Proceedings of Machine Learning Research 225:1–11, 2023 Machine Learning for Health (ML4H) 2023

Eigen: Expert-Informed Joint Learning Aggregation for High-Fidelity Information Extraction from Document Images

Abhishek Singh[1] abhisheksingh@cse.iitb.ac.in
Venkatapathy Subramanian[1] venkatapathy@cse.iitb.ac.in
Ayush Maheshwari[1] ayusham@cse.iitb.ac.in
Pradeep Narayan[2] pradeep.narayan.dr@narayanahealth.org
Devi Prasad Shetty[2] devi.shetty@narayanahealth.org
Ganesh Ramakrishnan[1] ganesh@cse.iitb.ac.in
[1] *Indian Institute of Technology Bombay*
[2] *Narayana Health*

Abstract

Information Extraction (IE) from document images is challenging due to the high variability of layout formats. Deep models such as LayoutLM and BROS have been proposed to address this problem and have shown promising results. However, they still require a large amount of field-level annotations for training these models. Other approaches using rule-based methods have also been proposed based on the understanding of the layout and semantics of a form such as geometric position, or type of the fields, *etc.* In this work, we propose a novel approach, EIGEN (Expert-Informed Joint Learning aGgrEatioN), which combines rule-based methods with deep learning models using data programming approaches to circumvent the requirement of annotation of large amounts of training data. Specifically, Eigen consolidates weak labels induced from multiple heuristics through generative models and use them along with a small number of annotated labels to jointly train a deep model. In our framework, we propose the use of labeling functions that include incorporating contextual information thus capturing the visual and language context of a word for accurate categorization. We empirically show that our Eigen framework can significantly improve the performance of state-of-the-art deep models with the availability of very few labeled data instances.

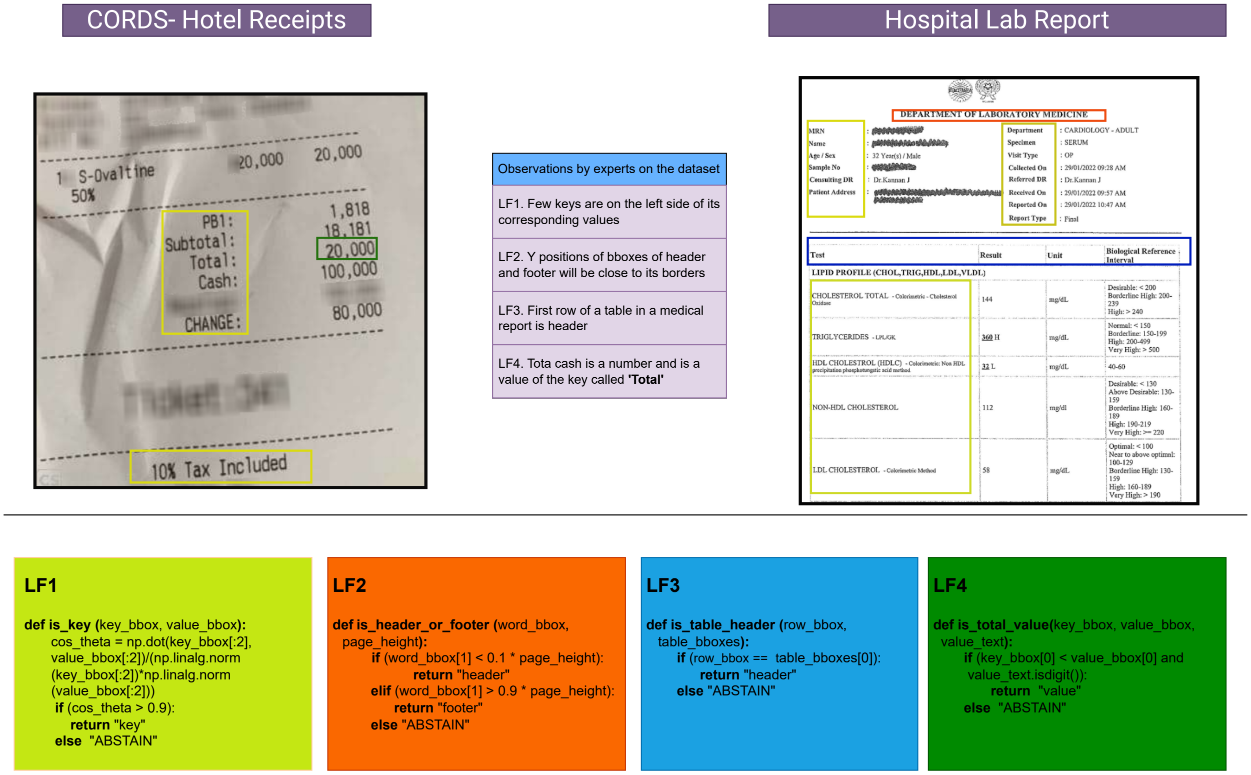
1. Introduction

In today's information-driven world, the ability to efficiently extract and process information from document images is crucial for various applications, ranging from document management systems to intelligent search engines. Large-scale pre-trained language models, such as BERT (Devlin et al., 2018), [1]

In today's information-driven world, the ability to efficiently extract and process information from document images is crucial for various applications, ranging from document management systems to intelligent search engines. Large-scale pre-trained language models, such as BERT (Devlin et al., 2018), GPT (Radford et al.), and RoBERTa (Liu et al., 2019), have demonstrated exceptional performance in various NLP tasks, including named entity recognition (NER) and relation extraction, which are key components of information extraction (IE). Although advances in large language models (LLMs) (Wolf et al., 2020) have led to significant progress in natural language understanding and processing (Zhao et al., 2023), the task of high-fidelity information extraction from document images remains a challenging endeavor. State-of-the-art models like LayoutLM (Xu et al., 2020) and DocVQA (Mathew et al., 2021) combine visual and textual information to better understand document layouts and answer questions about document content, addressing the issue of diverse document formatting.

While many such LLM models (Xu et al., 2020; Hong et al., 2022) that combine language and visual representation have outperformed all previous approaches in IE from document images, they still need to be fine-tuned for specific tasks in order to yield optimal performance. This introduces certain disadvantages that may hinder their widespread adoption

[1] The Source code is available at https://github.com/ayushayush591/key_value_extraction_jl



[redacted]

1 S-Ovaltine 20,000 20,000

50%

PB1: 1,818

Subtotal: 18,181

Total: 20,000

Cash: 100,000

[redacted]

CHANGE: 80,000

[redacted]

10% Tax Included

DEPARTMENT OF LABORATORY MEDICINE

MRN	: [redacted]	Department	: CARDIOLOGY - ADULT
Name	: [redacted]	Specimen	: SERUM
Age / Sex	: 32 Year(s) / Male	Visit Type	: OP
Sample No	: [redacted]	Collected On	: 29/01/2022 09:28 AM
Consulting DR	: Dr.Kannan J	Referred DR	: Dr.Kannan J
Patient Address	: [redacted]	Received On	: 29/01/2022 09:57 AM
		Reported On	: 29/01/2022 10:47 AM
		Report Type	: Final

