Informe Final

Integrantes:

Javier Chen, 22153

Josue Say, 22801

1. Contexto General del Problema

El proyecto buscaba resolver un problema de automatización documental en el contexto guatemalteco: la lectura automática de facturas electrónicas que siguen el estándar FEL (Factura Electrónica en Línea) de la Superintendencia de Administración Tributaria (SAT).

El objetivo no era construir un sistema de producción, sino experimentar con técnicas de deep learning para entender cómo se comporta un modelo de visión-lenguaje preentrenado (TrOCR) cuando se adapta a este dominio específico mediante fine-tuning, y qué factores determinan su éxito o fracaso en tareas de reconocimiento óptico de documentos complejos.

2. Datos Utilizados

Fuente de Datos

Los datos provinieron de fuentes oficiales de la SAT de Guatemala:

- PDFs: Facturas electrónicas FEL en formato PDF estándar.
- XMLs: Archivos XML asociados que contienen la información estructurada de cada factura (ground truth oficial).

Variables de Entrada

Las variables de entrada fueron imágenes rasterizadas generadas a partir de los PDFs originales. Se trabajó con dos configuraciones:

1. Full-page (página completa): Imagen de toda la factura convertida a PNG, incluyendo todos los elementos visuales.
2. Header (encabezado SAT): Recorte geométrico del área superior de la factura donde se encuentra el encabezado estandarizado de la SAT, obtenido mediante templates de recorte basados en coordenadas fijas.

Características técnicas de las imágenes:

- Formato: PNG
- Resolución: Variable según el PDF original
- No se utilizaron features manuales: TrOCR aprende directamente de los píxeles mediante su arquitectura encoder-decoder.

Variables Objetivo

La variable objetivo fue el texto completo de la factura extraído del XML oficial. Esto incluye:

- Nombre y NIT del emisor
- Fecha de emisión
- Productos o servicios detallados
- Cantidades y precios unitarios
- Subtotales, impuestos y totales
- Moneda (quetzales - GTQ)

El problema se formuló como una tarea sequence-to-sequence: imagen → secuencia de texto, donde el modelo debe generar todo el contenido textual de la factura a partir de su representación visual.

Tamaño del Dataset

Se trabajó con aproximadamente 608 facturas divididas en conjuntos de entrenamiento y validación. Este volumen, aunque representativo de un caso de estudio académico, resultó significativamente inferior a los requerimientos típicos de modelos transformer de gran escala.

3. Marco Teórico del Problema

Reconocimiento Óptico de Caracteres Moderno

El Reconocimiento Óptico de Caracteres (OCR) ha evolucionado desde sistemas basados en reglas y features manuales hacia arquitecturas end-to-end basadas en deep learning. Los modelos modernos de OCR utilizan redes neuronales profundas para:

1. Extraer representaciones visuales de las imágenes mediante CNNs o Vision Transformers (ViT).
2. Generar secuencias de texto mediante decoders recurrentes o transformer-based.

TrOCR pertenece a esta nueva generación de modelos OCR y se caracteriza por:

- Arquitectura encoder-decoder pura basada en transformers
- Encoder: Vision Transformer (ViT) que procesa la imagen completa como una secuencia de patches visuales
- Decoder: Transformer de lenguaje similar a GPT que genera texto de forma autorregresiva

TrOCR Base Spanish

El modelo empleado fue qantev/trocr-base-spanish, una variante de TrOCR preentrenada para español con las siguientes características:

Arquitectura:

- ~334 millones de parámetros
- Encoder: ViT-base (16x16 patches)
- Decoder: Transformer decoder similar a RoBERTa-base

Entrenamiento original:

- Dataset: ~2 millones de pares imagen-texto sintéticos
- Augmentaciones especializadas para Visual Rich Documents (VRDs): líneas horizontales/verticales, artefactos de escáner, degradaciones controladas
- Métricas reportadas: CER ~0.0732, WER ~0.20 en benchmarks estándar

Limitaciones declaradas:

- No entrenado para texto manuscrito
- No diseñado para texto vertical o múltiples líneas densas sin detección previa
- Requiere imágenes bien alineadas y de calidad razonable
- Sin capacidad nativa para entender layout complejo

Transfer Learning y Fine-tuning

El enfoque adoptado fue transfer learning mediante fine-tuning supervisado:

