**Informe Entrega 1 Proyecto Procesamiento Lenguaje Natural**

Juan Camilo Acosta Rojas - 202324893

Samuel Goncalves Vergara - 202122595

Jefferson Alberto Hernández - 202120242

Juan Andrés Méndez – 201815808

Para esta primera etapa del proyecto, se realizó una investigación acerca de las posibles fuentes de información de las cuales podemos extraer contenido humorístico, dentro que las que se destacan las diferentes redes sociales como X o Reddit, dentro de las cuales se caracterizan por su posteo de contenido de humor en varios tipos de contenido (Imágenes, vídeos, gifs, entre otros). Es por ello, que podemos extraer contenido humorístico en distintas variaciones para lograr un perfecto modelado de este contenido en Modelos de Lenguaje.

**Proceso de recolección de datos:**

Dada las condiciones del enunciado del proyecto. Se realiza una investigación exhaustiva sobre las posibles fuentes de información de contenido humorístico las cuales nos puedan ser útiles para el propósito del proyecto. Dentro de las primeras fuentes, se obtuvo una fuente de datos las cuales presenta el juicio de personas humanas en un concurso sobre humor. En esta fuente de datos, se encuentra principalmente instancias de tweets seleccionados de cuentas enfocadas al humor en español. Por otro lado, se observa en cada uno de los jueces si para este fue gracioso o no el contenido aparte de un ranking entre 1 y 5 para cada registro del dataset [1]. Debido a las condiciones mencionadas inicialmente, se extraen 4000 de las mejores instancias rankeadas de esos registros para el dataset base con el que se trabajará.

Dentro de la segunda etapa de investigación, se comienza a indagar sobre las diferentes redes sociales existentes en las cuales, además de la variedad de contenido que presentan (imágenes, videos, entre otros), presentan una gran variedad de contenido humorístico. Dentro de las redes sociales a considerar se encuentran:

* X: esta red social, en cada uno de sus posts presenta diferente contenido multimedia del cual se puede extraer contenido humorístico en distintos idiomas. Para extraer información de su plataforma presenta un reto interesante, debido a que los altos costos para el consumo de su componente de extracción de información, se opta por usar otra alternativa, que en este caso es el Web Scrapping, el cual permite extraer información de sitios web. Una de las librerías que se emplean para este caso es twscrape, una librería de scrapping que perimite extraer información de varias cuentas de X por un intervalo determinado de tiempo (dadas las políticas de X, este bloquea la cuenta desde la cual se ingresa y se solicita la información). Para una extracción más eficiente, se opta por crear varias cuentas y realizar la extracción de información de forma paralela y asíncrona con dichas cuentas [2].

Una vez extraída la información, se realiza un preprocesamiento para eliminar caracteres que podrían afectar la integridad semántica del contenido, como lo es el caso de emojis, urls u algunos similares. Para este caso, se realiza un proceso de tokenización para clasificar cada URL como un solo token, nombres de usuario, entre otros escenarios específicos. (ver Dataset DatasetTwitterInternet.csv)

* Reddit: El proceso de extracción y etiquetado de contenido humorístico desde Reddit se realizó utilizando Python junto con varias bibliotecas especializadas. Empezamos por descargar imágenes de tres subreddits seleccionados por su contenido humorístico específico: **memexico**, **dankargentina**, y **yo\_cmt**. Para interactuar con la API de Reddit y acceder a estos subreddits, se utilizó la biblioteca **praw**, que permite acceder al contenido y sus metadatos de forma estructurada. Se desarrolló una función para descargar imágenes de los posts más populares de cada subreddit, filtrando solo aquellos posts que contenían imágenes en formatos como .jpg, .png, y .gif. Estas imágenes fueron descargadas y guardadas en una carpeta local, y se registraron en un archivo de log para evitar duplicados en futuras ejecuciones. (ver DatasetImagesTranslate.xlsx)
* Para la generación de descripciones de las imágenes, se empleó un modelo de **captioning** basado en la arquitectura **BLIP (Bootstrapped Language-Image Pre-training)**, que permite generar subtítulos automáticos en inglés para imágenes. Utilizamos el modelo blip-image-captioning-base de **Salesforce**, implementado con **transformers** y **torch** para procesar las imágenes y generar las descripciones en inglés. Posteriormente, estas descripciones fueron traducidas al español con la biblioteca **googletrans**, que facilita el uso de la API de Google Translate en Python [3]. El enfoque de captioning permitió automatizar el proceso de descripción de imágenes, proporcionando un análisis inicial de contenido visual que luego se tradujo al idioma del público objetivo. Esto no solo ahorró tiempo y esfuerzo en la revisión manual, sino que permitió obtener una interpretación en dos idiomas, haciendo que el contenido sea accesible y relevante para los usuarios.

Para la extracción de estos sitios, se espera obtener cerca de 4000 registros, dentro de los cuales 2000 se espera que sean captions o información extraída a partir de las imágenes obtenidas.

Una tercera fuente de información, en este caso, va a ser información generada por Large Language Models (LLM). Para generar este dataset humorístico, se utilizó ChatGPT-4o de OpenAI, un modelo de lenguaje grande (LLM) que facilita la creación de textos en lenguaje natural con diversidad temática y estilo. A través de instrucciones específicas, el modelo produjo frases y párrafos humorísticos sobre situaciones cotidianas como el trabajo, la vida familiar y las rutinas diarias. ChatGPT-4o, al estar entrenado en amplios volúmenes de datos, pudo generar contenido espontáneo y variado, adecuado para un dataset enfocado en el humor. El proceso fue iterativo: en cada solicitud, ChatGPT-4o generaba entre 80 y 100 datos, revisados posteriormente para asegurar que el tono y el humor fueran consistentes y variados. Se repitió el proceso anterior hasta completar los 2000 mil datos requeridos para completar, con las otras fuentes mencionadas anteriormente, cerca de 10000 registros.