Test	Result	Unit	Biological Reference Interval
LIPID PROFILE (CHOL,TRIG,HDL,LDL,VLDL)			
CHOLESTEROL TOTAL - Colorimetric - Cholesterol Oxidase	144	mg/dL	Desirable: < 200 Borderline High: 200-239 High: > 240
TRIGLYCERIDES - LPL/GK	360 H	mg/dL	Normal: < 150 Borderline: 150-199 High: 200-499 Very High: > 500
HDL CHOLESTROL (HDLC) - Colorimetric: Non HDL precipitation phosphotungstic acid method	32 L	mg/dL	40-60
NON-HDL CHOLESTEROL	112	mg/dl	Desirable: < 130 Above Desirable: 130-159 Borderline High: 160-189 High: 190-219 Very High: >= 220
LDL CHOLESTEROL - Colorimetric Method	58	mg/dL	Optimal: < 100 Near to above optimal: 100-129 Borderline High: 130-159 High: 160-189 Very High: > 190

Observations by experts on the dataset
LF1. Few keys are on the left side of its corresponding values
LF2. Y positions of bboxes of header and footer will be close to its borders
LF3. First row of a table in a medical report is header
LF4. Tota cash is a number and is a value of the key called **'Total'**

LF1

```
def is_key (key_bbox, value_bbox):
      cos_theta = np.dot(key_bbox[:2],
      value_bbox[:2])/(np.linalg.norm
      (key_bbox[:2])*np.linalg.norm
      (value_bbox[:2]))
       if (cos_theta > 0.9):
           return "key"
       else  "ABSTAIN"
```

LF2

```
def is_header_or_footer (word_bbox,
  page_height):
        if (word_bbox[1] < 0.1 * page_height):
                return "header"
        elif (word_bbox[1] > 0.9 * page_height):
             return "footer"
        else "ABSTAIN"
```

LF3

```
def is_table_header (row_bbox,
   table_bboxes):
        if (row_bbox ==  table_bboxes[0]):
             return "header"
        else "ABSTAIN"
```

LF4

```
def is_total_value(key_bbox, value_bbox,
  value_text):
        if (key_bbox[0] < value_bbox[0] and
         value_text.isdigit()):
              return  "value"
        else  "ABSTAIN"
```

Figure 1: Illustration of the Labeling Functions (LFs) creation process. We demonstrate how domain experts can leverage their knowledge to define LFs based on certain heuristics. Examples include, the position of specific fields within a document, the recognition of certain patterns or keywords in the text, or the spatial relationships between visual and textual elements. This encoding of expert knowledge enables our model to extend its learning from a few labeled data points to a much larger, unlabeled data set (The colour of boxes in image and LF same signifies that boxes classified by applying that particular labeling function).

and scalability, despite the humongous effort that goes into designing and training these models. Fine-tuning might become a bottleneck due to the following reasons: (i) High annotation cost, (ii) the possibility of labeling inconsistency, and quality degradation, and (iii) privacy where data cannot be shared for fine-tuning. In this work, we circumvent this bottleneck through the use of a semi-supervised approach of data-programming (Ratner et al., 2017) for such LLMs fine-tuning tasks. Data programming leverages labeling functions (LFs), which are a set of rules or heuristics created by domain experts or from prior knowledge. In our case, LFs can be used to encode knowledge such as the position of specific fields or regions within a document layout, patterns in textual content, semantic correlations between language and visual cues, or even domain-specific rules and conventions. For instance, in a standard invoice document, we know that the 'Invoice Number' is generally located at the top right corner; this is a positional heuristic that can be encoded as a labeling function. Similarly, we can encode patterns in text like those for recognizing dates, and monetary amounts, or for identifying certain keywords indicative of specific fields. Furthermore, LFs can help to encode the semantic relationship between visual and textual elements, such as the spatial proximity of text to specific symbols or images within the document, or the presence of certain textual content within specific visual containers. Domain-specific rules and conventions, such as the format of a medical prescription, tax invoice, or legal contract, can also be codified into LFs.

In Figure 1, we visually demonstrate how LFs can be created by experts based on a few example data points. However, these LFs may be (i) conflicting in nature *i.e.*, multiple LFs may assign conflicting labels to the same instance, and (ii) some LFs may not cover the complete dataset. Unsupervised (Ratner et al., 2017; Chatterjee et al., 2020) data programming approaches aggregate these conflicting labels only on unlabeled set. Semi-supervised data programming approaches (Awasthi et al., 2020; Maheshwari et al., 2021) leverage both unlabeled and labeled sets to further improve the final performance. They accept LFs, which learn a label aggregation model, and a small number of labeled instances, which learn a supervised feature-based model. Both of these models are jointly learned for improved performance on the end task. Summarily in this work, we combine the power of large language models with semi-supervised data programming to create a robust, scalable, and cost-efficient method for high-fidelity information extraction from document images, which we name Eigen(Expert-Informed Joint Learning aGgregation). Our contributions can be summarised as follows:

1. We introduce Eigen, a novel framework that integrates human-in-the-loop learning with the capabilities of language models through the utilization of data-programming techniques.
2. Within the Eigen framework, we present a methodology for defining contextual labeling functions specifically tailored to three distinct datasets capturing domain-specific information.
3. We provide empirical evidence showcasing the efficacy of Eigen and user-defined rules in circumventing the need for annotating a large number of domain-specific datasets. We conduct extensive experiments on three datasets (two public and one proprietary) and show improvements over state-of-the-art language models.

2. Related Work

Transformer models have proven to be very effective in recognition tasks and data programming. They have been widely used in document pre-training, but traditional pre-trained language models (Zhao et al., 2023) focus on text-level information, leaving out layout information. To address this, LayoutLM (Xu et al., 2020) was introduced, which leverages both text and layout information to significantly improve performance on various document understanding tasks. LayoutLM uses language models and image-text matching to find relationships between text and document layout, taking text, image, and location as input features. Its common functionalities include visual feature extraction, textual feature extraction, spatial relationship modeling, pre-training, and fine-tuning for document images and associated text.

The improved LayoutLMv2 (Xu et al., 2021) further utilizes self-attention with a spatially-aware model to better capture the layout and position of different text blocks in the document. These pretrained models work well for document classification and token labeling, but they are unable to learn geometric relationships since they use only absolute 1-D positional embeddings. Further improvement were made with LayoutLMv3 (Huang et al., 2022), which is similar to V2 but takes images as input in the RGB format instead of BGR format as used by V1 and V2. Further, unlike V1 and V2, which used WordPiece for text tokenization, LayoutLM V3 uses byte-pair encoding.

Weak supervision (Maheshwari et al., 2021; Sivasubramanian et al., 2023), a machine learning approach that deals with limited or noisy labeled training data, has also seen significant applications in document understanding. This approach requires heuristics to be applied to unlabeled data and the aggregation of noisy label outputs to assign labels to unlabeled data points(Maheshwari et al., 2022). Unsupervised approach such as Snorkel (Ratner et al., 2017) uses domain experts to develop heuristics, referred to as labeling functions, which output noisy labels that are aggregated using a generative model instead of a simple majority vote. Snuba Varma and Ré (2018) was later introduced to automate the creation of heuristics, making it simpler and more convenient for users.

