UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering, Mathematics & Computer Science

Predicting Semantic Labels of Text Regions in Heterogeneous Document Images

Somtochukwu C. Enendu

Masters in Computer Science
Specialization: Data Science and Technology

Master Thesis
15th August, 2019

External Supervisors:
Dr. Johannes Scholtes
Email: Johannes.Scholtes@zylab.com
Jeroen Smeets
Email: Jeroen.Smeets@zylab.com

ZyLAB
Laarderhoogtweg 25,
1101 EB Amsterdam-Zuidoost
The Netherlands

Supervisors:
dr. ir. Djoerd Hiemstra
dr. Mariet Theune

Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Abstract

This paper describes the use of sequence labeling methods in predicting the semantic labels of extracted text regions of heterogeneous electronic documents, by utilizing features related to each semantic label. In this study, we construct a novel dataset consisting of real world documents from multiple domains. We test the performance of the methods on the dataset and offer a novel investigation into the influence of textual features on performance across multiple domains. The results of the experiments show that the Conditional Random Field method is robust, outperforming the neural network when limited training data is available. Regarding generalizability, our experiments show that the inclusion of textual features does not guarantee performance improvements.

Acknowledgements

I would first like to express my sincere gratitude to my thesis supervisors, Djoerd Hiemstra and Mariet Theune. This research work was made much easier with your support, patient guidance, constant feedback, and useful critiques. Thank you, Djoerd for creating time for our Skype meetings and for always pointing me towards the right direction. Thank you Mariet for; always monitoring the progress of my work and your many emails that always motivated me. I would also like to thank Jan Scholtes and Jeroen Smeets for guidance throughout this work. Thanks for always leaving your doors open for whenever I ran into a trouble spot or had a question about my research or writing.

I wish to also appreciate everyone who contributed to the creation of the dataset used in this research work. Chukas, Tim, Kayode, Kingsley, Chi, Feyi, Zoe, Dave, Andre, Sanlap and Nivedita, special thanks to you all. I would also like to extend appreciation to everyone who willingly volunteered to partake in the annotation task but for reasons beyond their control, could not. Thank you Anda, Ofe, Udeme, Aize and Amtu.

I am particularly grateful for friends and family that have cheered me on for the past two years. All your words of encouragement and prayers lifted and kept me on during the tough times. My ICF family deserves special mention, what would the past two years have been without your encouragement and prayers!

I must express my very profound gratitude to my parents and to my siblings for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Finally, I'm grateful to the Almighty God for the strength and inner perseverance to complete this thesis.

Preface

I chose the topic of ”Predicting Semantic Labels of Text Regions in Heterogeneous Document Images” for my master thesis due to my keen interest in analyzing and mining data from textual sources. Understanding and predicting the different textual regions in documents remains a complex and challenging problem for computers. This is mainly due to the variety of ways documents are represented in the real world.

After carrying out research on the above-stated topic, my general remarks on the topic are that;

- A good segmentation of the textual regions is very important for reliable prediction of their semantic roles,
- Larger datasets with ‘high-variety’ characteristic (i.e. different layouts and formats) are needed and they are crucial to improve generalizability of methods for the task of semantic role labeling,
- End-to-end approaches provide a more complete and unified procedure for the task and can benefit from dependencies between segmentation and semantic labeling.

This master thesis report is divided into two parts. The first part consists of the research paper on my masters project, containing a concise overview of the work and the important results. The research paper was a deliverable for the assessment of my research work. The paper was also submitted to a workshop in a conference. The second part consists of a detailed appendix providing further explanation on the motivation, the models, data and error analysis, and additional experiments to provide the reader with more information and it is also an additional deliverable for assessment.

Contents

Part 1 - Research Paper

Predicting Semantic Labels of Text Regions in Heterogeneous Document Images

Somtochukwu Enendu
University of Twente
Enschede, The Netherlands
senendu5@yahoo.com

Johannes Scholtes
ZyLAB
Amsterdam, The Netherlands
Johannes.Scholtes@zylab.com

Jeroen Smeets
ZyLAB
Amsterdam, The Netherlands
Jeroen.Smeets@zylab.com

Djoerd Hiemstra
University of Twente
Enschede, The Netherlands
Djoerd.Hiemstra@ru.nl

Mariet Theune
University of Twente
Enschede, The Netherlands
m.theune@utwente.nl

Abstract

This paper describes the use of sequence labeling methods in predicting the semantic labels of extracted text regions of heterogeneous electronic documents, by utilizing features related to each semantic label. In this study, we construct a novel dataset consisting of real world documents from multiple domains. We test the performance of the methods on the dataset and offer a novel investigation into the influence of textual features on performance across multiple domains. The results of the experiments show that the Conditional Random Field method is powerful, outperforming the neural network when limited training data is available. Regarding generalizability, our experiments show that the inclusion of textual features don't guarantee performance improvements.

1 Introduction

On a daily basis, legal departments of corporations produce many electronic documents for documentation of cases, investigative reporting, internal communication etc. Whenever these corporations are involved in litigation or investigations as part of regulatory requests, the need arises to collect and review these documents and share their contents with third parties. As document data sets increase, the corporations turn to e-discovery technology to facilitate the process of collecting, reviewing and sharing. E-discovery technology helps to automatically analyze the documents by using text mining and other text-related analytics to discover relevant information. However, these text mining techniques for automatic document analysis only work optimally when the roles of different text sections in a document are known. For example, by recognizing tables, headers and footers, we can apply different extraction and analysis techniques than on normal paragraphs, and expect better results.

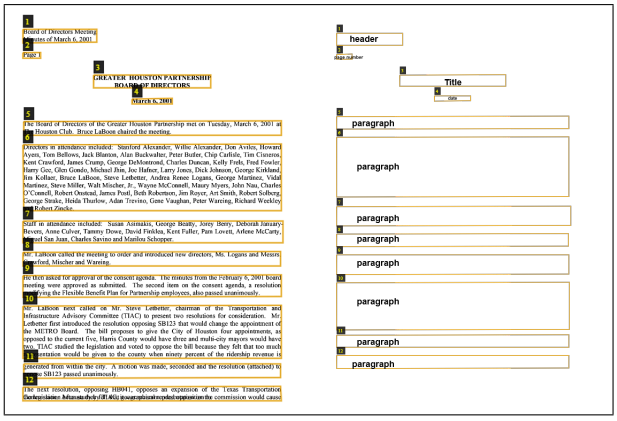


Figure 1: Example of a segmented document and its corresponding labels

For safety reasons however, electronic documents in the legal domain are mostly transformed into images (e.g. jpg, tiff) so the corporation or firm can have control of what they share with other parties. Electronic documents usually contain hidden information (information that can't be seen when the document is viewed) and these pieces of information could contain hidden details they don't want to disclose to the receiving party. On the other hand, transforming the documents to images creates another problem as it makes it more difficult to automatically identify the specific role of the document areas. Hence, to provide automatic tools to determine the function of textual regions derived from document images, we need to do document image understanding.

The primary goal in document image understand-

ing is to (1) identify regions of interest in a document image (page segmentation) and (2) recognize the role of each region (semantic structure labeling). Many related studies treat these two tasks as separate sequential tasks. However, they are also often handled as one unified task. In this work, we specifically address the second step in the understanding of document images: the task of semantic structure labeling. The goal of this task is to label a sequence of physically segmented regions in a document image with semantic labels such as header, paragraph, footer, caption, etc. (see Figure 1). We treat the task as a sequence labeling problem, which involves assigning a categorical label to each member of a sequence of observations i.e. a sequence of document segments in our scenario. Though the work of document image understanding covers various types of document images, our work focuses on electronic and digital-born documents. Typical examples of electronic documents which can be converted to images are PDF, Word, Powerpoint, E-mails, etc.

Even though extracting the semantic information from a document is a task that is easily done by a human, it is still an open and challenging problem for computers due to the inherent complexity of documents (Rangoni et al., 2012), especially when the set of documents in focus are diverse in layout and format. Similar works on semantic labeling such as (Tao et al., 2013) and (Shetty et al., 2007) are usually very specific to a document format or a set of related document types and problematic when we try to generalize to other document types. There is still a high demand for robust methods, capable of dealing with a broad spectrum of layouts found in digital-born documents (Clausner et al., 2011).

Our work addresses this gap in research by comparing sequential labeling methods for the semantic labeling task, and considering heterogeneous document images. Homogenous formats and lack of fine-grained semantic labels relevant for real world documents, limit understanding of previous document image datasets. To address these issues, we annotated a new dataset containing documents from an infamous legal case - the Enron Corporation scandal investigation. We also compare the performance of the following sequence labeling methods on the annotated dataset: (i) A feature-based Conditional Random Field (CRF) (ii) A recurrent neural network with a Bidirectional Long Short-Term Memory (LSTM) architecture.

Our methods perform fine-grained recognition on text regions and include identification of tables. Furthermore, we check the influence of textual related features on the generalizability of our methods to a different domain. Luong et al. (2010) and Yang et al. (2017) prove that the performance of methods improves when text information in a region is considered for semantic labeling. We extend this by checking its influence across a different document domain.

Our main contributions are summarized as follows:

- We compare two sequential labeling methods to address document semantic structure labeling. Unlike previous works, we consider heterogeneous document formats and identify both fine-grained semantic-based classes and tables.
- We offer a novel investigation into the influence of text-related features on the performance of our methods across a different document domain.
- We provide an evaluation dataset for the task of semantic labeling on digital-born documents.[1]

In section 3, we present our evaluation dataset. We then provide a detailed description of our system architecture in section 4. Section 5 is a breakdown of the sequence labeling methods performed for the task. We show the results of our experiments in section 6 and conclude on our work in section 7.

2 Related Work

Previous works on document image understanding (Chen and Blostein, 2007; Marinai, 2008; Kamola et al., 2015) divide the task into two parts: a physical decomposition or segmentation of document images into regions (page segmentation) and a logical/semantic understanding of these regions (semantic structure labeling). Though the focus of our work is on semantic labeling, we also present a high-level discussion on existing page segmentation techniques.

[1]The dataset will be made publicly available at a later date.

2.1 Page Segmentation

Page segmentation techniques involve identifying segments enclosing homogeneous content regions, such as text, table, figure or graphic in a document page or image. These techniques fall into three categories: *bottom-up*, *top-down* and *hybrid* approaches. Bottom-up approaches (Kise et al., 1998; Adnan and Ricky, 2011) begin by grouping pixels of interest and merging them into larger blocks or connected components, which are then clustered into words, lines or blocks of text. However, such approaches are expensive from a computational point of view. Top-down approaches (Antonacopoulos, 1998; Gatos et al., 1999) recursively segment large regions in a document into smaller sub regions. Both approaches however, are limited by their inability to successfully segment complex and irregular layout documents. Hybrid methods, such as proposed in Pavlidis and Zhou (1992) combine both top-down and bottom-up techniques. With recent advances in deep neural networks, neural based models have become state-of-the-art for segmentation. Siegel et al. (2018) utilized a neural network to extract figures and captions from scientic documents. Vo et al. (2016) proposed using a fully convolutional network (FCN) to detect lines in handwritten document images.

2.2 Semantic Structure Labeling

Our work focuses on the second aspect of document image understanding. Semantic labeling couples semantic meaning to a physical region or zone of a document after it has been segmented. Two types of approaches have been considered in the literature to handle this task: the *model-driven approach* and the *data-driven approach* (Mao et al., 2003). Early work in semantic structure labeling focused on the model driven approach. Models made up of rules, or trees, or grammars contained all the information that was used to transform a physical structure into a logical or semantic one. Rule based systems (Kim et al., 2000), though fast and human-understandable proved to be poorly flexible and unable to handle irregular cases and varying layouts.

