Invoice IQ Project Report

Abstract

The Invoice IQ project aims to automatically recognize and extract relevant information from invoices. We developed a system utilizing **Machine learning techniques** to identify labeled data and retrieve data from it. This system makes use of labeled datasets, an **OCR pipeline**, and a **(Named Entity Recognition) NER Transformer Model trained** on annotated data. With a **95% accuracy rate**, the completed model demonstrated significant potential for accelerating invoicing processes.

Problem Definition and Project Goals

Problem Definition

Invoicing, an essential business process, often involves manual data entry, leading to potential errors and inefficiencies. Despite the challenge posed by the diverse invoice formats, automating the invoicing process can enhance accuracy and save time.

Project Goals

The project aims to:

- Develop a system to automate invoice data extraction.
- Train a machine learning model to recognize and extract invoice attributes such as invoice number, dates, vendor and buyer details, and financial information.
- Evaluate the system's performance on diverse datasets.

Dataset Details

- **Source:** labeled datasets containing PDF invoices.
- **Size:** Approx. 250 invoice pdfs.
- **Features:** Include INVOICE_NO**,** DATE_OF_ISSUE, SELLER**,** BUYER**,** order_id**,** Net_worth**,** Gross_worth**, and** IBAN**.**
- **Annotation Tool:** Annotated using **Doccano** deployed via **Docker**.

Related Work

Academic Research: Prior work includes leveraging **OCR** and rule-based methods, which often fall short in handling diverse formats. Our approach, based on **NER Transformer Model** and fine-tuned models, builds on these foundations to improve accuracy and flexibility.

Methodology

The project followed a phased approach:

- **Research and Development**
  - Evaluated libraries like **LangChain** and **spaCy**.
  - Identified the need for a custom model to address diverse invoice formats.
- **Data Annotation and Preprocessing**
  - Annotated data with labels such as INVOICE_NO, DATE_OF_ISSUE, etc., using **Doccano**.
  - Preprocessed data by cleaning **OCR**-extracted text and aligning annotations for training.
- **Model Development**
  - **PDF to Text Conversion**: It starts with developing a **custom Python program** to extract text data from PDF invoices. The program processed invoices and converted them into text format suitable for annotation.
  - **Exporting Annotations**: Exported the labeled data from Doccano in **JSONL format**.
  - **Data Cleaning and Formatting**:
    - Convert the JSONL file to JSON
    - Cleaned the JSON data to ensure consistency
  - **SpaCy-Compatible Format**: Transformed the cleaned JSON data into spaCy's training format

Testing and Refinement

- **Model Evaluation:**
  - Initially, we worked with a large model that achieved 62% accuracy. Since we were aiming for accuracy above 70%, we decided to experiment with a transformer model.
  - Next, we trained the transformer model on labeled data using CPU support, which improved accuracy to 90%.
  - Finally, incorporating GPU support with the transformer model with fine tuning yielded a slight improvement, bringing the accuracy up to 95%.

Hyperparameter Tuning

- We set different values of epochs to enhance the training process and each change gave us a diverse outcome. Values like: 10, 50, 100.
- Adjusted random seeds, batch sizes like 42, 64, and 128, to observe variations in results and ensure reproducibility.
- Used a learning rate of 0.001, which provided stable training and helped avoid overfitting or underfitting.

Tools and Technologies

- **Doccano:** Doccano is an open-source text annotation tool. It can be used to create labeled datasets for: Text classification
- **Docker:** Environment for deploying Doccano.
- **spaCy:** A free and open-source Python library called spaCy offers sophisticated features for quickly processing large amounts of text through natural language processing (NLP).

Data Exploration and Preprocessing

- After labeling the data in Doccano, we downloaded it in JSONL format and then converted it to JSON and SpaCy formats.
- We then cleaned the data and subsequently validated it.
- We cleaned repeated and overly spaced data, classified unclassified entries, and removed any unwanted data.

Feature Engineering

- Cleaned text by removing irrelevant characters, punctuation, and applying tokenization.
- Normalized numerical features (e.g., dates, quantities) to standard formats.
- Used SpaCy’s NER to label entities and applied custom labeling for domain-specific terms.
- Addressed ambiguities and overlapping entities through fine-tuning.

Data Analysis and Experimental Results

Models Used

- ·Large Model (en_core_web_lg):
- English transformer pipeline(Transformer(name='roberta-base', piece_encoder='byte-bpe', stride=104, type='roberta', width=768, window=144, vocab_size=50265)). Components: transformer, tagger, parser, ner, attribute_ruler, lemmatizer.
- Type: Vocabulary, syntax, entities, vectors.
- Genre: written text (blogs, news, comments).
- Size: 382 MB
- Vectors 685k keys, 343k unique vectors (300 dimensions)
- Accuracy: 62%

Transformer Model (en_core_web_trf)

- English transformer pipeline (Transformer(name=‘roberta-base’, piece_encoder=‘byte-bpe’, stride=104, type=‘roberta’, width=768, window=144, vocab_size=50265)). Components: transformer, tagger, parser, ner, attribute_ruler, lemmatizer.
- Type: Vocabulary, syntax, entities.
- Genre: written text (blogs, news, comments)
- Size: 436 MB
- Vectors 0 keys, 0 unique vectors (0 dimensions)
- Accuracy: 94% with CPU, 95% with GPU

Why Transformer is used

- It leverages advanced self-attention mechanisms, which allow it to capture long-range dependencies and variations across invoices, making it more robust for tasks requiring deep contextual understanding.
- In comparison to simpler models like en_core_web_lg, it provided significantly better results.

Benchmarking:

Baseline Performance

- Initial performance metrics using a baseline model en_core_web_lg achieved 62% accuracy.
- **Improved Model Performance**
  - Performance metrics after transitioning to an advanced model en_core_web_trf achieving 95% accuracy.
  - Include specific metrics like precision, recall, F1-score, and execution time.
- **Efficiency Metrics**
  - Computational efficiency with CPU 94% and GPU 95%.
  - ents_ f = 0.95, ents_p = 0.95, ents_r = 0.95.
  - Ner_loss = 27.315.
- **Error Analysis**
  - Common errors or limitations observed and steps taken to reduce them.
- **Business Value**
  - Highlight the real-world impact of improved accuracy, such as time saved, reduced manual effort, or improved reliability.

Fairness Audit

The model was evaluated for fairness across invoices from various vendors to ensure unbiased extraction.

Conclusion

- Implementing AI for tailored use cases, such as invoice processing, significantly enhances user efficiency.
- Future advancements in training for broader real-world applications can be achieved using Amazon Sage Maker's automated annotation capabilities.
- Leveraging a transformer-based model, the Invoice IQ system successfully boosted invoice data extraction accuracy from 62% to an impressive 95%.
- Efficiency was further optimized through GPU acceleration, highlighting the critical role of advanced models and continuous refinement for practical deployment.
- This project exemplifies the application of supervised machine learning in solving real-world challenges.

Team Contributions

- **Harsh Khatri (UIN - 657600128)**: Managed project schedules, created Docker, labeled datasets in Doccano, divided data into training and testing sets, trained the model on the data, and tracked developments.
- **Saurabh Chauhan (UIN - 669573191)**: Developed the technical pipeline, built the **OCR** and machine learning models, and implemented model refinement. Contributions: Managed timelines, trained model.

References

- https://dl.acm.org/doi/10.1145/3590003.3590034
- https://ieeexplore.ieee.org/abstract/document/8977951
- https://cs229.stanford.edu/proj2016/report/LiuWanZhang-
- UnstructuredDocumentRecognitionOnBusinessInvoice-report.pdf