However, the use of discrete labeling functions can leave gaps in the labeling process. To address this, CAGE (Chatterjee et al., 2020), or ‘Data Programming using Continuous and Quality-Guided Labeling Functions’ was introduced, which uses continuous labeling functions to extract more accurate information for labeling and introduces a Quality Guide that extends the functionality of the generative model for aggregation. This user-controlled variable can effectively guide the training process of CAGE.

3. Methodology

Like for any visual NER task, for Eigen framework, we start with a small set of document images where each image contains words, associated bounding box (bbox) coordinates, and the respective class to which each word belongs. Additionally, we have a large set of images where only the words and their bbox coordinates are annotated. The classes for the words in these images remain unlabelled, thereby forming a semi-supervised data set. To complement these data sets, a set of Labeling Functions (LFs) are also provided. These LFs are designed to capture the heuristic rules based on domain knowledge and document layouts. They play a pivotal role in providing surrogate labels for the words in the larger unlabeled data set, thereby extending the reach of our supervised training mechanism. In our framework, we also leverage two models: the large language model (LLM) for information extraction from document images and a probabilistic model for label aggregation. The LLM can be any state-of-the-art model that has demonstrated robust performance in document understanding tasks, such as LayoutLM or DocVQA. This model's role is to predict the class labels of words in the document images, given the words and their bbox coordinates. The probabilistic model is used for aggregating the labels produced by the LFs. When multiple LFs give conflicting labels for a particular word, this model, based on the parameters reflecting the reliability scores of each LF, determines which label to assign to the word. This model helps reconcile conflicts and uncertainties among the LFs, ensuring a reliable and consistent labeling system that guides the learning process of the LLM. To fine-tune the LLM and train the probabilistic model, Eigen uses both the small labeled data set and the large unlabeled data set. The LFs are applied to all words in both data sets, producing surrogate labels for the words. In the case of the small labeled data set, each word now has two labels: the original human-annotated label and the LF-generated surrogate label. In the case of the large unlabeled data set, each word only has the surrogate label.

The entire process is presented in Figure 2. The methodology is divided into three main stages:

1. Pre-processing: Eigen utilizes Optical Character Recognition (OCR) techniques to extract text from the images, and layout analysis tools to identify the spatial structure and relationships between different elements within the documents. This step provides a unified representation of the document that can be effectively utilized by LLMs.

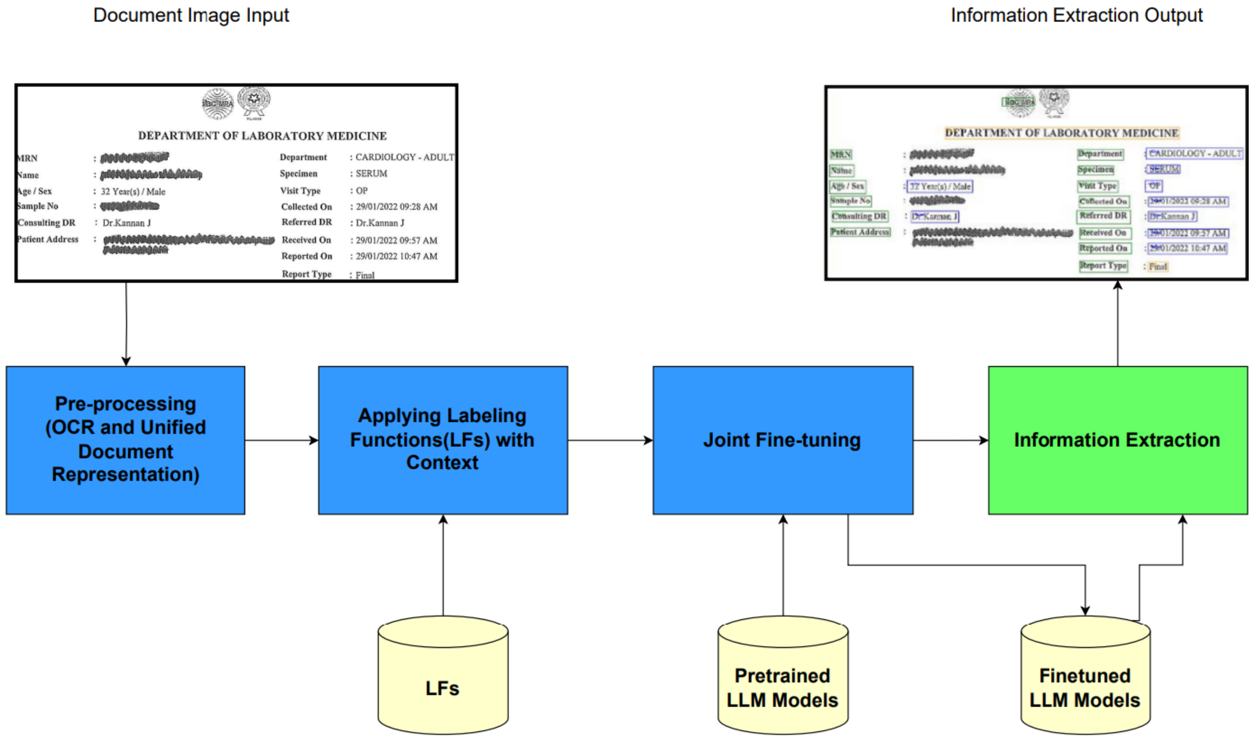
2. Labeling Function Design: In this stage, for Eigen, we develop a set of labeling functions (LFs) that can generate approximate labels for the training data. These LFs are heuristics or weak supervision sources, designed based on domain knowledge and available resources, such as dictionaries, rule-based systems, or pre-trained models. The LFs are designed to capture specific patterns and structures in document images relevant to the target information extraction tasks, such as named entity recognition and relation extraction. Several previous approaches to NER apply ruled-based or some heuristic methods. In our methodology, we utilize these rule-based methods as wrappers to our LFs.

3. Joint Fine-tuning: The joint fine-tuning process incorporates the designed LFs into the training loop of LLMs. The model is initially pre-trained on a large corpus of text using unsupervised learning, followed by supervised fine-tuning with the weak supervision provided by the LFs. During fine-tuning, the model learns to focus on the patterns and structures captured by the LFs, which enhances its ability to perform information extraction tasks on document images. This joint fine-tuning approach allows the model to leverage both the power of LLMs and the flexibility of LFs, leading to improved extraction accuracy and robustness.

3.1. Framework

Eigen framework consists of a pre-trained deep neural network model that tags each word with a corresponding entity class. In Eigen, we consider the recent LayoutLM (Xu et al., 2020) as our choice of the pre-trained deep neural network model, though this model can be replaced with any other deep neural model for visual NER tasks such as BROSHong et al. (2022), *etc.* We call this a featurized pre-trained deep model. Featurized model can be trained in a supervised setting with the availability of labeled data. We also utilize a graphical model as proposed in Maheshwari et al. (2021) which, along with a set of labeling functions(LFs) can be used to pseudo-label unlabelled words with the entity class by aggregating the output from the LFs.