Recent studies have considered the data-driven approach using supervised learning methods as an alternative to avoid the inflexibility and rigidity of manually built rule systems and mechanisms. These data-driven approaches make use of raw physical data to analyze the document and no knowledge or predefined rules are given. Various document image datasets have been created for this purpose including images in the document space of electronic documents, scanned documents, magazines, newspapers etc. (Todoran et al., 2005; Antonacopoulos et al., 2009) but they are usually confined to a single domain or class. Chen et al. (2007) define a document space as the set of documents that a classifier is expected to handle. The labeled training and test samples are all drawn from this document space. Our dataset includes heterogeneous formats of electronic documents such as Microsoft Office files, PDF and email files which cover multiple domains like business letters, articles, memos, forms, reports, invoices etc. that significantly vary in layout, structure and content.

Most existing supervised learning methods for semantic labeling use CRF and deep neural network approaches. Tao et al. (2013) built a CRF model as a graph structure to label fragments in a document. Shetty et al. (2007) used CRFs utilizing contextual information to automatically label extracted segments from a document. Yang et al. (2017) and Stahl et al. (2018) used visual cues and deep learning methods to analyze documents. In this study, we treat the semantic structure labeling task as a sequential labeling problem where a document image is modeled as a sequence of regions. The motivation for this is to model spatial dependencies and possible transitions between the different regions. Shetty et al. (2007) model spatial inter-dependencies between sequential segments in documents. Luong et al. (2010) also treat their semantic labeling task as an instance of the sequential labeling problem. CRFs and recurrent neural networks are popular sequential learning methods for this type of modeling. We offer a comparison of these state-of-the-art methods for semantic labeling across heterogeneous document formats in this study.

Luong et al. (2010) report in their work that adding textual information to a CRF model for semantic labeling improves its performance. We build on this work by also checking the influence of textual information on the performance of our methods across different document domains.

3 Datasets

This section describes the construction of our evaluation dataset for the task of semantic labeling which we call SemLab (SemLab coined from Se-

Dataset	SemLab	PRIMA
Document images	400	478
Document space	Office docs, PDF & Email	Magazine
Label categories	13	9

Table 1: Overview of the datasets used in this study.

mantic Labeling). The documents we used were gathered from the Enron Corpus.[2] This corpus is a large database of approximately 600,000 emails generated by 158 employees of the Enron Corporation and acquired by the Federal Energy Regulatory Commission, a United States federal agency, during its investigation after the company's collapse.

To compare the performance of the sequence labeling methods across different domains, we used the PRIMA dataset of Antonacopoulos et al. (2009). Table 1 contains an overview of both datasets.

3.1 Dataset Creation

We select documents for our dataset from the email folder of the then CEO of Enron corporation. Of all the employees in the corporation, he received the most emails. The documents comprise of sent and received email messages in the folder as well as document attachments. For attached documents, we consider four formats of documents: Word, PDF, Excel and Powerpoint documents, and ignore other file formats in the folder. This selection of different document formats meets the *variety* characteristic of an ideal dataset as described in Antonacopoulos et al. (2006) because several classes of document pages are represented. In total, we select 100 email messages and 406 unique documents from the CEO's email folder. With each document containing different pages, the full set we collected from the email folder contained 2,447 document pages.

After selection of the electronic documents, we converted them to TIFF images since document images are the focus of our work. For conversion, we used the Group 4 compression standard - a lossless method of image compression. The SemLab evaluation dataset is a random selection of 400 documents from the 2,447 document images, containing a total of 2,869 regions and their ground truth representation in CSV format (see section 3.3).

3.2 Document Semantic Labels

We attempt to identify 13 labels in a document: *paragraph*, *page header*, *caption*, *section heading*, *footer*, *page number*, *table*, *list item*, *title*, *email header*, *email body text*, *email signature* and *email footer*. Our choice of labels is specific to regions in a document that contain text. Hence we didn't consider regions in a document that are devoid of text e.g. figure, image, graphic etc.

3.3 Annotation Process

Apart from the document images part of our dataset, we created the geometric hierarchical structure of each image (in CSV format) as ground truth for the dataset. We achieved this as follows: For each region, the corresponding bounding box was given in terms of its x and y coordinates on the document image. Each region was also given a label from the set of 13 labels we defined. The bounding box coordinates were defined by page segmentation using the Tesseract OCR engine[3] while the labeling of the regions was done manually. Tesseract OCR performs an automatic full page segmentation of the document image thereby producing the bounded regions in the document. We allowed for manual correction of the regions by the annotators in case of a faulty or overlapping region. In total, 5 non-domain experts took part in annotating the sample of 400 document images independently. Each document image was annotated by 3 annotators (fixed number).

To make the manual annotation effort easier for the annotators, we split the 400 documents into 40 groups i.e. 10 documents per group, so that they had the liberty to annotate a minimum of 10 documents and a maximum of 400 documents. We set up the process by providing the annotators with a simple image editor tool to manually correct the segmentation (by specifying imprecise region boundaries using a variety of drawing modes such as using rectangles or arbitrary polygons) and label each region in a document image. We pre-loaded the labels into a toggle annotation editor to improve annotation efficiency. Hence, the annotator only needed to select the labels from a drop-down. To ensure that the annotators understood the annotation task, we provided a user guide containing com-

[2] See `en.wikipedia.org/wiki/Enron_Corpus`, accessed 2019-06-19

[3] github.com/tesseract-ocr/tesseract accessed 2019-06-09

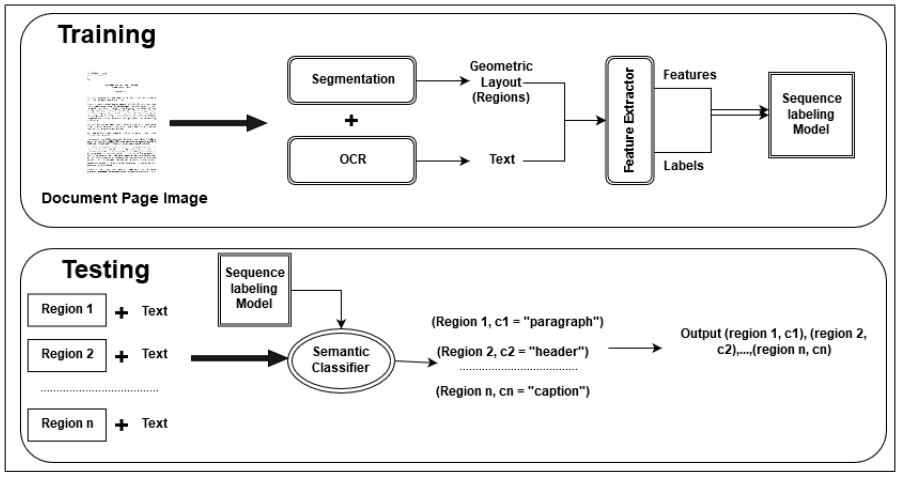


Figure 2: Implementation architecture, showing training and testing phases including the input and output for the sequence learning models

plete instructions on how to use the image editor tool and carry out the labeling of the regions.

We measured the Inter-Annotator Reliability (IAR) of agreement using the Fleiss' Kappa measure.[4] It has been shown to be more suitable to measure IAR when more than 2 annotators are involved, compared to other measures such as Cohen Kappa.[5] The Fleiss' Kappa value measured for our annotation task was 0.52. This value indicates *moderate agreement* between the annotators, going by the table given in (Landis and Koch, 1977) for interpreting Fleiss' Kappa values. It has been noted however, such as in (Sim and Wright, 2005) that the table interpretation is flawed, as the number of categories and subjects will affect the magnitude of the value. For example, the Kappa value will be higher when there are fewer categories. After annotation, the main author of this paper reviewed 8,977 annotations and resolved the disagreements between the three annotators for each document image. Disagreements were resolved by majority voting and in instances where each annotator had unique annotations, the author revisited the annotated samples and made the most logical choice of label to form the gold-standard set.

3.4 Data Augmentation

To artificially expand the size of the dataset for carrying out experiments on our deep neural network models, we employed traditional augmentation techniques as described in Perez and Wang (2017). The goal of carrying out data augmentation is to add more variation to the dataset and enable the neural network generalize better. A detailed discussion on the augmentation operations can be found in Appendix A.

4 System Architecture

Figure 2 summarizes the architecture of our semantic labeling system. During the training process, we run all input document images through the Tesseract OCR software to obtain raw text data as well as geometric layout information. The feature extractor utilizes both the layout information and raw text, when available, to produce features which go through the sequence labeling trainer together with corresponding manually labeled data, to produce the learned models. The trainer learns to assign a semantic label to the segmented regions R of a document image D. Each region $R_i \in R$ is bounded by a bounding box $B_i \in B$ that includes coherent text content and each bounding box is a set of pixels between its top left corner and bottom right corner coordinates. None of the bounding boxes overlap the other.

During testing, we want to assign a label $L_i \in$ W : $i = \{1,...,n\}$ to each region R_i. Given a sequence of regions $x = (x_1, x_2,..., x_n)$ in a document image, the task is to determine a corresponding sequence of labels $y = (y_1, y_2,..., y_n)$ for x. This can be seen as an instance of a sequence labeling problem, which attempts to assign labels to a sequence of observations. We take into account the contextual information for each of the regions in the sequence i.e. the labels of preceding or following regions are taken into account for label classification.

[4]Fleiss' Kappa works for any number of annotators giving categorical ratings, to a fixed number of items

[5]See `en.wikipedia.org/wiki/Fleiss_kappa`

5 Methods

In this section, we present the sequence labeling methods for semantic labeling of document images and the evaluation procedure.

5.1 Linear-Chain CRF (LC-CRF)

CRFs are probabilistic models used to segment and label sequential data. They are reported to be very effective for semantic structure detection (Peng and McCallum, 2004; Luong et al., 2010). An inherent merit of the CRF model to perform this task is its ability to combine two classifiers: a local classifier which assigns a label to the region based on local features and a contextual classifier to model contextual correlations between adjacent regions. Linear-chain CRFs are one well known type of CRFs which are similar to Hidden Markov Models but are reported to perform better (Peng and McCallum, 2004). They have one chain of connected labels. As CRF is a feature-based method, we implement two models with different feature sets in our work (see Table 2). We use the scikit-learn Python package, sklearn-crfsuite for implementation of our CRF models.

LC-CRF without OCR (LC-CRF$_1$): In this model, we exclude any features that can be extracted from the OCR output. That is, we consider only geometric/physical layout features to predict the label of a region in a document. The LC-CRF classifier will learn regions based on their position and location on the bounding box level of the document image. For example, it is common for *titles* to appear at the top of documents so the model may learn this observation from the extracted features.

LC-CRF with OCR (LC-CRF$_2$): By virtue of the generality and flexibility of CRF model, it is promising to achieve better performance by extending feature sets and exploring higher-level dependencies (Shetty et al., 2007). Luon et al. (2010) and Yang et al. (2017) report that by adding textual information to their models, there was an improvement in performance. We implement another LC-CRF model extending the feature set by including textual features from the OCR output. We also consider features for detecting tables. We re-use a subset of features for table detection in (Ghanmi and Abdel, 2014).

Feature set	Description
Without OCR	
Block coordinates	The location of the region bounding box within the document image (x and y coordinates)
Height	Normalized height of block
Width	Normalized width of block
Area	Normalized area of block
Aspect ratio	Width/height of block
Vertical position	Vertical position of region in the image (top, middle, bottom)
With OCR	
Digit	Binary feature indicating if the text in the region consists of digits or contains digits
Capital letters	Binary feature indicating if if the text in the region is all in capital case or contains capital letters
Nr of tokens	The number of tokens in a region block
Nr of lines	Binned number of lines in a region block (small, medium, large bins)
List item pattern	Binary feature indicating if text contains bullet items
Caption pattern	contains caption keywords (table, source, fig., figure)
Email keywords	Keywords found in different parts of an email
Has multi-white space (table feature)	Binary feature indicating if bounded region contains multiple white spaces between tokens.
% of white space (table feature)	The sum of white space lengths divided by the line length
Avg white space length (table feature)	The mean length of the white spaces within a line.