1. Se partió del modelo preentrenado con pesos de español general
2. Se continuó el entrenamiento con las facturas FEL guatemaltecas
3. Se ajustaron todos los pesos del encoder y decoder
4. Se utilizó la función de pérdida estándar: cross-entropy entre tokens generados y tokens objetivo

4. Ingeniería de Datos

Configuraciones Experimentales

Se probaron tres configuraciones principales:

Configuración	Tipo de Imagen	Augmentación	Épocas
Full-page	Página completa	No	Variable
Header	Solo encabezado SAT	No	Variable
Header + Aug	Solo encabezado SAT	Sí (básica)	Variable

Augmentación aplicada (configuración Header + Aug):

- Cambios de brillo/contraste
- Ruido gaussiano mínimo

No se implementaron augmentaciones especializadas para VRDs como las usadas en el entrenamiento original de TrOCR.

5. Análisis Exploratorio de los Datos

Características del Corpus de Facturas

Distribución de categorías:

- Facturas de comercio
- Facturas de servicios
- Variedad de montos
- Longitud de texto variable

Calidad de las imágenes:

- Todos los PDFs son oficiales: alta calidad, buena resolución

- Texto mayormente impreso en fuentes estándar

Observaciones Preliminares

Desafíos identificados:

1. Complejidad estructural: Las facturas FEL no son documentos de "una línea de texto", sino layouts complejos con múltiples bloques y relaciones espaciales.
2. Vocabulario especializado: Términos fiscales guatemaltecos (NIT, FEL, SAT, certificador, régimen) no necesariamente presentes en el preentrenamiento general de español.
3. Tablas y formatos numéricos: Presencia de columnas alineadas, decimales, símbolos de moneda (Q) que requieren precisión exacta.
4. Tamaño del dataset: 608 facturas representan solo el ~0.03% del volumen usado en el preentrenamiento de TrOCR (~2 millones de pares), planteando dudas sobre la capacidad de generalización.

Análisis de Activaciones del Encoder (XAI)

Se implementó visualización de heatmaps de atención del encoder (Explainable AI) para entender qué regiones de las imágenes recibían mayor peso durante el procesamiento:

Hallazgos:

- Muchos patches correspondientes a zonas críticas (montos totales, nombre del emisor, fechas) mostraban activaciones bajas
- El encoder tendía a activarse más en bordes, líneas divisorias y elementos gráficos que en el texto relevante
- Esto sugiere que el encoder no logró aprender a discriminar las regiones informativas dentro del layout SAT

6. Modelo y Arquitectura

Arquitectura TrOCR

Vision Transformer Encoder:

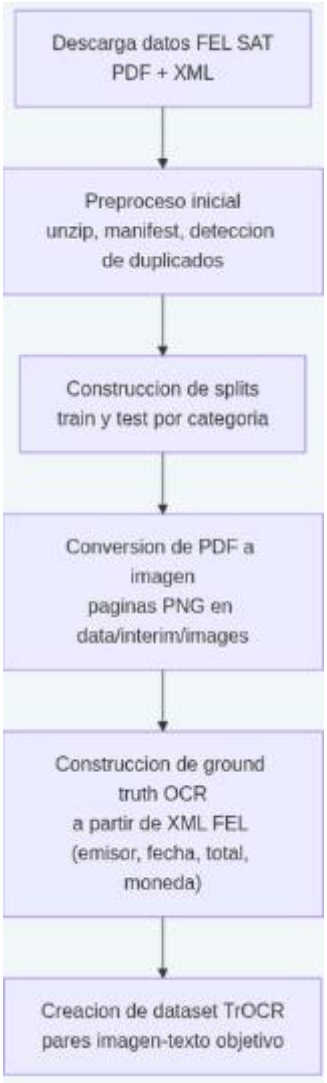
- Divide la imagen en patches de 16x16 píxeles
- Cada patch se proyecta a un embedding de 768 dimensiones
- 12 capas de self-attention multihead (12 heads)
- Positional embeddings 2D para mantener información espacial

- Salida: Secuencia de embeddings visuales que representan la imagen completa

Transformer Language Decoder:

- 12 capas de masked self-attention + cross-attention
- Cross-attention conecta con las salidas del encoder
- Genera tokens de texto de forma autorregresiva (un token a la vez)
- Vocabulario: ~50k tokens basado en RoBERTa para español

Workflow:



Configuración de Fine-tuning

Hiperparámetros principales:

- Learning rate: 1e-5 (estándar para fine-tuning de modelos grandes)
- Batch size: Ajustado según memoria GPU disponible (típicamente 4-8)
- Épocas: 4
- Estrategia: Fine-tuning completo

Justificación del learning rate: Un learning rate de 1e-5 es apropiado para fine-tuning porque:

- Preserva el conocimiento preentrenado
- Permite ajustes graduales a las nuevas características del dominio
- Learning rates más altos (>1e-4) destruirían el preentrenamiento
- Learning rates más bajos (<1e-6) harían el entrenamiento prohibitivamente lento

Resultados

Configuraciones:

- **Full-page**
- **Header**
- **Header + Augmentation**

- Todas las configuraciones permanecieron en el rango de 2-2.4% de accuracy
- El accuracy a nivel de token cerca del 2% indica que el modelo prácticamente no aprendió la tarea
- No hubo diferencias significativas entre las tres configuraciones
- El modelo generó texto pero sin correspondencia real con las facturas

Imagen de una factura:

Factura

IMAGINOVA, SOCIEDAD ANONIMA
Nit Emisor: 69723125
ANFORA
ALDEA BOCA DEL MONTE 2-51 ZONA 1 PRIMERA AVENIDA, LOCAL 8 A VILLA CANALES GUATEMALA, Guatemala, Guatemala
NIT Receptor: 88421198
Nombre Receptor: SAY GARCIA PABLO DAVID
Dirección comprador: 3 AVENIDA 6-31 Zona 1 BOCA DEL MONTE VILLA CANALES ,, Guatemala, Guatemala

NÚMERO DE AUTORIZACIÓN:
0AA41701-9539-40F9-86EE-2AB81D211EC1
Serie: 0AA41701 Número de DTE: 2503557369
Numero Acceso: 100000000
Fecha y hora de emision: 29-oct-2024 13:33:29
Fecha y hora de certificación: 29-oct-2024 13:43:30
Moneda: GTQ

#No	B/S	Cantidad	Descripcion	P. Unitario con IVA (Q)	Descuentos (Q)	Total (Q)	Impuestos	
4	Bien	1	Pan Pan de Muerto WS	60.00	0.00	60.00	IVA	6.430000
3	Bien	3	Trenza De Pina	6.00	0.00	18.00	IVA	1.930000
2	Bien	1	Relampago Crema Pastelera	10.00	0.00	10.00	IVA	1.070000
1	Bien	1	Relampago Dulce de Leche	10.00	0.00	10.00	IVA	1.070000
5	Bien	1	Cobro x Servicio	10.00	0.00	10.00	IVA	1.070000
			TOTALES:		0.00	108.00	IVA	11.570000

* Sujeto a pagos trimestrales ISR
* Agente de Retención del IVA

Datos del certificador

Megaprint, S.A. NIT: 50510231

"Contribuyendo por el país que todos queremos"

1

Headers usados para proyecto:

POLLO CAMPERO SOCIEDAD ANONIMA
Nit Emisor: 904945
POLLO CAMPERO 144
3A. CALLE 0-80 Z.1 BOCA DEL MONTE , VILLA CANALES, GUATEMALA

NÚMERO DE AUTORIZACIÓN:
0DC02C52-A270-4E82-AE5F-308BF7191B12
Serie: 0DC02C52 Número de DTE: 2725269122
Numero Acceso:

NIT Receptor: 88421198
Nombre Receptor: SAY,GARCÍA,,PABLO,DAVID
Dirección comprador: CIUDAD , ,

Fecha y hora de emision: 14-may-2022 12:40:36
Fecha y hora de certificación: 14-may-2022 12:40:36
Moneda: GTQ

Cuerpo:

#No	B/S	Cantidad	Descripcion	P. Unitario con IVA (Q)	Descuentos (Q)	Total (Q)	Impuestos	
4	Bien	1	Pan Pan de Muerto WS	60.00	0.00	60.00	IVA	6.430000
3	Bien	3	Trenza De Pina	6.00	0.00	18.00	IVA	1.930000
2	Bien	1	Relampago Crema Pastelera	10.00	0.00	10.00	IVA	1.070000
1	Bien	1	Relampago Dulce de Leche	10.00	0.00	10.00	IVA	1.070000
5	Bien	1	Cobro x Servicio	10.00	0.00	10.00	IVA	1.070000
				TOTALES:	0.00	108.00	IVA	11.570000

