Invoice Data Extraction Using Machine Learning: A Report

This report details the process of developing and deploying a machine learning model for invoice data extraction. The model aims to automate the task of identifying key information from invoices, such as sender, receiver, VAT number, and amount.

1. Data Acquisition and Preprocessing:

- Data Source: Invoices in PDF format stored in a dedicated folder.
- Extraction:
    - Used pdfplumber to extract text content from the PDFs.
    - Implemented OCR (Optical Character Recognition) using pytesseract for scanned documents.
- Preprocessing:
    - Converted text to lowercase.
    - Removed numbers and special characters.
    - Tokenized the text.

2. Model Architecture and Training:

- Model Selection: BERT (Bidirectional Encoder Representations from Transformers) was chosen due to its strong performance in natural language processing tasks.
- Fine-Tuning:
    - Utilized the bert-base-uncased pre-trained model.
    - Created a custom dataset of annotated invoices with labelled key information.
    - Fine-tuned the model on this dataset using the transformers library and PyTorch.

3. Training Process:

- Dataset:
    - The dataset consisted of preprocessed text and annotations for each invoice.
    - Annotations included fields like sender, receiver, VAT number, and amount.
- Training Hyperparameters:
    - Batch size: 32
    - Epochs: 5
    - Optimizer: Adam
    - Learning rate: 1e-5
- Training Setup:
    - Trained on a GPU to accelerate the process.
    - Used a cross-entropy loss function to evaluate the model's performance.

4. Evaluation and Optimization:

- Metrics:
    - Accuracy: Measured the percentage of correctly extracted information.
    - Precision: Calculated the ratio of correctly predicted instances to all instances predicted.
    - Recall: Evaluated the ability to identify all relevant instances.
    - F1-score: Combined precision and recall.
- Optimization Techniques:
    - Model Conversion: Converted the trained model to ONNX format for efficient deployment.
    - Quantization: Optimized the ONNX model using ONNX Runtime to reduce its size and improve performance.

5. Deployment Steps:

- Client Environment:
    - Installed the necessary libraries: `onnxruntime`, `joblib`, etc.
- Model Loading and Execution:
    - Created a Python script to load the optimized ONNX model.
    - Used the `onnxruntime` library to run the model for inference.

- Provided input invoice text to the model and obtained the extracted information.

Code:

The code for this project can be found in the attached Jupyter notebook. The notebook contains the following sections:

1. Data Acquisition and Preprocessing: Extract text from PDF invoices using pdfplumber and pytesseract libraries.
2. Model Architecture: Utilize a pre-trained BERT model and fine-tune it on the annotated dataset.
3. Training Process: Train the model using the annotated dataset and evaluate its performance using metrics such as accuracy, precision, recall, and F1-score.
4. Optimization Techniques: Convert the trained model to ONNX format and optimize it using ONNX Runtime.
5. Deployment: Create a Python script to load the optimized ONNX model and run it for inference.

Future Improvements:

The following improvements can be made to enhance the model's accuracy, performance, and usability:

1. Larger Dataset: Training on a more extensive and diverse dataset could improve generalization.
2. Advanced NLP Techniques: Exploring more sophisticated NLP methods like Named Entity Recognition (NER) could enhance accuracy.
3. Improved OCR: Investigating more robust OCR solutions could improve extraction from scanned documents.
4. Web Application: Creating a web-based interface for easy user interaction.

6. Performance on Client Desktop:

- The deployed model achieved satisfactory performance on the client desktop.
- The model accurately extracted key information from invoices, significantly reducing manual effort.
- Optimization techniques resulted in faster inference times and reduced memory consumption.

7. Future Improvements:

- Larger Dataset: Training on a more extensive and diverse dataset could improve generalization.
- Advanced NLP Techniques: Exploring more sophisticated NLP methods like Named Entity Recognition (NER) could enhance accuracy.
- Improved OCR: Investigating more robust OCR solutions could improve extraction from scanned documents.
- Web Application: Creating a web-based interface for easy user interaction.

Conclusion:

This project successfully implemented a machine learning model for invoice data extraction. The model's deployment on the client desktop significantly automated the process, saving time and effort. Further improvements can be made to enhance its accuracy, performance, and usability.