Table 2: Features used by the CRF methods.

5.2 Recurrent Neural Networks (RNNs)

RNNs are a class of nets that are used for sequence learning. They can simultaneously take a sequence of inputs and produce a sequence of outputs. They have shown great power in learning latent features, finding the most representative features from an input sequence and training the best model given these features (Akhundov et al., 2018).

Here, we use a Bidirectional-LSTM architecture for our network. We transform the feature sets of the CRF models into a 3D tensor and use this as input to the network. Two neural models (RNN$_1$ and RNN$_2$) are trained and evaluated as such implemented for the CRF models, using feature sets with and without OCR features. Hyper-parameters are set in reference to the best performing configurations in Reimers and Gurevych (2017) with minor deviations. We use the adam algorithm for gradient descent optimization (Kingma and Ba, 2015). We don't include an embedding layer and set the number of recurrent units to 100 for all 3 hidden layers. Kernel and recurrent (l2) regularizers are added to

	LC-CRF$_1$	LC-CRF$_2$	RNN$_1$	RNN$_2$
Overall Micro F_1	0.736	0.830	0.564	0.580
table	0.667	**0.885**$^{+0.22}$	0.370	0.378
paragraph	0.617	**0.754**$^{+0.14}$	0.506	0.502
page number	0.946	**0.959**	0.688	0.694
list item	0.336	**0.589**$^{+0.25}$	0.206	0.268
heading	**0.564**	0.545	0.514	0.502
page header	0.868	**0.875**	0.654	0.660
title	0.571	**0.720**$^{+0.15}$	0.432	0.412
footer	0.781	**0.875**$^{+0.09}$	0.666	0.724
caption	0.667	**0.708**	0.116	0.072
email header	0.907	**0.980**$^{+0.07}$	0.678	0.704
email body text	0.944	**0.980**	0.718	0.792
email signature	0.935	**0.974**	0.866	0.858
email footer	0.969	**0.985**	0.774	0.768

Table 3: Comparative performances among LC-CRF$_1$, LC-CRF$_2$, RNN$_1$ and RNN$_2$ models for semantic labeling. Category-specific performance given in F1. Results in bold mark the best system for each category. Superscripts indicate large improvements in F1 (> 0.05 points) between first and second ranked systems.

our first hidden layer. We add dropout with a value of 0.1 and use a batch size of 32. Furthermore, if the training loss does not decrease for 3 epochs, the learning rate is reduced by a 0.8 factor. Training is stopped if the minimum change in validation loss is less than 10^{-5} for 8 epochs or when 100 epochs are reached. We use the keras deep learning library running on top of tensorflow, for implementation of our RNN models.

5.3 Evaluation

The aim of our evaluation is to compare how sequence labeling methods perform for the task of semantic labeling of document regions and compare how their performances change with an extended feature set. We also evaluate the generalizability of our methods to a different document domain.

Let TP denote the number of correctly classified text regions (true positive); similarly, FN for false negatives, FP for false positives, and TN for true negatives. We assess category-specific results according to the F_1 measure, defined as $\frac{2xPxR}{P+R}$ where P is Precision = $\frac{TP}{TP+FP}$, and R is Recall = $\frac{TP}{TP+FN}$. Overall results are evaluated using the micro-averaged F_1 measure, the average of the results of 3 runs is reported per experiment. We split our dataset into train/test sets with a 70/30 ratio. We perform 3-fold cross validation on the train set to tune the hyper-parameters of the model.

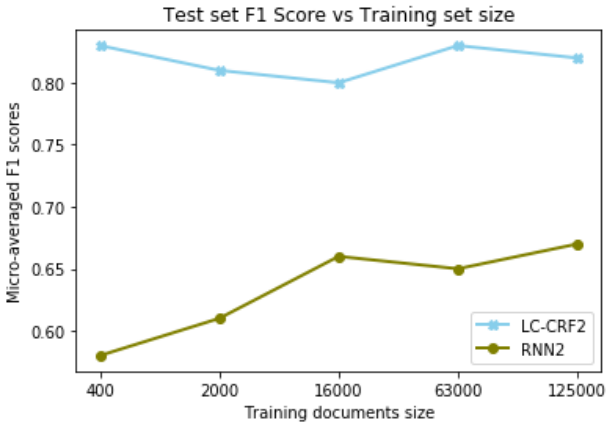


Figure 3: Comparison of LC-CRF$_2$ and RNN$_2$ with different training data set sizes. Training documents $>$400 are created from data augmentation.

6 Results

This section presents the results of our experiments.

6.1 Semantic Labeling of SemLab Dataset

Table 3 shows an overview of the results of our models comparison on the non-augmented dataset. The LC-CRF model without OCR output (LC-CRF$_1$) performs fairly well, approaching an F_1 score of 0.74. It is clear however that including features from the OCR output has a significant impact: the LC-CRF$_2$ model with OCR increases micro-averaged F_1 to 0.83. LC-CRF$_2$ greatly improves performance on the majority of categories, out of which 6 categories have F_1 improvements greater than 0.05. Though RNN$_2$ performs better than RNN$_1$, both models generally score less than the LC-CRF models on the dataset. This is at least partly because of the very small amount of training data used as input to the model. We show that for our specific task, neural networks perform slightly better with more training data as seen in Figure 3 and start to flatten out after about 40 times the original dataset size. The CRF models on the other hand seem to remain stable even with more training data. In addition, we make the following observations.

We observe that *list items*, *titles* and *headings* have the lowest scores for the best performing model. These categories usually have very similar features. For example, headings and list items are often started with numbering. Titles and headings also usually contain similar features such as having all capital letters. We also observe that list items have a very low F_1 score without OCR features. The classifier is able to only learn geometric

and positional features of this category and misclassifies a lot of its samples as paragraph since both have similar locations on a document image and more so, paragraph is the majority category. The email related categories generally have high F_1 scores irrespective of the local feature sets included. This is because of the ability of sequence labeling methods to take into account the neighborhood of items; for example, an email body text is very likely to appear after an email header and thus the classifier learns this contextual knowledge.

6.2 Comparison across different document domain

In many real life scenarios, the datasets available to train models for the semantic labeling task are mainly homogeneous document images with similar or comparable layout and format. This raises the question about how generalizable a model that has been trained on a set or related set of document images is, to different domains. We trained the sequence labeling methods on our SemLab dataset which contains documents from multiple domains and tested each model on the records from the PRIMA dataset which contains documents from the magazine domain, not represented in our own dataset. For fair comparison, we evaluated only labels applicable to both datasets i.e. intersecting labels (header, paragraph, section heading, caption, page number, footer). For this reason we excluded some features in the 'With OCR' feature set that are directly related to the excluded labels.

Table 4 provides a summary of the performance of each method on the different domains. The results show that the methods have lower performances when evaluated on unseen data of a different domain than the trained. Interestingly, both LC-CRF and RNN methods perform better when OCR information is not included for the cross domain experiment. This proves that the inclusion of textual features harms generalizability of methods across new domains for semantic labeling. This can be explained by considering the diverse ways text is written in different types of document. It is difficult for models to capture these variations from one document domain to another as some of the semantic categories are not very generalizable across different domains. Furthermore, we observe that RNN_1 is able to generalize better than the LC-CRF_1. This could be explained by the techniques specifically employed to reduce overfitting in the RNN such as the use of dropout, early stopping, l2 regularization etc. However, these techniques seem limited as the generalization performance decreases more significantly for the RNN_2 when the feature space is extended compared to the LC-CRF_2.

	Testing Domain	
Method	SemLab	PRIMA
LC-CRF_1	0.820	0.615
LC-CRF_2	0.845	0.567
RNN_1	0.716	0.693
RNN_2	0.726	0.543

Table 4: Review of the transfer learning experiment. Each method is trained on the SemLab dataset and tested on in-domain and cross-domain documents. All scores are micro-averaged F_1 scores.

7 Conclusion and Future Work

In this work we have presented a comparison between state-of-the-art sequential learning models applied to the task of semantic labeling of document regions. We constructed a novel evaluation dataset to benchmark model performance on. The experimental results reveal that the LC-CRF method is able to perform well using only a small amount of training data; a contrast to the RNN method which needs more data to see increasing performances. Though there is improvement with the RNN method with more training data, the slightness of its improvement indicates a limitation in our augmentation technique or limited variation in the original document set for the augmentation technique to benefit from. Also, including OCR information in the feature set is promising to achieve better performance as they reduce confusion between ambiguous semantic classes. Nevertheless, their inclusion might negatively affect generalization performance, as shown by our transfer learning experiments on the PRIMA domain.

Future work includes extending the document dataset in terms of size and variety to cover more document spaces, domains and classes. Models can exploit these characteristics to better generalize to new domains. Other types of augmentation techniques other than the traditional transformations listed in Appendix A could be beneficial as well to create variety in the expanded set. By virtue of neural networks' great power to learn latent features, we believe more (varying) data will also contribute to improving the performance levels of our neural method.

References

[Adnan and Ricky2011] Amin Adnan and Shiu Ricky. 2011. Page segmentation and classification utilizing bottom-up approach. *International Journal of Image and Graphics*, 01.

[Akhundov et al.2018] Adnan Akhundov, Dietrich Trautmann, and Georg Groh. 2018. Sequence labeling: A practical approach. *CoRR*, abs/1808.03926.

[Antonacopoulos et al.2006] A. Antonacopoulos, D. Karatzas, and D. Bridson. 2006. Ground truth for layout analysis performance evaluation. In *Document Analysis Systems VII*, pages 302–311, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Antonacopoulos et al.2009] A. Antonacopoulos, D. Bridson, C. Papadopoulos, and S. Pletschacher. 2009. A realistic dataset for performance evaluation of document layout analysis. In *2009 10th International Conference on Document Analysis and Recognition*, pages 296–300, July.

[Antonacopoulos1998] A Antonacopoulos. 1998. Page segmentation using the description of the background. *Computer Vision and Image Understanding*, 70(3):350–369.

[Chen and Blostein2007] Nawei Chen and Dorothea Blostein. 2007. A survey of document image classification: problem statement, classifier architecture and performance evaluation. *International Journal of Document Analysis and Recognition (IJDAR)*, 10(1):1–16.

[Clausner et al.2011] C. Clausner, S. Pletschacher, and A. Antonacopoulos. 2011. Scenario driven in-depth performance evaluation of document layout analysis methods. In *2011 International Conference on Document Analysis and Recognition*, pages 1404–1408, Sep.

[Gatos et al.1999] B. Gatos, S. L. Mantzaris, K. V. Chandrinos, A. Tsigris, and S. J. Perantonis. 1999. Integrated algorithms for newspaper page decomposition and article tracking. In *Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR '99 (Cat. No.PR00318)*, pages 559–562, Sep.

[Ghanmi and Abdel2014] Nabil Ghanmi and Belaïd Abdel. 2014. Table detection in handwritten chemistry documents using conditional random fields. In *ICFHR*, pages p. 146–151, Crete, Greece.

[Kamola et al.2015] Grzegorz Kamola, Michal Spytkowski, Mariusz Paradowski, and Urszula Markowska-Kaczmar. 2015. Image-based logical document structure recognition. *Pattern Anal. Appl.*, 18(3):651–665.

[Kim et al.2000] Jongwoo Kim, Daniel X. Le, and George R. Thoma. 2000. Automated labeling in document images. In *Document Recognition and Retrieval*.

[Kingma and Ba2015] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

[Kise et al.1998] Koichi Kise, Akinori Sato, and Motoi Iwata. 1998. Segmentation of page images using the area voronoi diagram. *Comput. Vis. Image Underst.*, 70(3):370–382.