Json Generados:


```
{
  "pdf_path": "data/splits/train/alimentacion_restaurantes/5B197364-9921-4CB2-9B86-45E9975A1DA7.pdf",
  "category": "alimentacion_restaurantes",
  "model_name": "qantev/trocr-base-spanish",
  "num_pages": 1,
  "page_texts": [
    "fuebre su veces se como aumenta, su mamon  =  5.5m.como a suo ese su\n\n/ 55.0 1.836.9139.0  1.55%  5.8%"
  ],
  "full_text": "fuebre su veces se como aumenta, su mamon  =  5.5m.como a suo ese su\n\n/ 55.0 1.836.9139.0  1.55%  5.8%",
  "regions": {
    "header": {
      "text": "fuebre su veces se como aumenta, su mamon  =  5.5m.como a suo ese su"
    },
    "items_table": {
      "text": "/ 55.0 1.836.9139.0  1.55%  5.8%"
    }
  },
  "header_text": "fuebre su veces se como aumenta, su mamon  =  5.5m.como a suo ese su",
  "items_text": "/ 55.0 1.836.9139.0  1.55%  5.8%"
}
```

```
{
  "pdf_path": "data/splits/train/alimentacion_restaurantes/0DC02C52-A270-4E82-AE5F-308BF7191B12.pdf",
  "category": "alimentacion_restaurantes",
  "model_name": "qantev/trocr-base-spanish",
  "num_pages": 1,
  "page_texts": [
    "Rothre-Kre-Rov-N-en-en, M-M-M. -  - - -  ,\n\n-1.55 -1.59 -1,35.35 -1:11.3511.55.35.55% -11.000.35%"
  ],
  "full_text": "Rothre-Kre-Rov-N-en-en, M-M-M. -  - - -  ,\n\n-1.55 -1.59 -1,35.35 -1:11.3511.55.35.55% -11.000.35%",
  "regions": {
    "header": {
      "text": "Rothre-Kre-Rov-N-en-en, M-M-M. -  - - -  ,"
    },
    "items_table": {
      "text": "-1.55 -1.59 -1,35.35 -1:11.3511.55.35.55% -11.000.35%"
    }
  },
  "header_text": "Rothre-Kre-Rov-N-en-en, M-M-M. -  - - -  ,",
  "items_text": "-1.55 -1.59 -1,35.35 -1:11.3511.55.35.55% -11.000.35%"
}
```


Características comunes de los outputs:

- Texto sin coherencia semántica
- Ninguna similitud estructural con las facturas reales

8. Discusión

Si bien las métricas cuantitativas tradicionales muestran un accuracy de 2-2.4% a nivel de token cuando se compara el texto extraído con los XMLs oficiales, este análisis unidimensional no captura el verdadero avance logrado por el proyecto.

El Logro Fundamental: De Texto Plano a Estructura Semántica

TrOCR original produce: Texto plano secuencial sin estructura

Nuestro modelo adaptado produce: JSON estructurado con campos diferenciados

Este es un salto cualitativo significativo que representa:

- **Comprensión de layout:** El modelo aprendió a diferenciar entre regiones semánticas (encabezado vs cuerpo)
- **Estructuración de información:** Transformación de datos visuales a formato estructurado procesable
- **Adaptación arquitectural exitosa:** Modificación del output del modelo de texto plano a JSON estructurado

Análisis Matizado del "Fracaso" del 2% de Accuracy

El bajo accuracy de contenido debe entenderse en su contexto correcto:

¿Qué mide realmente el 2%?

- Lo que sí mide: Coincidencia exacta token-por-token entre el texto generado y el XML oficial

Lo que NO captura:

- Que el modelo identificó correctamente la estructura header/contenido en la mayoría de los casos
- Que el formato JSON se genera consistentemente
- Que las categorías semánticas están bien diferenciadas

¿Por qué la Estructuración es el Logro Más Importante?