**Etiquetado de información:**

Luego de haber reunido todos los datos en una sola fuente de información, se dispone un archivo compartido Excel para, guardar y mostrar cada una de las anotaciones realizadas en cada uno de los registros del dataset por cada uno de los 3 jueces. En el archivo de entrega de esta parte del proyecto se adjunta dicha evidencia de etiquetado. Adicionalmente, se evidencia el etiquetado respecto a la escala de gracia (de 1 a 5) de cada uno de los registros. Cabe destacar, que, pese a que se consideren algunos registros como contenido humorístico, puede que no tengan una escala alta promedio de gracia, lo cual podría presentar esta diversificación de, lo que puede ser gracioso, pero en una escala baja.

Como herramienta complementaria para lograr este proceso, usamos el conjunto de librerías denominado tkinter, el cual permite realizar interfaces de una manera rápida y sencilla. Con una plantilla generada a partir de este paquete, es posible importar y manipular el archivo Excel mencionado anteriormente, para lograr el etiquetado y poder utilizar este archivo entre varios usuarios.  
  
**Proceso de Etiquetado Usando Python y tkinter**

El objetivo de este proyecto fue realizar un proceso de etiquetado para un conjunto de datos que contiene contenido humorístico. Este etiquetado incluye:

1. La calificación de si cada entrada es "graciosa" o "no graciosa".
2. Una puntuación en una escala de gracia del 1 al 5.

El archivo compartido de Excel fue preparado para capturar la evaluación de tres jueces, quienes anotaron y calificaron cada entrada. Para simplificar y hacer eficiente este proceso de etiquetado colaborativo, utilizamos tkinter en Python para construir una interfaz gráfica que permitió que cada juez evaluara cada entrada de manera intuitiva y rápida.

**Herramientas Utilizadas**

1. **Python**: el lenguaje principal para el desarrollo de la aplicación.
2. **tkinter**: una biblioteca estándar en Python que permite crear interfaces gráficas de usuario (GUI) de forma sencilla y rápida.
3. **Pandas**: para manipular y leer los datos en formato CSV.
4. **Pillow**: para cargar y redimensionar imágenes en la GUI, cuando sea necesario.

**Descripción de la Aplicación de Etiquetado**

1. **Carga de Datos**: El programa utiliza un cuadro de diálogo de archivo para seleccionar el archivo CSV que contiene el conjunto de datos. Una vez cargado, se muestra la primera entrada al usuario para su evaluación.
2. **Interfaz de Evaluación**:
   1. La aplicación muestra el texto o la imagen de cada registro.
   2. Incluye opciones de evaluación mediante dos casillas de verificación: "Gracioso" y "No Gracioso".
   3. Incluye una escala deslizable (de 1 a 5) para calificar cuán graciosa es cada entrada. (Cuando el usuario marca el resultado como “No gracioso” no se guarda el resultado de la escala.
3. **Navegación**:
   1. Se puede avanzar al siguiente registro una vez que el usuario ha guardado la calificación actual.
   2. La aplicación cuenta con atajos de teclado para mejorar la eficiencia:
      1. f para marcar como gracioso,
      2. n para marcar como no gracioso,
      3. Teclas numéricas (1 a 5) para ajustar la escala de gracia.
4. **Guardado de Resultados**: Cada evaluación es guardada en un archivo CSV llamado results.csv, donde se almacena la información de cada registro, junto con la calificación proporcionada por el juez. Este archivo se combinó por cada uno de los Jueces para llegar a resultado final que se encuentra en ‘humor\_dataset.cvs’ y se añadio una columna adicional que es cuando los jueces llegaron a un concenso llamda es gracioso.

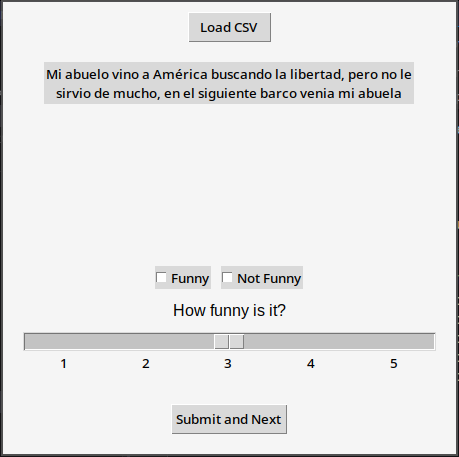


Figura 1. Screenshot de interfaz elaborada en tkinter para etquetado de datos

**Conclusiones:**

* El tener contenido considerado gracioso, pero en una escala baja de gracia, podría permitir diversificación de detección de contenido gracioso por parte del Modelo de Lenguaje a utilizar.
* Las redes sociales son una fuente útil de información para una variedad de casos, ya que presentan diferentes tipos de contenido multimedia los cuales se puede extraer para su análisis en un Modelo de Lenguaje.
* El Web Scrapping es una metodología importante de extracción de información, permite obtener contenido relevante de plataformas web para su procesamiento y análisis.
* Los sistemas de captioning permiten hacer inferencia a la información y el contenido que tienen cada uno de los registros multimedia, puede ser una gran herramienta para comprobar la eficiencia de modelos de lenguaje para extraer información de este tipo de contenido.

**Referencias:**

1. “HAHA @ IberLEF2021. Humor Analysis based on Human Annotation”. [Online] Available: <https://www.fing.edu.uy/inco/grupos/pln/haha/index.html>
2. “twscrape”. [Online]. Available: <https://github.com/vladkens/twscrape>
3. “googletrans 3.0.0” [Online] Available: <https://pypi.org/project/googletrans/>