[Landis and Koch1977] J Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33 1:159–74.

[Luong et al.2010] Minh-Thang Luong, Min-Yen Kan, and Thuy Dung Nguyen. 2010. Logical structure recovery in scholarly articles with rich document features. *Int. J. Digit. Library Syst.*, 1(4):1–23.

[Mao et al.2003] Song Mao, Azriel Rosenfeld, and Tapas Kanungo. 2003. Document structure analysis algorithms: a literature survey. In *Document Recognition and Retrieval X, Santa Clara, California, USA, January 22-23, 2003, Proceedings*, pages 197–207.

[Marinai2008] Simone Marinai, 2008. *Introduction to Document Analysis and Recognition*, pages 1–20. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Pavlidis and Zhou1992] Theo Pavlidis and Jiangying Zhou. 1992. Page segmentation and classification. *CVGIP: Graph. Models Image Process.*, 54(6):484–496.

[Peng and McCallum2004] Fuchun Peng and Andrew McCallum. 2004. Accurate information extraction from research papers using conditional random fields. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2004, Boston, Massachusetts, USA, May 2-7, 2004*, pages 329–336.

[Perez and Wang2017] Luis Perez and Jason Wang. 2017. The effectiveness of data augmentation in image classification using deep learning. *CoRR*, abs/1712.04621.

[Rangoni et al.2012] Yves Rangoni, Abdel Belaïd, and Szilárd Vajda. 2012. Labelling logical structures of document images using a dynamic perceptive neural network. *International Journal on Document Analysis and Recognition (IJDAR)*, pages 45–55.

[Reimers and Gurevych2017] Nils Reimers and Iryna Gurevych. 2017. Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. *CoRR*, abs/1707.06799.

[Shetty et al.2007] Shravya Shetty, Harish Srinivasan, Sargur Srihari, and Matthew Beal. 2007. Segmentation and labeling of documents using conditional random fields. *Proceedings of SPIE - The International Society for Optical Engineering*, 6500:6500–1.

[Siegel et al.2018] Noah Siegel, Nicholas Lourie, Russell Power, and Waleed Ammar. 2018. Extracting scientific figures with distantly supervised neural networks. *CoRR*, abs/1804.02445.

[Sim and Wright2005] Julius Sim and Chris C Wright. 2005. The kappa statistic in reliability studies: use, interpretation, and sample size requirements. *Physical therapy*, 85 3:257–68.

[Stahl et al.2018] Christopher Stahl, Steven Young, Drahomira Herrmannova, Robert Patton, and Jack Wells. 2018. Deeppdf: A deep learning approach to extracting text from pdfs. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

[Tao et al.2013] Xin Tao, Zhi Tang, and Canhui Xu. 2013. Document page structure learning for fixed-layout e-books using conditional random fields. *Proceedings of SPIE - The International Society for Optical Engineering*, 9021.

[Todoran et al.2005] Leon Todoran, Marcel Worring, and Arnold W. M. Smeulders. 2005. The uva color document dataset. *International Journal of Document Analysis and Recognition (IJDAR)*, 7(4):228–240.

[Vo and Lee2016] Q. N. Vo and G. Lee. 2016. Dense prediction for text line segmentation in handwritten document images. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3264–3268.

[Yang et al.2017] X. Yang, E. Yumer, P. Asente, M. Kraley, D. Kifer, and C. L. Giles. 2017. Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4342–4351.

A Appendix

A.1 Augmentation Techniques

To carry out experiments comparing model performance on different data set sizes, we expanded our training dataset using traditional transformation operations. While doing the augmentation, we considered techniques and operations that will not create unreal variations of each document image which may confuse the models even further. For example, since each document contains multiple regions that are sequentially arranged, it is possible that performing a geometric expansion of each bounding box region on the vertical and horizontal axis could lead to overlapping regions or regions going beyond the width or height of the document page. Hence, we carefully selected operations and set rules that will avoid these scenarios, almost completely. We artificially expanded the set using augmentation operations as follows:

1. Horizontal Shifts: Regions were shifted to the left and right of the document image based on a shifting factor. The shifting factor was set between 100 - 200 pixels. For instance, if the shifting factor was set to 150 pixels, the regions and their bounding box were shifted to the left or to the right by 150 pixels. We included rules to ensure the horizontal shifts do not violate the nature of possible real life documents by for example, ensuring regions that are already close to the left or right border of the image are not moved further beyond the border of the image.

2. Vertical Shifts: Regions were shifted upward and downward of the document image based on a shifting factor. We applied similar rules as those described in the horizontal shift

3. Shrink: This operation shrinks regions to a smaller size by a shrinking factor. The rules applied here prevent shrinking beyond a reasonable factor as this will violate certain semantic regions e.g. page number, as they are already of a minute size in height and width.

Part 2 - Appendices

Appendix A

In-Depth Overview of Research Work

In this section, we present a detailed overview of the research work done on 'predicting semantic labels of heterogeneous document images' and also include more details on the research process that has been left out in part 1 of this report.

A.1 Motivation

The research work was carried out in ZyLAB, Amsterdam. ZyLAB is a company involved in the legal tech industry, working closely with law firms, corporations, and governments to deal with e-discovery, answering regulatory requests, internal investigations, audits and handling public records requests. ZyLAB's approach to dealing with these requests is a smart fact-finding solution which utilizes machine learning and information extraction techniques to provide answers and insights to their customers and their needs. However, for ZyLAB, it is not just about providing a solution, but also how to deal with large unstructured data in various forms which is a part of the everyday e-discovery and fact finding process. Manual analysis of these data is a time consuming process that is neither beneficial to ZyLAB nor their clients. Hence, ZyLAB provides the most powerful legal search engine, data analytics and machine learning on the market. ZyLAB's solution provides support to legal service providers by assisting them to review data automatically, filter and prioritize data and, most importantly, eliminate the dull and tedious work involved in handling customer requests manually.

One of the foremost steps in the fact-finding mission performed by the ZyLAB software solution is to assign semantic roles to named entities (i.e. Named Entity Recognition) in documents. Other steps involve topic modeling, sentiment and emotion mining, relation mining etc. These steps are mostly classification approaches based on statistical models that classify text entities according to statistical properties of continuous natural language. However, these approaches only work optimally

when they are used on the type of data that they are trained on, e.g. clean and full sentences.

This creates the need for a 'pre-processing' step in which ZyLAB desires to understand the role of different text regions in a document, and thus be able to apply the right models during further processing. Such a pre-processing step allows for the choice of an optimal technique to use for specific parts of the document. Most advanced models can be run on cleaner segments with full text sentences while more robust methods can be chosen for the unstructured parts with text information.

The above motivation is the reason for the research work carried out at ZyLAB. Understanding the semantic structure or predicting the semantic labels of text regions in documents is a task that is easily done by humans (though it may be time consuming and is still prone to error due to ambiguity), however, it is still an open and challenging problem for computers due to the inherent complexity of documents, high variety of documents and noisy documents (such as OCR scans). At ZyLAB, these problems are exhibited in the documents received for the fact finding mission:

- High variety in types and formats of documents such as office documents, PDF, emails, document images etc.
- High variety of textual contents in documents (i.e. there are no strict rules applied when creating the documents. They can consist of a combination of lists, paragraphs, headings and other types of textual content.)
- Image versions of these documents with only image information available (no metadata or any information on document structure in the file).

These highlighted scenarios and problems affirm the need for understanding the semantic role or structure of different text regions in a document at ZyLAB. The focus of the research work was on document images (i.e. images of documents), which is the most common way ZyLAB's clients send the documents needed for e-discovery, investigations, among others.

A.2 Discussion and Findings

In this section, we summarize how the problem was studied. We discuss the questions we attempted to answer, the approach used and its justification, and a review of our findings from the research work.

As has been highlighted in the previous section, the main subject of the research was to assist computers to understand and assign semantic labels to text regions

in heterogeneous document images. The specific labels we focused on were *paragraph*, *page header*, *caption*, *section heading*, *footer*, *page number*, *table*, *list item*, *title*, *email header*, *email body text*, *email signature* and *email footer*. Their definitions are presented in the user guide in appendix D. Our choice of labels is specific to regions in a document that contain text. Hence we didn't consider regions in a document that are devoid of text e.g. figure, image, graphic.

We firstly defined the task as a sequence labeling problem. Sequence labeling problems involve assigning a categorical label to each member of a sequence of observations. We set up the task as a sequence labeling task because documents of various types contain a sequence of segments or regions which are read or analyzed in a particular direction depending on the language of the document and sometimes, its format. With this representation, methods can take a sequence of input instance (i.e. document segments) and learn to predict an optimal sequence of labels. This is how the problem was set up and inherently, the goal - to predict an optimal sequence of labels for the sequence of text segments in a document.

Therefore, the main question we attempted to answer was: ***"To what degree can we successfully perform reliable prediction of semantic labels of text regions in heterogeneous document images using sequence labeling methods?*** Our focus was on CRFs and deep learning methods, in particular, LSTMs. As sub-activities in answering the main research question, our main contributions in this research work were the following:

- A comparison on the performance of the aforementioned sequence labeling methods in addressing the semantic labeling problem,
- An investigation into the influence of textual-related features on the performance of these methods when tested across a different document domain,
- Creation of an evaluation dataset for the task of semantic labeling on document images.

To better understand the general background of the research topic and what is already existing about the topic in the scientific community, we reviewed various related literature in the domain of natural language processing, image processing, computer vision etc. In summary, we found that most works divide the task of document image understanding (understanding the different segments in a document image) into two stages: *page segmentation* (which has to do with segmentation of document images into homogeneous regions), and a *semantic/logical structure analysis* (which is concerned with assigning segmented physical regions with semantic labels that define the function of such regions in the document). The second step was the focus of our research work.

Page segmentation uses low-level algorithms and methods to segment document images. However, there are limitations for the existing methods. They are usually not suitable for real document image datasets, their performances are unsatisfactory for documents that have complex layouts and evaluating such algorithms is also a difficult task because many authors tend to compare their algorithms with other algorithms based on the same technique. Semantic structure labeling is another aspect of the task that needs annotated data to be able to label segmented regions of a document image. Most methods used here rely on classification models to classify regions of a document image. Before deep learning models became predominant both in applications and research, which was not a very long time ago, Hidden Markov Model (HMM) and Conditional Random Fields (CRF) were the best models for this problem. Many recent literature however, now point to deep learning models as very promising to achieve best performance for the semantic labeling task.

A.2.1 Sequence Labeling and Chosen Methods for Prediction

Sequence labeling is a task that involves assigning categorical values to each member of a sequence of observed values. Depending on the nature of the problem, the output label for each member of the sequence can be any item from a set of 2 or more labels. Thus, sequence labeling can be considered as a type of binary or multi-class classication problem. The sequence labeling problem is most commonly assigned for tasks such as Named Entity Recognition and parts-of-speech tagging in which the labels of neighboring members in the sequence are taken into consideration for prediction.