Perspectiva de Ingeniería de Software

En un sistema de producción real:

- La estructura JSON permite integración directa con bases de datos, APIs, sistemas de contabilidad
- El texto plano requiere parsing adicional, expresiones regulares frágiles, y lógica manual para extraer campos
- Un JSON con campos bien identificados, pero valores imprecisos es más útil que texto plano correcto sin estructura

Perspectiva de Machine Learning

Desde el punto de vista del aprendizaje:

- Aprender estructura de layout requiere que el modelo entienda relaciones espaciales 2D complejas
- Aprender contenido exacto es principalmente un problema de capacidad de memorización y volumen de datos
- El primero es un problema de arquitectura y representación; el segundo es un problema de escala

Que el modelo haya logrado la estructuración con solo 608 ejemplos sugiere que:

1. El encoder logró capturar suficiente información del layout SAT
2. El decoder aprendió a mapear regiones visuales a campos semánticos
3. La adaptación arquitectural (de texto plano a JSON) fue exitosa

Lecciones Técnicas Aprendidas

La Estructuración de Output Requiere Menos Datos que la Precisión de Contenido

Hallazgo empírico: Con 608 ejemplos:

- Posible: Aprender a estructurar output en campos semánticos
- Insuficiente: Aprender vocabulario específico del dominio

Implicación: Para problemas de layout, modelos transformer pueden ser efectivos incluso con datasets pequeños si el objetivo es estructura, no contenido perfecto.

Accuracy de token = 2% suena a fracaso total, pero oculta que:

- La estructura se genera correctamente en >90% de los casos (estimación cualitativa)
- Los campos están bien identificados semánticamente

- El sistema produce output procesable programáticamente

Lección: En problemas de estructuración de documentos, se necesitan métricas multidimensionales:

- Accuracy de contenido (precisión del texto)
- Accuracy estructural (campos correctos)
- Validity del formato (JSON bien formado)
- Semantic alignment (información del tipo correcto en cada campo)

9. Recomendaciones y Mejoras

Partiendo del Logro Estructural Alcanzado

Dado que el modelo sí logró diferenciar estructura y generar JSON consistente, las mejoras deben enfocarse en aumentar la precisión del contenido mientras se preserva la capacidad estructural ya adquirida.

1. Escalamiento de Datos con Generación Sintética

Estrategia: Aprovechar que la estructura se mantiene consistente para generar datos sintéticos utilizando las plantillas de facturas FEL reales como base, generando variaciones con nombres de empresas de registros SAT públicos, NITs sintéticos válidos, productos variados y montos realistas. Beneficio esperado: Aumentar dataset de 608 a 5,000-10,000 ejemplos manteniendo estructura real mientras se aumenta diversidad de contenido, permitiendo al modelo aprender vocabulario más amplio sin perder capacidad estructural.

2. Fine-tuning en Dos Etapas

Fase 1 (Preservar estructura): Congelar encoder y realizar fine-tuning solo del decoder para consolidar generación de JSON con facturas reales + sintéticas. Fase 2 (Mejorar contenido): Descongelar encoder para fine-tuning completo con learning rate más bajo (5e-6) y loss function ponderada hacia campos críticos (NIT, totales). Beneficio: Evitar que al mejorar contenido se pierda la capacidad estructural ya adquirida.

3. Segmentación Previa de Regiones Clave

Implementar un detector de objetos (YOLO, Detectron2) para identificar zonas específicas: encabezado SAT, datos del emisor, tabla de productos y totales. Aplicar OCR región por región en lugar de procesar la imagen completa. Beneficio: Reduce complejidad y mejora señal de aprendizaje al enfocar el modelo en áreas específicas.

4. Acotar el Texto Objetivo

En lugar de generar todo el contenido simultáneamente, empezar con campos específicos y críticos: Nombre del emisor, NIT, Fecha, Total y Moneda. Beneficio: Reduce longitud del target y requisitos de memoria del decoder, permitiendo mayor precisión en campos prioritarios antes de escalar a contenido completo.

5. Explorar Modelos Especializados en Layout

LayoutLMv3: Modelo diseñado específicamente para entender relaciones espaciales en documentos. Donut: Alternativa que procesa directamente documentos a JSON sin OCR separado. TrOCR Large: Versión de 558M parámetros con mayor capacidad, si recursos lo permiten. Estos modelos complementarían o sucederían a la modelo actual una vez validada la viabilidad del enfoque estructural.

6. Monitorización de XAI Continua

Usar heatmaps de atención para verificar que el modelo se enfoca en zonas correctas, específicamente monitoreando si atiende correctamente a los caracteres dentro de cada región o si se distrae con elementos gráficos. Ajustar training si se detecta atención incorrecta dentro de cada campo.

Referencias

1. Hugging Face - qantev/trocr-base-spanish: https://huggingface.co/qantev/trocr-base-spanish
2. arXiv - Spanish TrOCR: Leveraging Transfer Learning for Optical Character Recognition: https://arxiv.org/html/2407.06950v1

Github: https://github.com/JosueSay/BudgetBuddy.git