Universidad Intercontinental de la Empresa

Asignatura

Sistemas Interactivos Inteligentes

Práctica 2

Unidad IV

Pre-CLIP y CLIP

Profesor: David Rivas Villar

Contenido

1 Objetivo de la actividad

En esta práctica, los estudiantes, trabajando individualmente deberán realizar 3 tareas:

1. Crear un dataset de 20 imágenes con descripciones, agrupadas en 4 categorías
2. Computar la similitud entre los embeddings de esas imágenes y sus descripciones, producidos por una red de imagen y por un modelo de texto
3. Computar la similitud entre los embeddings de esas imágenes y sus descripciones, producidos por el modelo CLIP

Con los resultados de estas tareas se deberá elaborar una memoria de la práctica.

Posteriormente se detalla cada una de estas tareas.

Se entrega un entorno mínimo en Docker que los alumnos podrán personalizar con las dependencias o bindings necesarios (por ejemplo, CUDA o MLX).

La entrega será en ese mismo formato y estará acompañada, además del código fuente y desarrollos pertinentes, del dataset de imágenes y descripciones, así como de una **memoria**. En esta los alumnos detallarán las decisiones de implementación y harán un análisis de los resultados de las tareas 2 y 3 en función al contenido de su dataset, recogido en la tarea 1.

2 Resultados de aprendizaje relacionados

RA02	Explicar los principios, beneficios y desafíos asociados al diseño de sistemas de interacción multimodal.
RA03	Analizar e implementar arquitecturas de software y hardware que permiten la integración de múltiples modalidades de interacción en base a los tipos de entradas y salidas requeridas para cada problema
RA08	Utilizar herramientas y bibliotecas de software especializadas en reconocimiento de emociones, análisis facial, síntesis de voz, etc.

3 Descripción de la actividad

Los alumnos deberán, crear un dataset y, a continuación, usarlo para realizar las tareas restantes de la actividad. Finalmente deberán dejar por escrito sus análisis y conclusiones en una memoria.

3.1 Parte 1: dataset

Los alumnos deberán crear un dataset de imágenes y captions (pie de foto, título, breve descripción) que deberán ser emparejadas, es decir, para cada imagen el alumno deberá elaborar un caption que sea adecuada.

Este dataset deberá constar de 20 imágenes con sus respectivas captions, que deberán estar divididas en 4 categorías. De esta forma, el alumno tendrá 4 grupos de 5 imágenes. Las categorías son a elección del alumno, según sus intereses. Algunos ejemplos podrían ser: “Escenas del hogar”, “Ambientes urbanos”, “Comidas del mundo”, “Animales salvajes” etc.

Las imágenes que compongan el dataset (así como sus captions, cuando aplique) deberán:

- Deben ser **apropiadas para un entorno educativo** (sin contenido violento, sexual, ofensivo, etc.). En caso de incluirse una imagen o caption manifiestamente inapropiada, implicará un 0 en la evaluación
- Tener una resolución y calidad mínima, es decir, ser nítidas, claras, etc.
- Cada caption debe estar directamente relacionada con la imagen correspondiente
- El texto de la caption, por simplicidad, deberá estar redactado en **inglés**
- No se aceptarán imágenes o captions generadas por IA

El formato de entrega del dataset será, preferiblemente, un archivo comprimido. Alternativamente, se puede incluir en un repositorio, tipo Git, cuyo link deberá ser indicado en la entrega (por ejemplo, en la memoria o en un archivo de texto extra). El contenido (tipo de archivos) del dataset puede ser escogido por el alumno pero, en todo caso, deberá ser fácilmente legible (por programas comunes).

3.2 Pre-CLIP

A modo de ejemplo de un mundo “pre-clip” los alumnos deberán comparar los embeddings de una red de imagen y otra de texto. Posteriormente, los alumnos deberán analizar, midiendo distancia o similitud, si los embeddings de las imágenes *matchean* con los de sus captions. Los alumnos deberán describir sus resultados, así como métricas o métodos en la memoria a entregar.

Para realizar este proceso se recomienda usar redes típicas como ResNet, no es necesario usar una red de “vanguardia” ni muy pesada. No obstante, si **es fundamental que la red usada esté pre-entrenada**. En este

sentido **no es necesario entrenar ni hacer fine-tunning de la red**. En concreto, típicamente, para el ejemplo de ResNet, la cabeza de clasificación deberá ser eliminada/*bypased* para poder acceder a las características de la red directamente.