Formally, let $\mathcal{X}$ and $\mathcal{Y} \in \{1...K\}$ be the feature and label spaces, respectively. A feature, $x_i \in \mathcal{X}$, consists of a word w_i and its corresponding bounding box b_i. For each feature x_i, the context set $\mathcal{C}$ where $\mathcal{C} \subseteq \mathcal{X}$


Document Image Input
Information Extraction Output
Pre-processing (OCR and Unified Document Representation)
Applying Labeling Functions(LFs) with Context
Joint Fine-tuning
Information Extraction
LFs
Pretrained LLM Models
Finetuned LLM Models

DEPARTMENT OF LABORATORY MEDICINE

MRN	: [redacted]	Department	: CARDIOLOGY - ADULT
Name	: [redacted]	Specimen	: SERUM
Age / Sex	: 32 Year(s) / Male	Visit Type	: OP
Sample No	: [redacted]	Collected On	: 29/01/2022 09:28 AM
Consulting DR	: Dr.Kannan J	Referred DR	: Dr.Kannan J
Patient Address	: [redacted]	Received On	: 29/01/2022 09:57 AM
		Reported On	: 29/01/2022 10:47 AM
		Report Type	: Final

Figure 2: Illustration of the joint learning process in the EIGEN framework. The process is divided into three main stages: (1) pre-processing, where the document images are annotated with bounding box coordinates and labels (if available), (2) Labeling Function (LF) design, where domain-specific heuristic rules are applied to generate surrogate labels, and (3) joint fine-tuning, where LLM and a probabilistic model are simultaneously trained using both the human-annotated labels and the LF-generated surrogate labels. This methodology enables robust Named Entity Recognition (NER) from document images leveraging semi-supervised learning.

and $\mathcal{C} = \{\forall c_i \in \mathcal{X} \setminus \{x_i\}\}$ $\mathcal{C}$ represents the surrounding words w_j and their respective bounding boxes b_j for the instance x_i. This context acts as the prior information for w_i and provides valuable information in the form of labeling functions. Furthermore, we have m LFs, $\lambda_1 \dots \lambda_m$, designed by either some prior knowledge or by inspecting very few examples of a specified document type, such as the few labeled data instances used for the initial training. Each LF λ_j is attached to one of the class $k_i \in K$, that takes an x_i, some context set $\mathcal{C}$, as input, and returns either k_i or 0 (which means ABSTAIN). Intuitively, LFs can be written to jointly understand the visual and language context of a word with respect to other words (specified by $\mathcal{C}$ in our framework) in a document image and can classify the word to a particular class it belongs to. The entire available dataset can be grouped into two categories:

- $\mathcal{L} = \{(x_1, y_1, l_1), .., (x_N, y_N, l_N)\}$ which denotes the labelled set and,
- $\mathcal{U} = \{(x_{N+1}, l_{N+1}, .., (x_M, l_M)\}$ which denotes the unlabelled set.

Here $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$ and $l_i = (l_{i1}, l_{i2}, ..., l_{im})$ denotes the firings of all the LFs on instance x_i.

Our joint learning model, borrowed from Maheshwari et al. (2021), is a blend of the feature-based model $P_\phi^f(x)$ and the LF-based graphical model $P_\phi(l_i, y)$. Our feature-based model, $P_\phi^f(x)$, is a Transformer-based neural network model Xu et al. (2020). For a given input x_i, the model outputs the probability of classes as $P_\phi^f(y|x)$. The LayoutLM is based on the Devlin et al. (2018) multi-layer bidirectional language model. The model computes the input embeddings by processing the corresponding word, position, and segment embeddings.

For each input x_i and LF outputs l_i, our goal is to learn the correct label y_i using a generative model on the LF outputs.

$$P_\theta(l_i, y) = \frac{1}{Z_\theta} \prod_{j=1}^{m} \psi(l_{ij}, y) \tag{1}$$

$$\psi_\theta(l_{ij}, y) = \begin{cases} \exp(\theta_{jy}) & \text{if } l_{ij} \neq 0 \\ 1 & \text{otherwise.} \end{cases} \tag{2}$$

For each LF l_j, we learn K parameters $\theta_{j1}, \theta_{j2} \ldots \theta_{jK}$ corresponding to each LF and class. Here, Z_θ is a normalization factor. The generative model assumes that each LF l_j is independent of other LFs and interacts with y_i to learn parameters θ. The model imposes a joint distribution between the true label y and the values l_i returned by each LF λ_i on the sample x_i. In this paper, we use a joint learning algorithm with semi-supervision to leverage both features and domain knowledge in an end-to-end manner.

3.2. Joint Learning (JL)

Our JL algorithm consists of two individual model loss and a KL divergence component to strengthen agreement among model predictions. We first specify the objective function of our JL framework and thereafter explain each component below:

$$\min_{\theta,\phi} \sum_{i \in L} L_{CE}\left(P^f_\phi(y|_i), y_i\right) + LL_u(\theta|U) +$$
$$\sum_{i \in \cup} KL\left(P^f_\phi(y|x_i), P_\theta(y|l_i)\right) + R(\theta|\{q_j\})$$

Feature Model Loss : The first component of the loss is the LayoutLM (Xu et al., 2020) loss over labeled data. The loss is defined as: $L_{CE}\left(P^f_\phi(y|_i), y_i\right) = -\log\left(P^f_\phi(y = y_i|x_i)\right)$ which is the standard cross-entropy loss on the labeled dataset L, toward learning ϕ parameters.

Graphical Model Loss: We borrow the graphical model loss from Chatterjee et al. (2020) which formulates $LL_u(\theta|U)$ as the negative log-likelihood loss for the unlabelled dataset. $LL_u(\theta|U) = -\sum\limits_{i=N+1}^{M} \log \sum\limits_{y \in Y} P_\theta(l_i, y)$, where P_θ is defined in Equation 1.

Kullback-Leibler (KL) divergence : $KL(P^f_\phi(y|x_i), P_\theta(y|l_i))$ aims to establish consensus among the models by aligning their predictions across both the labeled and unlabeled datasets. We use KL divergence to make both the models agree in their prediction over the union of labeled and unlabeled datasets.

Quality Guides: Following Chatterjee et al. (2020), we employ quality guides denoted as $R(\theta|q_j)$ to enhance the stability of unsupervised likelihood training while utilizing LFs. Let q_j be the fraction of cases where l_j is correctly triggered, and let q^t_j represent the user's belief regarding the proportion of examples $_i$ for which the labels y_i and l_{ij} agree. In cases where the user's beliefs are not accessible, we utilize the precision of the LFs on the validation set as a proxy for the user's beliefs. If $P_\theta(y_i = k_j|l_{ij} = 1)$ is the model precision associated with the labeling functions (LFs), the loss function guided by the quality measures can be expressed as:

$R(\theta|\{q^t_j\}) = \sum_j q^t_j \log P_\theta(y_i = k_j|l_{ij} = 1) + (1 - q^t_j) \log(1 - P_\theta(y_i = k_j|l_{ij} = 1))$

Each term is weighted by the user's beliefs q^t_j concerning the agreement between the LFs and the true labels, and their complement $(1 - q^t_j)$. This loss formulation serves as a guiding principle to optimize the model's performance based on the model predictions and the user's beliefs.