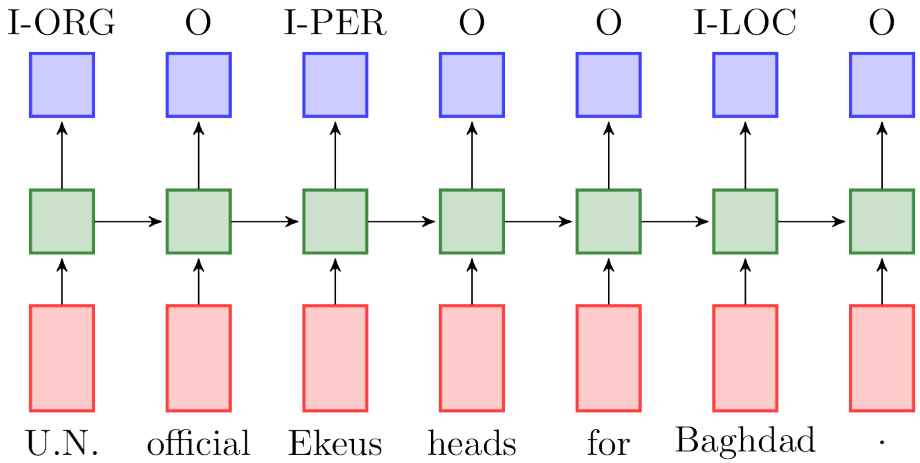


Figure A.1: *Named Entity Recognition as a Sequence Labeling Problem*[1]

[1]Source: www.depends-on-the-definition.com/guide-sequence-tagging-neural-networks-python

The figure above shows an example of the sequence labeling formalism. A sequence of words are assigned a sequence of labels with labels of neighboring members taken into consideration in assigning a label. Examples of traditional machine learning algorithms developed for sequence labeling are Hidden Markov Model, Maximum Entropy Markov Model and Conditional Random Field. These algorithms make a Markov assumption, which means the choice of label of one member of the sequence is directly dependent on the label of the previous member. However, it has been seen that Conditional Random Field performs the best in sequence labeling task over these other algorithms [1]. CRFs provide several advantages over HMM including the ability to relax strong independence assumptions made in those models. CRFs also avoid the fundamental limitation of MEMM, that is, bias towards states with few successor states, known as label bias problem. Deep Learning models such as Long Short Term Memory (LSTMs) are also very promising for the task as they can generalize better than previously mentioned models, on unseen data. Thus, we chose these models (CRF and LSTM) to answer the stated research questions.

Conditional Random Field

CRFs directly model the conditional probability p(y|x) i.e. the probability of a label y given x (where x is a sequence of observed feature vectors, x = (x_1,...,x_n) and y is a sequence of class labels, y = (y_1,...,y_n)). It was developed by John Lafferty, Andrew McCallum and Fernando Pereira in the year 2001, as a framework to build probabilistic models to segment and label sequence data [2]. CRFs in their formalism take into account the context of the input x, that is to say, the labels of surrounding or neighbouring inputs.

CRF is expressed using the following simplified form:

$$p(y|x,\lambda) = \frac{1}{Z(x)} \exp(\sum_j \lambda_j f_j(y,x)) \tag{A.1}$$

The above equation represents the probability of a particular sequence *y* given an observation sequence *x*. $f_j(y,x)$ represents either a state function $s(y_i,x,i)$ or a transition function $t(y_{i-1},y_i,x,i)$. λ_j are the feature weights to be set while training and $Z(x)$ is a factor normalizing the sum of probabilities to 1. The state and transition functions are represented as binary features and this is given in the form as shown below using our problem context:

$$s(y_{i,x,i}) = \left\{ \begin{array}{ll} 1, & \text{if text of } x_i \text{ has uppercase letters and } y_i = \text{"heading"} \\ 0, & \text{otherwise} \end{array} \right\}$$

$$t(y_{i-1,x,i}) = \left\{ \begin{array}{ll} 1, & \text{if } y_{i-1}\text{= 'heading' and } y_i\text{= 'paragraph'} \\ 0, & \text{otherwise} \end{array} \right\}$$

These functions (state and transition) are called feature functions in the CRF formalism. During training, each feature function f_j is assigned a weight λ_j, and the sum of the weighted features give a score s of the observed item. These scores are further transformed into probabilities by exponentiation and normalizing the scores. One way to learn the feature weights is by using gradient descent, where the weights are randomly initialized but moved in the direction of the gradient to minimize the error (difference between predicted output and actual output). If the weight λ_j associated with a feature function is large and positive, then the feature is essentially aligning itself or giving preference to a particular label. For example, say we have a state binary feature function $s(y_i, x, i)$ where $s(y_i, x, i)$ = 1 if y_i = Heading and the x observation contains all capital letters, and 0 otherwise. If the weight λ_j associated with this feature is large and positive, then this feature is essentially saying that it prefers labelings where the observation contains all capital letters, gets labeled as Heading.

Apart from CRFs difficulty to generalize to unseen data, they have other limitations which are specific to Markov-Chain based models (of which they are a member). They have difficulty handling longer sequential dependency, due to their Markovian assumptions, e.g., dependencies of the input sequence longer than 3 steps or larger are often ignored. For our experiments, we used Linear chain CRFs. They are one common type of CRFs and are similar to HMMs. CRFs are generally undirected graph structures but linear-chain CRFs are sequence structures conditioned on previous transitions with a linear structure i.e. they have only one connected chain of labels where their parameters are tied across time.

Bi-directional Long Short Term Memory (LSTM)

LSTMs are a version of the recurrent neural network (RNN). To get a better understanding of LSTMs, we will briefly discuss RNNs. Recurrent (Feedback) neural networks add an interesting twist to basic neural networks as they are graphs in which loops occur because of feedback connections. This is unlike feed-forward neural networks which are graphs with no loop and their neurons have only unidirectional connections between them. RNN models are also designed to capture local dependencies and find longer patterns unlike the Markov-Chain models previously described. They can be applied to connect long-term dependent contextual information to a current task since their feedback loops allow information to persist. However, in practice they suffer two problems called the exploding gradient

and vanishing-gradient problems i.e. error signals flowing backwards in time tend to either blow up or vanish.

The way RNNs are trained is that first, both the inputs and outputs are provided to the network. The network processes the inputs and compares the actual outputs with the predicted outputs. The error between the actual and predicted outputs is then pushed backed through the network to adjust the weights, which control the network. Error gradient is the direction and magnitude calculated in training RNNs, which is used to update network weights in the right direction and by the right amount. This creates the exploding gradient problem which occurs when gradients having values larger than 1 are repeatedly multiplied through the layers of the network. On the other hand, vanishing gradients occur when the gradients are so small values that they no longer update the weights, hence the weights do not change. This prevents the network from learning long-term dependencies between the inputs.

LSTMs were then introduced by Sepp Hochreiter and Jurgen Schmidhuber in 1997 to deal with these error problems faced by simple RNNs [3]. LSTMs can learn to bridge time intervals in excess of 1000 steps in input sequences without the loss of short time lag capabilities. This is achieved by enforcing constant flow of error via the hidden states of the network. For our prediction task, we used a bi-directional architecture of the LSTM network. Bi-directional LSTM differs from a normal LSTM in that it offers a forward and backward looking network as compared to LSTM which only uses contextual information from the past. Bi-directional LSTM will run inputs in two ways, one from past to future and the other from future to past and thus using the two hidden states combined, the network is able in any point at time to preserve information from both past and future.

For example, let's say an LSTM model tries to predict the word *'swimming'* in the sentence, **"The girls went swimming and they swam for 3 hours"**. On a high level, what a unidirectional LSTM will see is *'The girls went....'* and try to predict the next word. On the other hand, a bi-directional architecture will also see information further down the road, So, both *'The girls went....'* and *'and they swam for 3 hours'* to predict the word 'swimming'.

The two models described above were used for the task of predicting the semantic labels of text regions in document images. CRFs make use of a hand-crafted, high-quality feature set to learn weights to be set while training. However deep learning models like LSTM have shown great power to learn latent features. The training process is a joined learning of finding the most representative features and training the best model given these features. Many deep learning techniques deployed for sequence labeling in natural language processing, deal with labeling word se-

quences e.g. NER or POS-tagging. However, since our sequence labeling task is a different problem involving tagging document region sequences, we transformed the CRF features as input to the Bi-LSTM model. However, another viable representation could be to derive image-based or pixel-wise features using a pre-trained image feature extraction neural network. We opted for using the CRF features as input to our neural network because of the limited training data available for the task.

A.2.2 Approach Summary

In this section we provide a summary of the approach followed for the complete task of semantic labeling.

As one of the main contributions of this research work, an important task was to create an evaluation dataset that can be used to evaluate performance of the sequence labeling methods for the task of semantic labeling of text regions (see Appendix B for a detailed analysis of the dataset). There was no readily available dataset that fits the problem at ZyLAB and hence the need to create the dataset. We selected 400 document images - a combination of office documents, PDFs and email documents from the Enron Corpus.[2] As a first step, we carried out page segmentation of each document image. Page segmentation as discussed in several literature such as [4] and [5], is the first step in the task of document image understanding. For our work, we used the state-of-the-art Tesseract 4.0 OCR engine[3] to segment the document images. In version 4 of its engine, Tesseract also implemented a Long Short Term Memory (LSTM) based recognition engine to perform page segmentation as well as Optical Character Recognition (OCR). One reason for the choice of this engine is its seemingly popular use for evaluation comparison of other page segmentation methods used in international document analysis and recognition competitions [6], [7]. It is also an open source system and hence doesn't require commercial licensing which is beneficial for ZyLAB. We also used Tesseract to perform OCR recognition which produces the text content of each segment in a document image. OCRing of the document image was important for our work since one of our contributions was to analyze the influence of text-related features on the generalizability of our methods. When considering the results presented later in this chapter, it is worth mentioning that predictions of some of the text segments depends on this OCR tool.

After segmentation, 3 annotators manually annotated the document images to produce ground-truth information (the research paper contains detailed information

[2] See en.wikipedia.org/wiki/Enron_Corpus, accessed 2019-06-19
[3] github.com/tesseract-ocr/tesseract, accessed 2019-06-09

on the annotation process and Appendix B provides an analysis on the annotation task). Before extracting the features used in learning the models (see research paper), we had to map/match the OCR output (all tokens in document image with its bounding box coordinates) with the segmentation output (bounding box coordinates of all text regions in document image) into a ground-truth file.

We modeled the matching problem as such: We represent a set of polygon coordinate points in a text region as:

$$S = \{P_1(x, y), P_2(x, y), P_3(x, y)..P_n(x, y)\}$$

and the (rectangular) bounding box of each OCR token as:

$$T = (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$$

To match/map each token to its polygon text region, the following steps are taken:

1. For each ground truth file, we transform relevant polygon text blocks into rectangular blocks by finding the min(x,y) and max(x,y) in S.
2. For each token in OCR output, we check if bounding box coordinate points are within a rectangular block.
3. Assign token to block if it meets condition

A.2.3 Summary of Results

The experiments we carried out were with a view to answer the main research question as well as questions or problems introduced by the proposed contributions. After creating the evaluation dataset, we evaluated our sequence labeling models on the dataset to measure their performances for our task. Actually, only a part of the dataset was used for evaluation (30%) and the other split was used for training the model (70%). The evaluation procedure is described in the research paper in Part 1 of this report. We created 2 variations for each model since another goal was to investigate the influence of textual features on the generalizability performance of the models. The variations included models without the OCR related features in the feature set(LC-CRF$_1$ and RNN$_1$) and another set of models with OCR related features (LC-CRF$_2$ and RNN$_2$). The models comparison revealed the results shown in table A.1.