De manera similar, se recomienda usar *sentence-transformers/distiluse-base-multilingual-cased* como modelo de texto, si usamos la ResNet como método de imagen, por ejemplo.

Cabe destacar que no hay requisito de usar estas redes. Son ejemplos que los alumnos pueden adoptar, se proporcionan para facilitar el proceso. Es importante destacar que **el uso de otras redes no implica mayor nota** y, en todo caso, deberá ser justificado en la memoria.

3.3 CLIP

Se realizará un proceso equivalente al anterior, pero usando la red multimodal pre-entrenada CLIP. En este sentido, tanto la imagen como su caption deberá ser procesada por CLIP. De manera análoga a lo visto anteriormente, los alumnos deberán comprobar el nivel de coincidencia entre cada imagen y su caption mediante distancia o similitud de los embeddings. Los resultados deberán ser detallados y explicados en la memoria y, a mayores, comparados con los de la tarea anterior explicando su correlación o no correlación.

Se recomienda usar la versión de CLIP incluida en la librería Transformers (incluida en el Docker) por su mayor simplicidad de uso. No obstante, la derivada del repositorio de CLIP original es perfectamente válida.

Finalmente, usar otro modelo similar a CLIP (como SigLIP) no implicará mayor nota y deberá ser justificado en la memoria de manera adecuada.

4 Datos proporcionados

Se proporciona a los alumnos una **configuración mínima** para facilitar la implementación de la práctica. Esta configuración se basa en un **Dockerfile** sencillo, que incluye una serie de librerías que pueden ser usadas para implementar la práctica. Los alumnos tienen libertad para **modificarlo y añadir componentes, *stages* u optimizaciones** según consideren necesario.

Además, se incluye un **Makefile** que simplifica la creación y uso de las imágenes de Docker.

5 Memoria

Como se ha mencionado previamente, la memoria deberá incluir un resumen del trabajo realizado justificando las decisiones de diseño e implementación, cuando sea necesario. Adicionalmente, como se ha comentado, los alumnos deberán describir su dataset (tarea 1) y hacer los análisis pertinentes para las tareas 2 y 3.

Esta memoria podrá contener el soporte audiovisual escogido por el alumno tales como fotos o incluso videos (mediante enlaces, por ejemplo). Deberá tener una longitud acotada, no pudiendo superar en ningún caso las 3000 palabras ni ser más corta de 500.

La memoria es parte imprescindible del trabajo, en caso de no entregarse o entregarse de manera deficiente, la práctica será suspensa.

6 Entrega y evaluación

Se habilitará un **repositorio de entrega** con fecha límite a las **23:59:00 del viernes 31 de octubre**. En dicho repositorio deberán subirse los archivos correspondientes o, en su defecto, un **enlace a un repositorio tipo Git** que contenga la totalidad del software y la memoria del proyecto.

Las **entregas fuera de plazo** y/o los **commits realizados después de la fecha límite** serán motivo de **suspenso automático** en la práctica.

Del mismo modo, se considerará motivo de suspenso el **no haber realizado la práctica**, incluyendo casos de **plagio, copia de repositorios ajenos o uso indebido de herramientas de IA**.

Como se ha indicado previamente, la **memoria** es un elemento **fundamental** de la práctica; por tanto, una entrega incompleta, plagiada o de calidad insuficiente será considerada **no apta**.

La evaluación tendrá en cuenta, además de la calidad del código y de la memoria, especialmente el análisis realizado de los resultados y so comparación.

El **formato de entrega** es libre, siempre que se cumplan las siguientes condiciones:

- Debe incluir **todo lo necesario para ejecutar el código**.
- Debe incluir la **memoria del proyecto**.
- Debe incorporar un **Dockerfile** que permita instalar las dependencias y ejecutar el código.
- Debe incluir una **receta de Makefile** (especificada en el archivo *README* o en la memoria) para construir la imagen de Docker y ejecutar.

6.1 Criterios de evaluación

Criterio	Ponderación	Descripción
Dataset	20%	El alumno ha creado un dataset adecuado con imágenes y captions precisas.
Código tarea 2	15%	El código de la tarea 2 es adecuado y funcional para el objetivo a cumplir
Código tarea 3	15%	El código de la tarea 3 es adecuado y funcional para el objetivo a cumplir
Memoria y justificación técnica	50%	Argumentación de decisiones (en caso de ser necesario), claridad expositiva, coherencia técnica y análisis adecuado y profundo de los resultados, adaptados a las temáticas presentadas en el dataset.