The two individual model-specific loss components are invoked on the labeled and unlabeled data respectively. Feature model loss learns ϕ against ground truth in the labeled set whereas graphical model loss learns θ parameters by minimizing negative loss likelihood over the unlabeled set using labeling functions. Using KL divergence, we compare the probabilistic output of the supervised model f_θ against the graphical model $P_\theta(l, y)$ over the combination of unlabeled and labeled datasets. We use the ADAM optimizer to train our non-convex loss objective

4. Experiment

We present here the experiments conducted to evaluate the performance of our proposed joint fine-tuning approach.

4.1. Dataset

We conducted our experiments on a diverse set of benchmark datasets that encompass various information extraction tasks, such as named entity recognition (NER), relation extraction, and question answering on document images. These datasets represent different document structures, domains, and complexities, thereby providing a comprehensive evaluation of our approach:

SROIE (Huang et al. (2019)): This dataset consists of English receipts, containing a total of 973 scanned receipts. Each receipt is accompanied by a .jpg file of the scanned image, a .txt file holding OCR information, and a .txt file containing the key information values.

CORD (Park et al. (2019): The Consolidated Receipt Dataset for post-OCR parsing (CORD) is a collection of receipt images obtained from shops and restaurants. The dataset consists of more than 11,000 image and JSON pairs, providing a rich source of data for information extraction tasks.

Hospital Dataset: In addition to the publicly available datasets, we are using a medical dataset provided by a Hospital. This anonymized dataset primarily consists of lab reports such as Biochemistry, Clinical Pathology, Discharge Summaries, Haematology, and Molecular Laboratory reports. The dataset includes 1000, images, of which 800 images are annotated with text boxes, and 100 images are annotated and labeled with respective tags.

4.2. Baseline

We establish the baseline by training the LayoutLM-v1(version1)(Xu et al., 2020) and LayoutLM-v3(version3)(Huang et al., 2022) model on a limited amount of labeled data. From the complete labeled training set, we randomly select a small percentage of images for training purposes - typically 1%, 5%, or 10% of the total training set. It should be noted that the validation and test sets remain constant across all these scenarios. After training the LayoutLM with these differing quantities individually, we calculate the scores to establish the baseline.

When baseline systems are trained on 100% labeled data, it forms a skyline for our experiments. For **CORD** dataset, LayoutLM was trained on all 800 labeled training instances. Similarly, for the **Hospital** and **SROIE** dataset, we trained LayoutLM on 364 and 626 labeled images respectively.

4.3. Implementation Details

We used the LayoutLM (Xu et al., 2020) model as the base LLM for our experiments, as it has shown strong performance in information extraction tasks on document images. We implemented our approach Eigen, using the Hugging Face Transformers library (Wolf et al., 2020). We fine-tuned Eigenmodel using a batch size of 16 and a learning rate of 5e-5. We used the AdamW optimizer (Kingma and Ba, 2014) and a linear learning rate schedule with a warm-up period of 0.1 times the total training steps. The maximum training epochs were set to 5, and early stopping was employed based on the performance of the validation set.

We used Abhishek et al. (2022) for LF design and JL training. SPEAR framework provides a useful visualization tool to help us better understand and optimize the performance of LFs and JL. The tool assists in the rapid prototyping of LFs, providing an iterative and user-friendly interface for designing and refining these functions. Not only does it allow the visualization of LF performance statistics, but it also aids in identifying potential areas of conflict, overlap, and coverage amongst the LFs, which can significantly enhance the accuracy of weak supervision. In Appendix (Figure 3), we present a detailed visualization of the performance of our LFs model on the CORD dataset. Overall, these results underline the strength of our proposed Eigen method in terms of leveraging smaller proportions of labeled data to achieve superior performance across diverse datasets.

4.4. Setting

The Eigen model consists of CAGE jointly finetuned with the (pretrained) LayoutLM. We achieve this by replacing the simple neural network model in SPEAR by LayoutLM. We evaluate the performance of models using F1-score.

- For **CORD**, only 1000 samples are publicly available. We divide the dataset into 3 parts, *viz.*, train, test, and validation, having sizes of 800, 100, and 100 images respectively. Though the dataset contains 30 labeled classes, for our work, we consider only three labels namely Menu, Dish, and Price.
- For **Hospital Dataset**, we have 413 images which are further divided into 3 parts train, test, and validation. 364, 25, and 24 respectively. The labels associated with this are ‘Field’, ‘Value’, and ‘Text’.
- For **SROIE**, we got 973 images in that 626 are training and the rest are divided into two sets

% of L	Dataset	Model		
		Base-v1	Base-v3	Eigen
1%	CORD	0.684	0.685	**0.772**
	SROIE	0.236	0.058	**0.487**
	Hospital	0.301	0.212	**0.689**
5%	CORD	0.894	0.830	**0.896**
	SROIE	0.585	0.605	**0.647**
	Hospital	0.854	0.829	**0.865**
10%	CORD	0.905	0.844	**0.905**
	SROIE	0.698	0.656	**0.715**
	Hospital	0.862	0.883	**0.928**
		Skyline		
100%	CORD	0.963	0.965	
	SROIE	0.842	0.839	
	Hospital	0.961	0.961	

Table 1: F1 score of Eigen on various dataset and comparison with different versions LayoutLM baseline having varying amounts of labeled data (L). We also present skyline numbers for the baselines when the entire training data is used as labeled set.

one is validation which contains 173 images and test contains 174 images.

4.5. Results

Table 1 shows the performance of Eigenresults on different datasets with varying percentage of labeled set. We observe thatEigen consistently outperforms the LayoutLM baselines, particularly when limited quantities of labeled data is present. When the models are trained with 1% labeled data, Eigen achieves superior performance on all datasets. For instance, in the case of the SROIE dataset, baseline systems achieve less than 0.1 F1-score whereas Eigen achieves an F1-score of 0.48. We observe similar trend when labeled data is increased to 5% and 10%.

When the entire training dataset is treated as labeled, it can be viewed as a skyline. We obtain a skyline model for our baseline models, namely LayoutLM-v1 and LayoutLM-v3. We achieve 0.979, 0.842 and 0.961 F1-score on CORD, SROIE, and Hospital dataset for the LayoutLM-v1 model. Understandably, Eigen scores are lower than the skyline numbers mentioned in Table 1. However, with small amounts of labeled data, Eigen scores are closer to these numbers.

% of L	% of U	Dataset	F1
1%	90%	CORD	0.735
	95%		0.725
	97%		0.757
1%	90%	Hospital	0.590
	95%		0.602
	97%		0.689

Table 2: F1 score of Eigen on various Datasets, when % of L(labeled) is kept fixed and % of U(unlabeled) set is varying.

5. Ablation Study

5.1. When labelled data is fixed

To observe the impact of unlabeled loss components on the final performance of Eigen, we kept the amount of labeled data as fixed and varying the quantity of unlabeled data. Table 2 presents the performance of Eigen with 1% labeled data and varying proportions of unlabeled data, specifically 90%, 95%, and 97%. It is evident from the results that there is a consistent improvement in the F1-score as the volume of unlabeled data increases. This underscores the significance of joint learning with the unlabeled loss component (Graphical Model Loss) in our Eigen framework.