	LC-CRF$_1$	LC-CRF$_2$	RNN$_1$	RNN$_2$
Overall Micro F_1	0.736	0.830	0.564	0.580
table	0.667	**0.885**$^{+0.22}$	0.370	0.378
paragraph	0.617	**0.754**$^{+0.14}$	0.506	0.502
page number	0.946	**0.959**	0.688	0.694
list item	0.336	**0.589**$^{+0.25}$	0.206	0.268
heading	**0.564**	0.545	0.514	0.502
page header	0.868	**0.875**	0.654	0.660
title	0.571	**0.720**$^{+0.15}$	0.432	0.412
footer	0.781	**0.875**$^{+0.09}$	0.666	0.724
caption	0.667	**0.708**	0.116	0.072
email header	0.907	**0.980**$^{+0.07}$	0.678	0.704
email body text	0.944	**0.980**	0.718	0.792
email signature	0.935	**0.974**	0.866	0.858
email footer	0.969	**0.985**	0.774	0.768

Table A.1: *Comparative performances among LC-CRF$_1$, LC-CRF$_2$, RNN$_1$ and RNN$_2$ models for semantic labeling. Category-specific performance given in F1. Results in bold mark the best system for each category. Superscripts indicate large improvements in F1 ($>$ 0.05 points) between first and second ranked systems.*

We also carried out a transfer learning experiment to test how generalizable our methods are on a different document domain. Each method was trained on our evaluation dataset and tested on in-domain and cross-domain documents. The results are shown below:

	Testing Domain	
Method	In-domain	Cross-domain
LC-CRF$_1$	0.820	0.615
LC-CRF$_2$	0.845	0.567
RNN$_1$	0.716	0.693
RNN$_2$	0.726	0.543

Table A.2: *Review of the transfer learning experiment. Each method is trained on the evaluation dataset and tested on in-domain and cross-domain documents. All scores are micro-averaged F_1 scores.*

More concise analysis and conclusions of the results are presented in Part 1 of this report (research paper). Appendix B also contains an error analysis where we examine some instances in which our model produces wrong predictions. However, we present a summary of our findings:

On Label Ambiguity

Reviewing the evaluation performance of some individual labels, it is evident that there are some of the labels that generally have unsatisfactory performance scores such as caption, heading and list item, across all the compared models. We present a detailed analysis of the ambiguity of these labels in appendix B.

CRF (LC-CRF) vs RNN (Bi-LSTM)

The results reveal that the LC-CRF method is able to perform well using only a small amount of training data. On the other hand, we need more data to see increasing performances for the RNN method (see figure 3 in research paper). Increasing the amount of data for the CRF model has little or no effect. Generally, it has been surveyed such as in [8], that for traditional machine learning algorithms (linear or logistic regressions, SMVs, Random Forests, CRF and so on), performance increases as we train the models with more data, up to a certain point, where performance stops going up as we feed the model with more data. When this point is reached, the model's performance can not be improved any more by feeding more data. It is a scenario in which the model does not know what to do with the additional data.

On the other hand, this is not the case with deep neural networks. Performance, almost always increases with data (**if the data is of good quality**), and it does so at a faster pace depending on the size of the network.

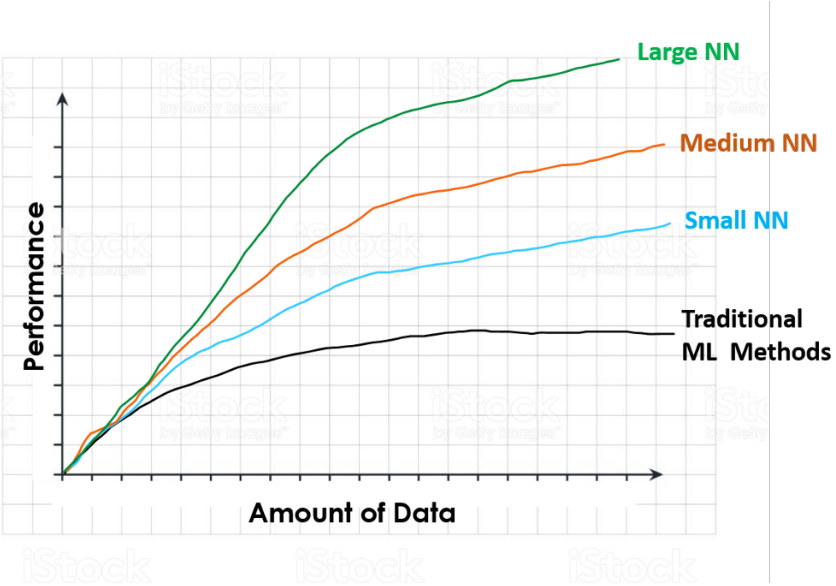


Figure A.2: *Figure showing the evolution of the performance of different algorithms as we feed them more training data* [5]

Therefore, to get the best possible performance, we would need to be somewhere on the green line (large neural network) and towards the right of the X axis (high amount of data) as shown in figure A.2.

Though we saw increase in the performance of the recurrent neural network in our experiments, it was not significant enough and still did not match the performance of the LC-CRF model. This indicates either a limitation in the additional data created using traditional augmentation techniques or a limited variation in the original document dataset for the augmentation technique to benefit from.

We also note the difficulty to train deep neural networks. There are a lot of optimization issues to deal with before getting the right configuration of the network, unlike CRFs which are easy and straightforward to train. Also, the weights to be trained in deep learning networks are much more than the weights trained for the CRF model, buttressing the point about their simplicity compared to the LSTM. CRF learns weights for each feature and hence the number of weights is proportional to the number of input features given to the model. However, due to back-propagation and feedback connections, the LSTM model learns more than 100 times more weights. We investigated our LSTM model to contain 507,734 trainable weights.

[5] towardsdatascience.com/deep-learning-for-nlp-anns-rnns-and-lstms-explained-95866c1db2e4

Inclusion of Textual Features vs Non-Incluson of Textual Features

We verified that textual-related features have a positive influence in seeing improving performances when evaluated on in-domain documents as they reduce confusion between ambiguous classes. However, their inclusion might negatively affect generalization performance as shown in the results of the transfer learning experiments on cross domain documents. This may be due to the model over-fitting on certain textual features of the training documents while not capturing the variations or diverse ways text is written or represented in documents of another domain.

A.2.4 Impact of the Research Work on ZyLAB and Scientific Community

The research work that has been carried out is a potentially viable addition to the smart fact finding solution or environment of the ZyLAB software. The work is a very important pre-processing step that will enable ZyLAB to understand the structure of documents that they deal with before carrying out information extraction steps to extract relevant information. This pre-processing step helps to apply more specific techniques for fact finding since document structure is known and thus improve the quality or performance of the said techniques. We have created an approach for document image analysis that spans across the 2 steps of document image understanding. The first step, using a state-of-the-art segmentation and OCR tool and the second step, making use of supervised learning/classification techniques to label various segments of a document but beyond that, evaluate their performances.

The evaluation dataset we created, which contains ground-truth information, also yields a lot of benefit to on-going research work at ZyLAB as it can be used to evaluate the performance of other models or techniques that may be applied for the task of semantic structure labeling. This evaluation dataset is also useful for the scientific community as it provides a starting point for researchers who are looking to evaluate models on the type of documents present in the dataset. Researchers on legal tech matters will find this very useful as the documents are mainly documents typically used in the legal domain.

We also recognize 13 semantic labels, which covers a large number of labels and is useful for comparison purposes for other research works. A known problem in the evaluation of semantic structure labeling is the lack of similar labels to compare different model performances. However, our list of labels includes categories that are present in various document types and we also take care of email related categories.

To the best of our knowledge, we are the first to represent the task of *seman-*

tic/logical structure labeling of text regions in a document as a sequence labeling problem using Bi-directional LSTMs. Many other works that use LSTMs for sequence labeling do so for tasks such as POS tagging, NER etc. and thus use or combine character and word embeddings/representations which are then fed into the LSTM to model context information of each word. In our work, we model features of each text region and feed into the LSTM model to learn context information. Our approach provides a signal or direction of the usefulness or usability of deep learning for such representation.

A.3 Limitations

The research work has some limitations which we discuss next. The major limitation is that there was no readily available dataset to evaluate our models on. Hence, we needed to manually label/annotate a set of documents for training and evaluation purposes. The annotation scheme developed for this purpose was time consuming and extensive, hence due to these constraints we were only able to annotate 400 documents. This rather limited number of documents is an acknowledged challenge in other research works to create models that are highly generalizable across document domains, classes, types and formats. Another challenge is the ability to generate large amount of data which is important especially where data is not available for learning the deep neural networks. This creates an extra task to find techniques for automatic data generation. Though we chose traditional augmentation techniques in this work, there are other probable techniques that may be used to create richer (higher quality) data.

Some of the document images in our dataset were scanned documents and documents with black backgrounds. These types of documents are problematic for OCR tools (they add noise) and thus may need further image processing operations before being used. We didn't consider this image processing step in our research work as it wasn't part of the scope. High resolution of the document images is also necessary for effective *OCRing*, at least 300 DPI. The OCR tool (Tesseract) is limited by its recognition quality. Its recognition quality decreases when lower resolutions are used. However, some of our document images fall below this 300 DPI threshold. It is also important to note the impact of the page segmentation and OCR tool used for our work. As earlier described, the tool makes use of a deep learning based recognition engine. So a lot of its accuracy on segmentation, is dependent on the data images the engine was trained on.

Another limitation of our work has to do with the lack of full evaluation of the document image understanding task. Since we created a fully annotated dataset with

region coordinates, giving the annotators freedom to adjust the regions' bounding boxes, it is possible to carry out an evaluation of the entire task. However, implementing a segmentation algorithm/model was beyond the focus of this research and hence, the absence of segmentation evaluation. We only evaluated the second step, which is the semantic structure labeling part.

A.4 Recommendations

There are several ways our research work can be extended. Some of these are described in this section.

In our work, we left out using image features in training our models. Generally, these features take more time to train. However, including image-related features per region block, in the feature set, is capable of improving the prediction performance. Another possible experiment to carry out is to compare the performance of the sequential models used in this research against non-sequential models (that look only to the document region itself) to attest to the benefits of sequential learning for this task.

For the task of predicting the labels of text regions in a document, it is important to further consider end-to-end segmentation approaches. Works such as [9] consider the task as a pixel-wise segmentation task, and propose a unified model that classifies pixels based on their visual appearance and underlying text content. Their work is a generalized page segmentation model that additionally performs fine-grained recognition on text regions i.e. their model handles the two steps of document image understanding.

A consideration which could be helpful to improve performance of the deep neural approach is to combine different techniques to achieve an optimal model. In [10], an architecture is presented combining Convolutional Neural Network (CNN), Bi-LSTM and CRF to perform linguistic sequence labeling which yields state-of-the-art performance. Other research works have also proposed more advanced RNN models such as attention-based models, Pixel-RNN, Convolution LSTM (ConvLSTM) which address some limitations in simpler LSTM models [8]. An exploration of these type of networks may yield results on the direction to go for deep learning on our specific task. Further improvement of the feature set used to learn the models could also enhance their performances. Though we already include an effective set of features, others such as the horizontal positioning of region tokens, indentation level, token distances (particularly for tables) may be effective in improving model performances.

For the CRF model, a 2-step or hierarchical method to label the regions could be

helpful to improve performance. It is likely that the label with the second highest conditional probability, is actually the correct label category. Passing the labels with the highest and second highest probability to another classifier with a specialized feature set or heurestic rules could limit prediction errors especially for the ambiguous label categories.

Furthermore, making cross-domain comparisons with documents having very diverse layouts is not very beneficial. Though we aim to achieve highly generalizable models, comparing for example, documents with a Manhattan layout (fixed composition with clear horizontal and vertical blocks) against documents with Non-Manhattan layout (very loose, non-rectangular blocks) will definitely damper performance values. A further consideration will be to carry out transfer learning experiments with more similar documents in layout (single column documents), as compared to the cross-domain dataset used for our transfer learning experiment.

As earlier stated, other advanced techniques for automatic data generation may also be considered. In [9], a synthetic document generation method is presented. The authors created a synthetic data engine, capable of generating large-scale, pixel-wise annotated documents. Other advanced augmentation techniques to scale up data from a small set are also discussed in [11].

Appendix B

Overall Analysis

In this appendix, a more detailed analysis of the dataset used for evaluation is carried out. First, an examination is done on the (dis)agreement between the volunteers that annotated the dataset, to be able to indicate the most agreed upon or disagreed upon semantic labels. Further analysis is done on the errors that our labeling models make, to be able to understand the ambiguities and complexities between the semantic labels. Finally, we propose disambiguated labels for new evaluation.

