extendido de Unix
- Otros estándares nacionales
- Unicode y UTF-8

Unicode es el estándar internacional para caracteres de codificación y cadena Unicode es una estructura de datos en lenguaje Python utilizada para almacenar texto, mientras que los bytes se utilizan para almacenar datos binarios arbitrarios. En Python 2, cada cadena Unicode debe estar marcada con el prefijo "u", ya que utiliza caracteres ASCII de manera predeterminada, que no es tan flexible como la codificación Unicode, lo cual no tiene problemas con cadenas de textos escritas en inglés. En Python 2 tenías que especificar una codificación que soportara caracteres Unicode en tu script, como por ejemplo UTF-8 (y guardar el archivo con esa codificación), de caso contrario se producían errores al ejecutarlo.

```
# -*- coding: utf-8 -*-
```

Sin embargo, Python 3 almacena cadenas como Unicode por defecto que son más versátiles que las cadenas ASCII, lo que ahorra tiempo de desarrollo adicional, y se puede escribir y mostrar fácilmente muchos más caracteres directamente en el programa. Sin embargo, si desea que el código Python 3 sea compatible con Python 2, se puede mantener el "u" antes de la cadena.

Tarea:

A) Crear una función en Python que:
   1. Permita leer un archivo (por ejemplo: trabalenguas.txt y/o frases_famosas.txt)
   2. Extraiga del archivo cada texto en una sola línea.
      Debe de tener en cuenta lo siguiente:
      - En el archivo *trabalenguas.txt*, los trabalenguas están denotados entre comillas
      - En el archivo *frases_famosas.txt*, las frases famosas están denotadas entre guiones
   3. Guarde los textos extraídos en un nuevo archivo.

B) Indagar en las diferencias de implementación entre Python2 y Python3. Profundice con respecto a la codificación de caracteres.

Conclusiones

Debido al incremento de los datos en formato texto, toma vital importancia el PLN a partir del reconocimiento de patrones y de la interpretación de cadenas de textos para analizar de forma efectiva éstos grandes volúmenes de datos.

Cuando se enfrenta al texto con la idea de descubrir conocimiento, se encuentra con el problema de la falta de estructura de este. Esta falta de estructura es solo aparente, porque, realmente, el texto presenta una estructura demasiado compleja y difícil de tratar computacionalmente. Dependiendo del tipo de operaciones usadas en este pre – procesamiento de datos, será el tipo de patrones a descubrir en esta colección.