5.2. When unlabelled data is fixed

To understand the significance of labeled loss components in the overall framework, we conduct an experiment in which the unlabeled set is constant, while the quantity of labeled data is varying. In Table 3, we present the performance of Eigen on CORD and Hospital dataset with varying quantities of labeled data. We observe that increasing labeled data from 1% to 5% leads to significant improvements in the F1-score. However, we do not observe a commensurate improvement when the labeled data is further increased from 5% to 10%. We observe marginal improvements when percentage of labeled dataset exceeds 5%. The feature model demonstrate the ability to harness the labeled data effectively, resulting in overall performance improvement. Both of these

% of L	% of U	Dataset	F1
1%	90%	CORD	0.735
5%			0.884
10%			0.905
1%	90%	Hospital	0.590
5%			0.872
10%			0.928

Table 3: F1 score of Eigen on various Datasets, when % of U(unlabeled) is kept fixed and % of L(labeled) set is varying.

.

ablation experiments signifies the importance of the unlabeled and labeled loss components, as well as the interaction between them, in our framework.

6. Conclusion

In this paper, we proposed Eigen, a joint fine-tuning approach for large language models along with data programming to improve the efficiency and accuracy of information extraction from document images. Eigen successfully leveraged the power of LLMs and the flexibility of labeling functions, resulting in information extraction from document images. LFs, used in our Eigen approach, provide a flexible, reusable, and efficient approach to learning from unlabeled data. They capture diverse heuristics, domain knowledge, and high-level patterns, which allow them to generalize well across various datasets. Instead of explicitly annotating each instance, we merely need to define high-level patterns or rules, thereby reducing the dependency on human annotation. As shown in our evaluation, Eigen achieves remarkable results even with as little as 1% or 5% of labeled data, across diverse datasets. This means we can reduce annotation efforts significantly without compromising on performance. This approach not only reduces the cost and time associated with data labeling but also enables models to learn from richer, diverse data sources, enhancing their generalizability and robustness.

7. Acknowledgement

We thank anonymous reviewers for providing constructive feedback, We acknowledge the support of a grant from IRCC, IIT Bombay, and MEITY, Government of India, through the National Language Translation Mission-Bhashini project and also the Koita Centre for Digital Health (KCDH-(www.kcdh.iitb.ac.in)) and Narayana Health (https://hospital.narayanahealth.org/) for providing hospital dataset which contains Lab report of a patient with various medical departments (cardiology, gastroenterology, oncology etc) in PDF format.

References

Guttu Abhishek, Harshad Ingole, Parth Laturia, Vineeth Dorna, Ayush Maheshwari, Ganesh Ramakrishnan, and Rishabh Iyer. Spear: Semi-supervised data programming in python. In *Proceedings of the The 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 121–127, 2022.

Abhijeet Awasthi, Sabyasachi Ghosh, Rasna Goyal, and Sunita Sarawagi. Learning from rules generalizing labeled exemplars. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=SkeuexBtDr.

Oishik Chatterjee, Ganesh Ramakrishnan, and Sunita Sarawagi. Robust data programming with precision-guided labeling functions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3397–3404, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Teakgyu Hong, Donghyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park. Bros: A pre-trained language model focusing on text and layout for better key information extraction from documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10767–10775, 2022.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. Layoutlmv3: Pre-training for document with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4083–4091, 2022.

Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and CV Jawahar. Icdar2019 competition on scanned receipt ocr and information extraction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1516–1520. IEEE, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Ayush Maheshwari, Oishik Chatterjee, Krishnateja Killamsetty, Ganesh Ramakrishnan, and Rishabh Iyer. Semi-supervised data programming with subset selection. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4640–4651, 2021.

Ayush Maheshwari, Krishnateja Killamsetty, Ganesh Ramakrishnan, Rishabh Iyer, Marina Danilevsky, and Lucian Popa. Learning to robustly aggregate labeling functions for semi-supervised data programming. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1188–1202, 2022.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209, 2021.

Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. Cord: a consolidated receipt dataset for post-ocr parsing. In *Workshop on Document Intelligence at NeurIPS 2019*, 2019.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners.

Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, volume 11, page 269. NIH Public Access, 2017.

Durga Sivasubramanian, Ayush Maheshwari, Prathosh AP, Pradeep Shenoy, and Ganesh Ramakrishnan. Adaptive mixing of auxiliary losses in supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

Paroma Varma and Christopher Ré. Snuba: Automating weak supervision to label training data. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, volume 12, page 223. NIH Public Access, 2018.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2579–2591, 2021.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1192–1200, 2020.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models, 2023.

Appendix A. Labeling function generation

As previously discussed in Section 1 and illustrated in Fig. 1, labeling functions entail the utilization of domain expert knowledge to construct functions that encapsulate specific knowledge relevant to the task. In our particular case, the need for a domain expert was obviated, as we employed rule-based labeling functions. These labeling functions incorporate a variety of techniques, including regular expressions and pattern matching rules. Additionally, we implemented context-based labeling, which takes into account not only patterns but also the positional relationship with respect to other words. An example of a **context-based** labeling function is illustrated in 1. These functions can become increasingly complex depending on the specific requirements of the dataset. For each new dataset, the creation of labeling functions is essential, and they can be formulated by examining a limited amount of labeled data or by drawing upon domain expert knowledge.

Therefore, it becomes evident that labeling functions contribute equivalently to what can be extracted by the feature model. It is of paramount importance to eliminate non-performing labeling functions and address conflicting ones, a task facilitated by the Quality Guide as described in 3.2. The evaluation of labeling functions can be conducted using specific metrics such as Coverage, Overlap, Conflicts, and others, all of which are already integrated into the CAGE model. For a visual representation of the performance of labeling functions on the CORD dataset, please refer to 3.

Appendix B. Miscellaneous Results

We conducted an experiment to assess the robustness of our approach (Eigen) by increasing the amount of labeled data. This experiment aimed to evaluate how our model performs when provided with a more substantial dataset. In Table 4, we present the experiment results and compare them with the performance of LayoutLMV1, which was fine-tuned using the same amount of data as the baseline. And It's evident from the table that the baseline occasionally outperforms EIGEN when labeled data is in the vicinity of 50%. This reaffirms our assertion: EIGEN truly shines when data is sparse. As more labeled data becomes accessible, the model naturally veers towards learning directly from the data rather than relying on weak functions.

Appendix C. Limition of Eigen

Crafting labeling functions isn't straightforward for all datasets, particularly when faced with high variability in layout, Labeling tricky key-value pairs is challenging using only these basic labeling functions, which is a concern for us. There is a significant amount of variability and ambiguity when creating labeling functions because, in some cases, a single word's class cannot be determined solely based on its semantic properties. (For example, certain words can be both keys and values), leading to confusion. Therefore, relying solely on the semantic meaning of a word is insufficient, and we must also take into account factors like its position, neighboring words, and structural properties. These considerations are essential not only for predicting the correct class for specific data but also for generalizing across future data. Even when humans are responsible for labeling, they might not always include all these valuable details in the labeling functions. Our ongoing research seeks to devise labeling functions rooted in exemplars.