B.1 Data Analysis

The evaluation dataset was annotated by 3 non-domain annotators, independently and in parallel. To ensure the annotators understood the task and the semantic labels that were to be assigned, a user guide was provided (see appendix D). After the annotation task, we measured the Inter-Annotator Reliability (IAR) of agreement between the annotators using the Fleiss' Kappa measure. The Fleiss' Kappa value measured was 0.52. We analyze the level of (dis)agreement between the annotators and further provide a description of which labels were the most/least agreed upon.

Table B.1 shows the agreement level of the annotators for each label. *Perfect agreement* represents a situation when all 3 annotators annotated a particular text region as the same label. *Partial agreement* indicates when 2 out of 3 annotators agreed on the same label while *Isolated* shows when only 1 of the annotators chose a particular label. We also report the percentage of *Perfect agreement* to indicate the most and least agreed upon labels. From the report table B.1, one can see that among the 3 annotators, **page number** was the most agreed upon followed closely by the **paragraph** label. The least agreed upon label was **caption**, in that no occurrence of the caption label in the dataset had a perfect agreement.

From this analysis, we can make conclusions based on the results. One conclusion is that even for humans, it is difficult to reach a consensus on the labels to

Labels/Agreement per region	3 (Perfect agreement)	2 (Partial agreement)	1 (Isolated)	% of Perfect agreement
Paragraph	478	86	204	62.2%
List item	178	126	107	43.3%
Heading	27	68	102	13.7%
Page header	70	65	191	21.4%
Caption	0	33	121	0%
Footer	67	63	57	35.8%
Page number	88	37	16	62.4%
Table	123	42	39	60.3%
Title	13	51	112	7.3%
Email header	89	137	134	24.7%
Email body text	121	211	234	21.3%
Email signature	13	84	96	6.7%
Email footer	15	75	104	7.7%

Table B.1: *Report on the level of annotator agreement per label. Each row represents a label and each column represents the level of agreement between the annotators.*

assign to a particular text region due to the ambiguity of some of them. An example is the caption label which received a 0% perfect agreement value. Analyzing some of the other labels with less ambiguity and which are easier to assign, we can also postulate that the user guide in which instructions on carrying out the annotation task were presented, may not have been very clear for the annotators who took part in the task.

B.2 Error Analysis

We also analyzed some predictions made by one of our sequence labeling models, the LC-CRF$_2$ model. The confusion matrix of figure B.1 shows an overview of the true and false positives, and true and false negatives for each semantic label class.

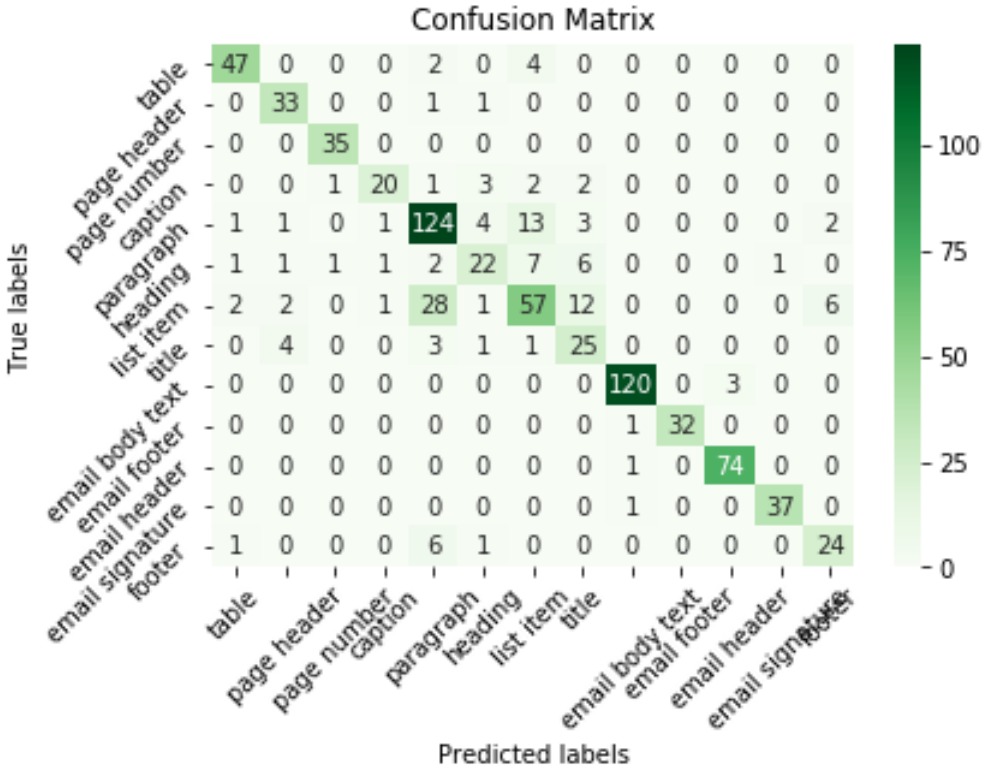


Figure B.1: *Confusion Matrix of test set LC-CRF$_2$ model*

To perform the prediction of labels, the LC-CRF$_2$ model takes into account the current observation and its features, as well as contextual information (information from neighborhood regions). In table B.2, we present the top 20 likely contextual transitions that the model learns (i.e. how likely it is for a label to be seen after a preceding one), as well as their weights. At the end of this appendix (appendix B), in table B.3, we also present a breakdown of the top two features for each label.[1] A brief explanation on how these feature weights are calculated is given in A.2.1

[1] Some additional features have been included apart from the ones presented in table 2 of the research paper.

Top Likely Transitions	Weights
email header $->$ email body text	6.81
email header $->$ email header	4.05
email body text $->$ email signature	3.33
page number $->$ title	3.24
page header $->$ title	2.77
page header $->$ page header	2.60
table $->$ table	2.30
page number $->$ footer	2.23
caption $->$ table	2.17
email body text $->$ email body text	1.94
list item $->$ list item	1.91
footer $->$ footer	1.79
page header $->$ page number	1.60
table $->$ caption	1.57
footer $->$ page number	1.51
paragraph $->$ paragraph	1.22
caption $->$ caption	1.19
heading $->$ paragraph	1.16
title $->$ heading	1.08
table $->$ heading	0.78

Table B.2: *Top likely label transitions and their weights*

For five labels, we highlight an example of predictions made by the model that goes wrong, and discuss their interpretations.

B.2.1 Footer

As shown in the confusion matrix of figure B.1, the model mis-classifies the footer label as paragraph 6 times out of the total number of footer instances. Figure B.2 is an example of one of such mis-classifications.

In this example, the model mis-classifies the last region in the document (i.e. footer) as a paragraph. The region contains the following text: "*This position was taken as a voice vote and no written resolution was developed*". We put forward two reasons why this mis-classification may have occurred:

- Preceding Paragraph Context: In this example, 8 preceding regions before the footer region are paragraphs. Since the LC-CRF model takes the neighborhood context into account, it learns (from training samples similar to this

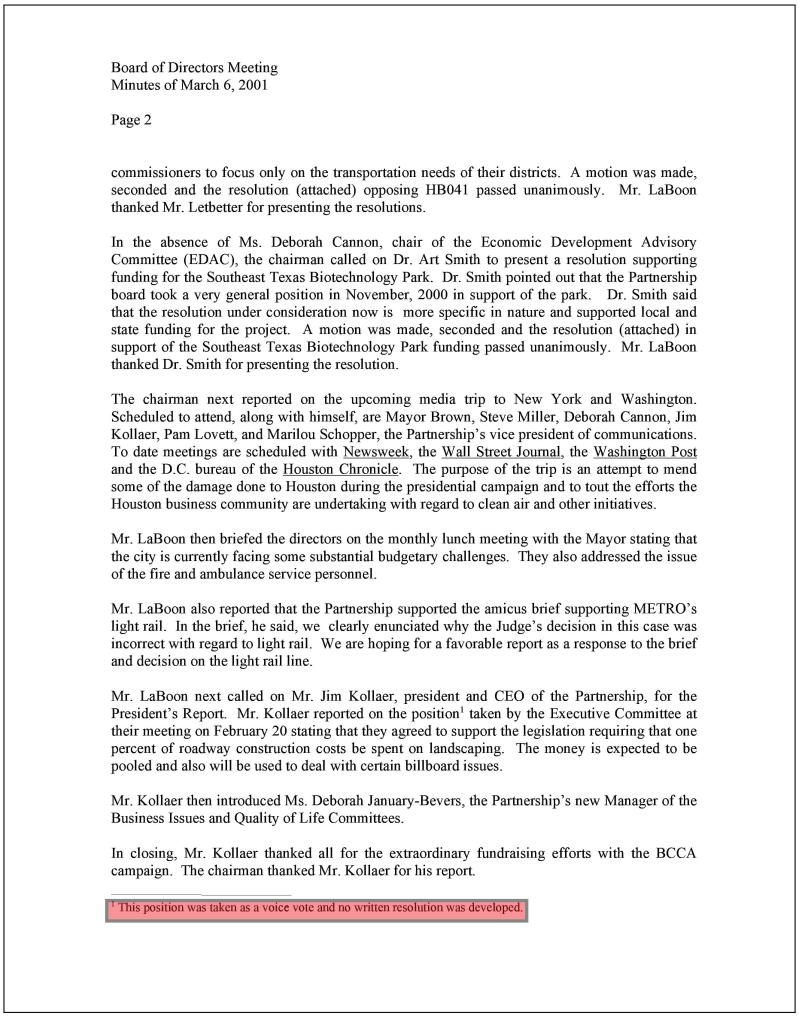


Figure B.2: *Figure showing mis-classified region. The highlighted region was labeled as a footer in the ground truth dataset. However, it was predicted as a paragraph by the model.*

document) that it is very likely a paragraph will follow another paragraph and this is the case in this document with 8 consecutive paragraphs.

- Absence of Page Number in Neighborhood: As seen in the transition likelihood of table B.2, it is common for a page number to appear in the neighborhood of a footer (before or after). In this example, this is not the case.

A possible solution to eliminate or reduce this prediction error is to include specific local features apart from the ones the model already benefits from. An example of such local feature which is not part of our feature set is, the presence of a dividing line above the footer text that separates it from the main text in a document. It is very common for footers to follow such a line in various types of documents.

B.2.2 Caption

We define captions as pieces of text that surround, or are in the neighborhood of tables, figures, charts etc. and define these items in a document page. Captions are mostly mis-classified as headings by the model. Below is an example of one of such mis-classifications:

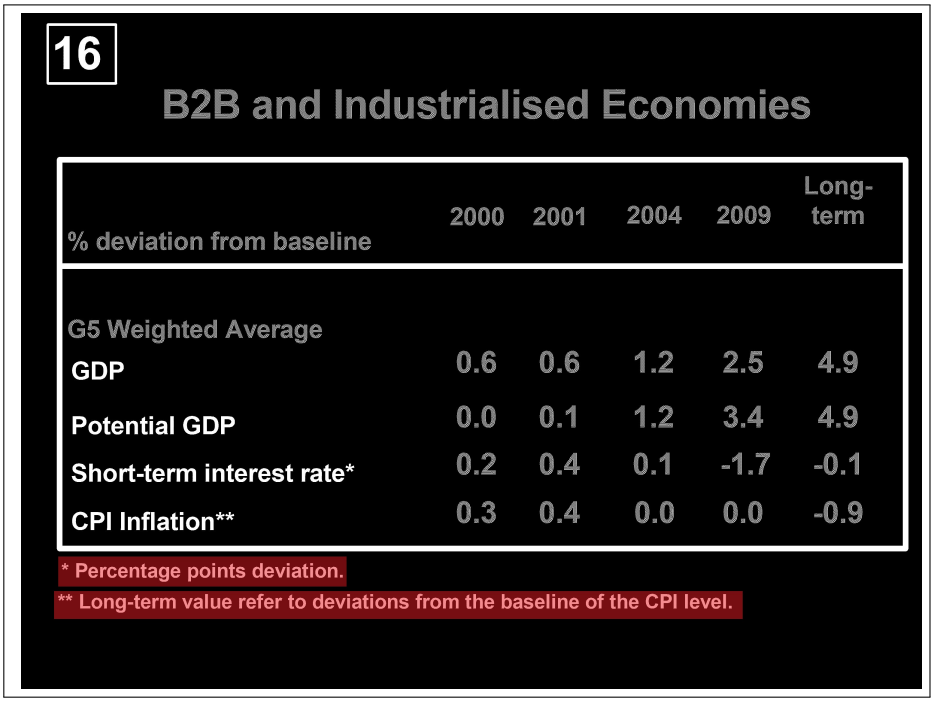


Figure B.3: *Figure showing mis-classified regions. The highlighted regions were labeled as captions in the ground truth dataset. However, they were predicted as headings by the model.*

The highlighted regions in the document page were annotated as captions in the evaluation dataset. However, the model predicts them as headings. One would assume that the transitional probability of 1.19 between $table -> caption$ is sufficient for the model to correctly predict this specific example as caption, but that isn't the case.

