Universidad Intercontinental de la Empresa

Asignatura

Sistemas Interactivos Inteligentes

Práctica 3

Unidad V

VLMs

Profesor: David Rivas Villar

Contenido

1 Objetivo de la actividad

En esta práctica los estudiantes, trabajando individualmente, deberán usar VLMs del estado del arte para continuar con lo realizado en la anterior práctica.

Los VLMs representan la evolución natural de CLIP y fusionan lo aprendido en las unidades IV y V, aunando visión y texto. Aprovechando las capacidades de estos modelos, los alumnos, usando el dataset recopilado para la práctica 2, deberán evaluar las respuestas del modelo.

Para esta práctica no se entrega un entorno Docker debido a la libertad que tienen los alumnos para seleccionar modelos y/o métricas. Sin embargo, se dan una serie de consejos sobre librerías que se pueden usar.

La entrega será, igual que en las anteriores, en formato Docker 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 obtenidos.

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

Usando el dataset creado en la práctica 2, los alumnos deberán obtener captions usando VLMs y compararlas con las de su dataset. Posteriormente, los alumnos deberán discutir y analizar los resultados en la memoria.

3.1 VLMs

Para la realización de la práctica los alumnos deberán crear código que permita ejecutar VLMs de forma local (i.e. no online) a los que, posteriormente, les pasarán su dataset para ver qué resultados se obtienen.

Los alumnos deberán, al menos**, probar un VLM,** con pruebas en más de un modelo siendo valoradas positivamente para la nota. Los alumnos tienen la libertad para escoger el modelo que deseen, no obstante, están limitados por la capacidad de su portátil para ejecutarlos. En ese sentido, está permitido usar modelos cuantizados.

Adicionalmente, los alumnos deberán tratar de extraer el máximo rendimiento del modelo que usen para el tipo de imágenes o captions que compongan su dataset, como, por ejemplo, mediante prompt-tuning. No obstante, **esto no implica, en ningún caso entrenamiento.**

Para esta parte **se recomienda** usar la librería transformers, que contiene gran cantidad y variedad de modelos del estado del arte con diferente cantidad de parámetros y cuantizaciones. Para facilitar el uso de este tipo de librerías se sugiere a los alumnos usar como base imágenes de Docker hub, como las oficiales de hugging fase que contienen, probablemente, gran cantidad de las herramientas y/o liberarías que los alumnos pueden necesitar para las tareas de esta práctica.

3.2 Evaluación

Los alumnos deberán evaluar el rendimiento de las diferentes configuraciones probadas mediante métricas objetivas y valoraciones subjetivas.

En primer lugar, se deberán seleccionar y usar, al menos, **dos métricas diferentes**. Estas métricas deberán ser adecuadas para la tarea objetivo, *image captioning*. Los alumnos deberán realizar un análisis de resultados del modelo o modelos probados con las diferentes configuraciones probadas basándose en estas métricas.

Adicionalmente, los alumnos también pueden usar métricas cualitativas, basándose en su *impresión* de los resultados del VLM y teniendo también en cuenta la calidad de las captions generadas por ellos mismos en la práctica anterior.

Para esta tarea se recomienda usar la librería *evaluate*, que permite usar múltiples métricas adecuadas para este tipo de tareas.

3.3 Memoria

Como siempre los alumnos deberán realizar una memoria. En este caso, la memoria deberá contener información sobre por qué se ha seleccionado el modelo, pruebas o cambios realizados con el y resultados obtenidos con el mismo en las diferentes pruebas, evaluados con al menos 2 métricas.

Es importante recalcar que cualquier trabajo hecho, pero no reflejado de manera adecuada en la memoria con el análisis debido, **no será tenido en cuenta**.

Los alumnos deberán responder, **como mínimo y de manera adecuada**, a las siguientes preguntas durante la redacción de la memoria:

- ¿Qué modelo VLM se ha escogido y por qué?
- ¿El modelo escogido responde como es esperado al prompt y a las imágenes?
- ¿Qué métricas se usaron para la evaluación cuantitativa y por qué?
- ¿Qué clases o casos funcionaron mejor y cuáles peor?
- ¿Son las métricas escogidas representativas del rendimiento del modelo?
- ¿Qué limitaciones se han encontrado en el modelo o en el dataset?
- ¿El tiempo de inferencia afecta a la usabilidad del modelo en un caso real? ¿Podría usarse en tiempo real? ¿Se requeriría hardware especializado?

La memoria tendrá una longitud acotada, no pudiendo superar en ningún caso las 3000 palabras ni ser más corta de 500. Esta memoria podrá contener el soporte audiovisual escogido por el alumno tales como fotos o incluso vídeos (mediante enlaces, por ejemplo). La memoria es parte imprescindible del trabajo, en caso de no entregarse o entregarse de manera deficiente, la práctica será suspensa.

4 Entrega y evaluación

Se habilitará un **repositorio de entrega** con fecha límite a las **23:59:00 del viernes 14 de noviembre**. 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.

4.1 Criterios de evaluación

Criterio	Ponderación	Descripción
Código e implementación	25%	El alumno ha creado código adecuado para la ejecución de las pruebas necesarias.
Memoria y justificación técnica	75%	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.