Appendix D. Quantative Result

In our study, we presented quantitative results 4, where we showcased the inference outcomes of Eigen trained on 1% of labeled data using a sample Hospital dataset. During the inference process, the input image undergoes initial processing through the Doctr model, producing OCR output. Subsequently, this output serves as input for Eigen, leading to the classification of each token into specific classes. The resulting classifications are then projected onto the image to facilitate visualization and comprehension.

% of (L)	**Models**	**Hospital**		**SROIE**		**CORDS**	
		Acc	F1	Acc	F1	Acc	F1
20%	**Base**	0.982	0.898	0.966	0.693	0.983	0.951
	EIGEN	0.983	0.918	0.971	0.696	0.977	0.933
40%	**Base**	0.988	0.943	0.967	0.738	0.985	0.961
	EIGEN	0.984	0.919	0.970	0.735	0.985	0.957
60%	**Base**	0.985	0.935	0.985	0.824	0.991	0.967
	EIGEN	0.985	0.933	0.986	0.821	0.986	0.957
70%	**Base**	0.987	0.937	0.985	0.828	0.988	0.958
	EIGEN	0.987	0.933	0.985	0.807	0.989	0.958
80%	**Base**	0.988	0.945	0.988	0.852	0.985	0.948
	EIGEN	0.984	0.918	0.986	0.798	0.984	0.951

Table 4: F1 score and accuracy of Eigen on various dataset and comparison with LayoutLM V1 baseline having varying amounts of labeled data (L).

Performance on Val set					
% of Labeled Data	Method	Acc	F1	Precision	Recall
1%	CORD(Eigen)	**0.953**	**0.843**	**0.830**	**0.858**
5%	CORD(Eigen)	**0.973**	**0.908**	**0.897**	**0.919**
10%	CORD(Eigen)	**0.983**	**0.943**	**0.945**	**0.941**
1%	SROIE(Eigen)	**0.954**	**0.519**	**0.551**	**0.491**
5%	SROIE(Eigen)	**0.978**	**0.690**	**0.763**	**0.630**
10%	SROIE(Eigen)	**0.978**	**0.721**	**0.791**	**0.663**
1%	Hospital(Eigen)	**0.944**	**0.762**	**0.728**	**0.800**
3%	Hospital(Eigen)	**0.941**	**0.823**	**0.789**	**0.861**
5%	Hospital(Eigen)	**0.969**	**0.867**	**0.840**	**0.896**
10%	Hospital(Eigen)	**0.972**	**0.906**	**0.877**	**0.906**

Table 5: Comparative Performance of Eigen method on the Val Set Across Diverse Datasets and Proportions of Labeled Data

Performance on Test set					
% of Labeled Data	Method	Acc	F1	Precision	Recall
100%	CORD(sky-v1)	0.989	0.963	0.968	0.957
1%	CORD(Base-v1)	0.881	0.684	0.662	0.706
5%	CORD(Base-v1)	0.964	0.894	0.880	0.908
10%	CORD(Base-v1)	0.971	0.905	0.884	0.926
100%	CORD(sky-v3)	0.989	0.965	0.957	0.973
1%	CORD(Base-v3)	0.872	0.685	0.638	0.741
5%	CORD(Base-v3)	0.946	0.830	0.812	0.849
10%	CORD(Base-v3)	0.979	0.844	0.840	0.849
1%	CORD(Eigen)	**0.928**	**0.772**	**0.746**	**0.800**
5%	CORD(Eigen)	**0.973**	**0.896**	**0.873**	**0.921**
10%	CORD(Eigen)	**0.973**	**0.905**	**0.880**	**0.930**
100%	SROIE(Sky-v1)	0.987	0.842	0.819	0.865
1%	SROIE(Base-v1)	0.913	0.236	0.297	0.196
5%	SROIE(Base-v1)	0.953	0.585	0.535	0.646
10%	SROIE(Base-v1)	0.957	0.698	0.675	0.721
100%	SROIE(Sky-v3)	0.986	0.839	0.838	0.840
1%	SROIE(Base-v3)	0.906	0.058	0.122	0.038
5%	SROIE(Base-v3)	0.960	0.605	0.621	0.590
10%	SROIE(Base-v3)	0.965	0.656	0.703	0.614
1%	SROIE(Eigen)	**0.934**	**0.487**	**0.433**	**0.557**
5%	SROIE(Eigen)	**0.965**	**0.647**	**0.615**	**0.683**
10%	SROIE(Eigen)	**0.978**	**0.715**	**0.713**	**0.717**
100%	Hospital(sky-v1)	0.988	0.961	0.956	0.966
1%	Hospital(Base-v1)	0.827	0.301	0.245	0.390
3%	Hospital(Base-v1)	0.949	0.731	0.685	0.783
5%	Hospital(Base-v1)	0.974	0.854	0.849	0.859
10%	Hospital(Base-v1)	0.979	0.862	0.849	0.875
100%	Hospital(sky-v3)	0.989	0.961	0.954	0.968
1%	Hospital(Base-v3)	0.757	0.212	0.173	0.274
3%	Hospital(Base-v3)	0.886	0.5	0.473	0.53
5%	Hospital(Base-v3)	0.953	0.829	0.804	0.856
10%	Hospital(Base-v3)	0.970	0.883	0.870	0.898
1%	Hospital(Eigen)	**0.949**	**0.689**	**0.658**	**0.724**
3%	Hospital(Eigen)	**0.959**	**0.821**	**0.809**	**0.835**
5%	Hospital(Eigen)	**0.977**	**0.865**	**0.863**	**0.867**
10%	Hospital(Eigen)	**0.982**	**0.928**	**0.925**	**0.930**

Table 6: Comparative Performance of Baseline and Eigen method on the Test Set Across Diverse Datasets and Proportions of Labeled Data

