Reporte Técnico: Destilación de Conocimiento

Implementación de BERT-Tiny asistido por DistilBERT en Hardware de Nueva Generación

Equipo de Desarrollo de IA

25 de enero de 2026

Resumen

Este documento detalla el proceso de preparación, entrenamiento y validación de un modelo de lenguaje mediante la técnica de *Knowledge Distillation*. El proyecto abordó desafíos técnicos significativos relacionados con la compatibilidad de hardware de última generación (NVIDIA RTX 5060 Ti) y la optimización de entornos de ejecución en Windows 11.

1 Fase 1: Preparación del Sistema

Debido a la incompatibilidad inicial de Python 3.13 con las librerías de *Deep Learning* actuales, se realizó un *downgrade* controlado a Python 3.11.9.

1.1 Configuración del Entorno

Se procedió a la creación de una estructura de proyecto aislada mediante entornos virtuales de Python para garantizar la reproducibilidad.

```
# Creacion del proyecto
mkdir PracticaDestilacion
cd PracticaDestilacion

# Entorno virtual
python -m venv venv
.\venv\Scripts\activate

# Instalacion de dependencias (Nightly para soporte serie 50)
pip3 install --pre torch --index-url https://download.pytorch.org/whl/nightly/cu124
pip install transformers datasets scikit-learn accelerate
```

Listing 1: Comandos de preparación en PowerShell

2 Fase 2: Script de Entrenamiento (destilacion_gpu.py)

Dada la arquitectura **sm_120** de la RTX 5060 Ti (Serie 50), se optó por una estrategia de ejecución en **CPU** para asegurar la estabilidad del proceso, evitando errores de falta de imágenes de kernel en PyTorch.

2.1 Lógica de Destilación

Se implementó un *Trainer* personalizado para calcular la pérdida de destilación utilizando la Divergencia de Kullback-Leibler (KL Divergence), balanceando la pérdida del estudiante con las predicciones del profesor.

Configuración de Modelos

Profesor: `distilbert-base-uncased-finetuned-sst-2-english`
Estudiante: `prajjwal1/bert-tiny`

3 Fase 3: Validación y Comparativa

Tras un entrenamiento de aproximadamente 2 horas en CPU (12,630 pasos), se ejecutó un script de validación para comparar el rendimiento real del modelo destilado frente al profesor.

3.1 Resultados Métricos

- **Precisión Final (Accuracy):** 82.45%
- **Parámetros del Profesor:** 67M
- **Parámetros del Estudiante:** 4.4M
- **Factor de Reducción:** 15.3x más pequeño.

4 Registro de Errores y Soluciones

A continuación se detallan los conflictos técnicos encontrados durante la práctica y las soluciones aplicadas:

#	Error Detectado	Solución Aplicada
1	`TypeError` en *evaluation_strategy*	Actualización de sintaxis a `eval_strategy` para Transformers v4.4x.
2	Incompatibilidad CUDA **sm_120**	Desactivación forzada de GPU mediante `CUDA_VISIBLE_DEVICES=""`.
3	Conflicto *token_type_ids*	Filtrado manual de entradas para el modelo DistilBERT dentro del *compute_loss*.
4	Error de carga de tensores mixtos	Forzado explícito de `device="cpu"` en todos los componentes del modelo.

Cuadro 1: Bitácora de Troubleshooting.

5 Conclusiones

La práctica demuestra que es posible obtener un modelo altamente eficiente (BERT-Tiny) con una pérdida de precisión aceptable respecto a un modelo mayor. A pesar de las limitaciones temporales del hardware Serie 50, la ejecución en CPU permitió validar la arquitectura de destilación de forma exitosa.