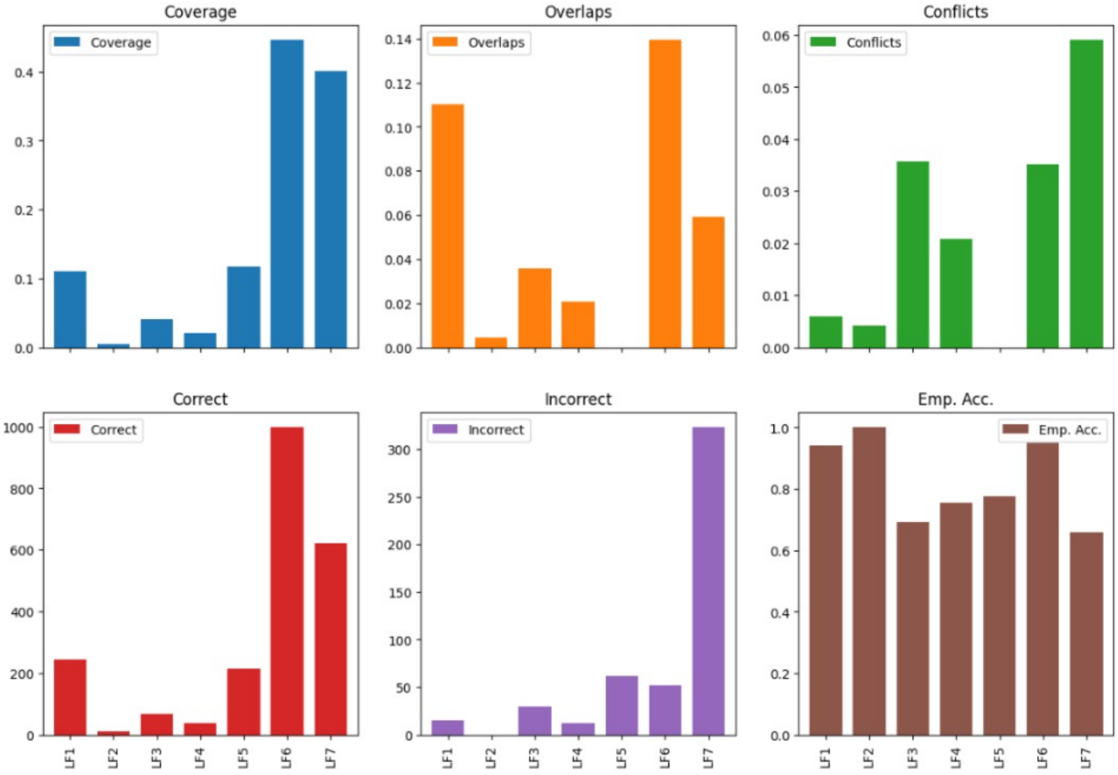
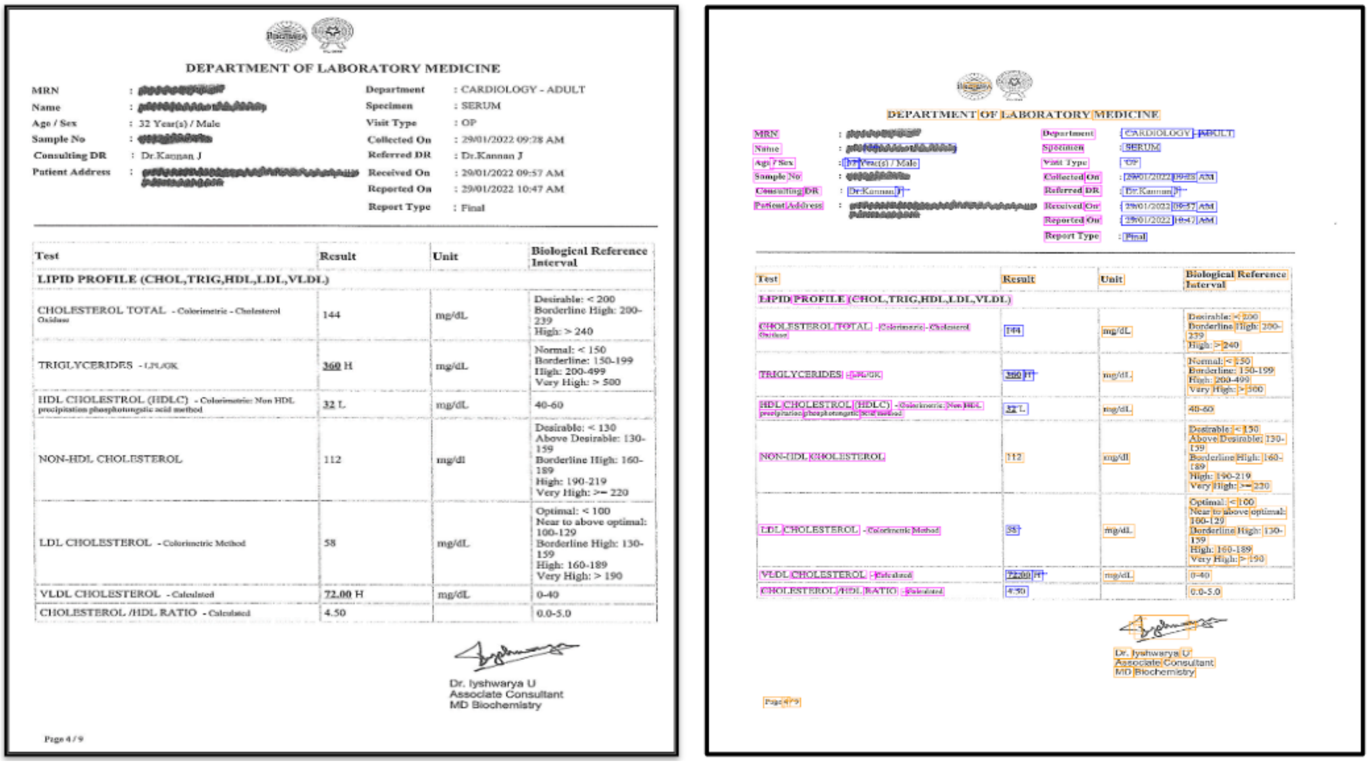


Figure 3: Comparison of the performance of the Labeling functions on the validation set of the CORD dataset.



DEPARTMENT OF LABORATORY MEDICINE

MRN	: [redacted]	Department	: CARDIOLOGY - ADULT
Name	: [redacted]	Specimen	: SERUM
Age / Sex	: 32 Year(s) / Male	Visit Type	: OP
Sample No	: [redacted]	Collected On	: 29/01/2022 09:28 AM
Consulting DR	: Dr.Kannan J	Referred DR	: Dr.Kannan J
Patient Address	: [redacted]	Received On	: 29/01/2022 09:57 AM
		Reported On	: 29/01/2022 10:47 AM
		Report Type	: Final

Test	Result	Unit	Biological Reference Interval
LIPID PROFILE (CHOL,TRIG,HDL,LDL,VLDL)			
CHOLESTEROL TOTAL - Colorimetric - Cholesterol Oxidase	144	mg/dL	Desirable: < 200 Borderline High: 200-239 High: > 240
TRIGLYCERIDES - LPL/GK	360 H	mg/dL	Normal: < 150 Borderline: 150-199 High: 200-499 Very High: > 500
HDL CHOLESTROL (HDLC) - Colorimetric: Non HDL precipitation phosphotungstic acid method	32 L	mg/dL	40-60
NON-HDL CHOLESTEROL	112	mg/dl	Desirable: < 130 Above Desirable: 130-159 Borderline High: 160-189 High: 190-219 Very High: >= 220
LDL CHOLESTEROL - Colorimetric Method	58	mg/dL	Optimal: < 100 Near to above optimal: 100-129 Borderline High: 130-159 High: 160-189 Very High: > 190
VLDL CHOLESTEROL - Calculated	72.00 H	mg/dL	0-40
CHOLESTEROL /HDL RATIO - Calculated	4.50		0.0-5.0

Dr. Iyshwarya U
Associate Consultant
MD Biochemistry

Page 4 / 9

Figure 4: **Quantative Result-** Sample Hospital data is when input to Eigen trained on 1% (i.e. 4 images) labeled images, Color of the boxes in right side image (i.e. output image) signifies that a particular token classified among one of the class (**Color-Class**: Magenta-field ,blue-value, orange-text).