The local feature used by the model to learn captions consists of keywords such as table, source, figure, fig. etc. However, the text content in the first highlighted region: *" * Percentage points deviation."* doesn't contain any of those keywords. In fact, it begins with an asterisk and has a similar sentence-length as a heading may have. These type of errors are again caused by ambiguities in the text and the features used to recognize them.

A possible solution to this prediction error is to include asterisks as a 'caption' keyword in the feature set. However, this doesn't really solve the problem as other labels such as list items and headings also begin with asterisks. An alternative

solution will be to introduce new labels that are more representative of such text regions as the ones highlighted in this example (see B.3) or generally, introduce more training data containing caption labels.

B.2.3 List Item

List items are misclassified as paragraphs 28 times by the model. Below is an example of one of such mis-classifications.

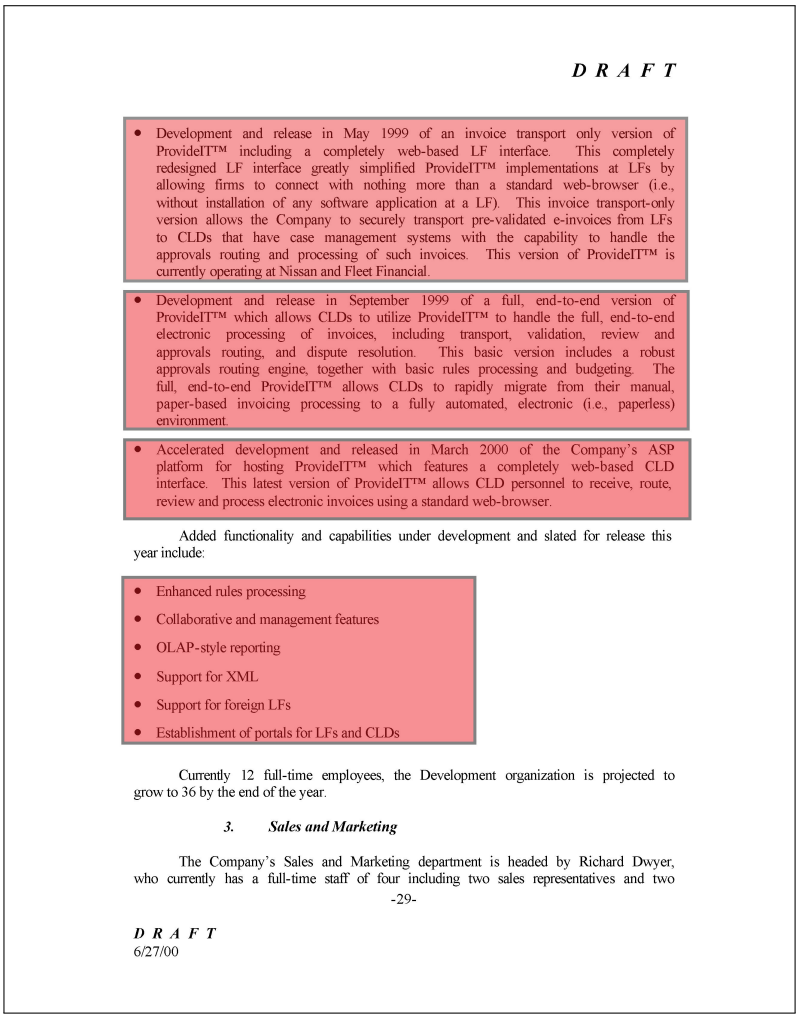


Figure B.4: *Figure showing mis-classified regions. The highlighted regions were labeled as list items in the ground truth dataset. However, they were predicted as paragraphs by the model.*

In this example, all the highlighted list item regions in the document are predicted as paragraphs. After inspecting the OCR output for this document, it was discovered that the bullet items are represented by a non-ASCII character, not captured by the list item pattern feature. In that case, the model predicts those regions as

paragraphs which is furthermore strengthened by the transitional likelihood between paragraph to paragraph.

A possible solution to reduce this prediction error is to expand the list item pattern feature to include the different characters represented as bullet points by the OCR engine. Another possible solution is to add an indentation feature, which checks the indentation level of the region in relation to the rest of the document regions.

B.2.4 Title

A title is a name given to a text region that describes the entire content of a document page. It usually occurs at the top center of a document. It differs from a page header in that it actually describes the document content. Titles are mostly mis-classified as page headers by the model. As shown in figure B.1, the mis-classification of titles as page headers occurs 4 times. Below is an example:

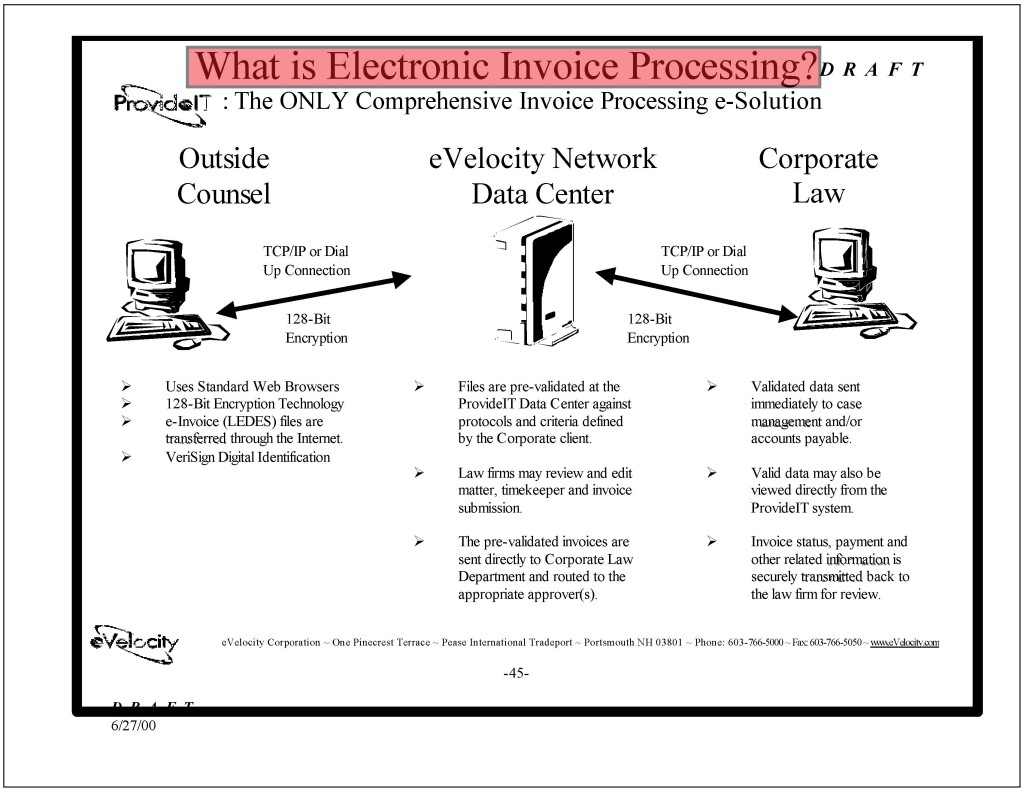


Figure B.5: *Figure showing the mis-classified region. The highlighted region was labeled as title in the ground truth dataset. However, it was predicted as a page header by the model.*

In this example, the model mis-classifies the highlighted region as a page header. The region contains the following text: "*What is Electronic Invoice Processing?*".

It is common for page headers to appear at the very beginning or top of a document page. The model captures this in the 'vertical position' local feature, which

represents the vertical position of the region in relation to the document page i.e. top, middle or bottom. Since the highlighted region (i.e. title) appears at the top of the document page, some ambiguity is encountered by the model and not enough information is present to predict the region as a title.

A possible solution is to introduce more local features to help distinguish a title from a page header in case they appear around the same location - at the very top of the document page. One of such local features could be a 'horizontal position' of the region. It is more common for titles to appear at the middle of a document page compared to page headers that are usually placed at the extreme left or right of the document page.

B.2.5 Heading

The model predicts headings as list items 7 times, as shown in figure B.1. Below is an example of one of such mis-classifications:

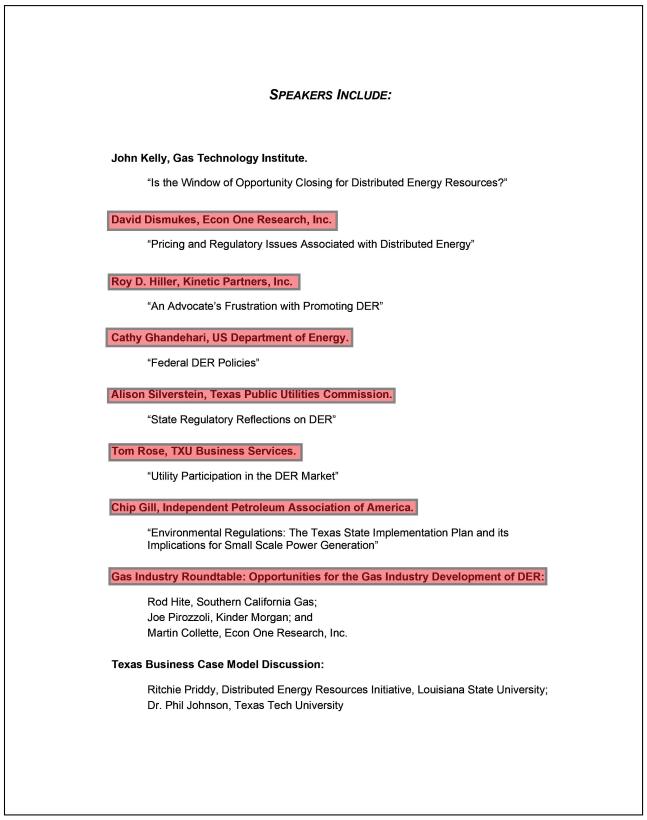


Figure B.6: *Figure showing the mis-classified regions. The highlighted regions were labeled as headings in the ground truth dataset. However, they were predicted as list items by the model.*

In the example above, the highlighted regions show the mis-classified samples. Instead of being identified as headings, the model predicts the regions as list items.

The reason for this is because the previous region before the first highlighted (i.e. region 3), was also wrongly predicted as a list item (instead of paragraph) and the model seems to begin to assign subsequent labels based on the contextual classifier (taking into account the likely transition of list item $->$ list item). As a digression, the mis-classification of region 3 i.e list item instead of paragraph, is another type of prediction error the model makes, which has not been discussed previously. In this case, the quotation marks in the region's text - " *"Is the Window of Opportunity Closing for Distributed Energy Resources?"* " are interpreted as non-ASCII characters by the OCR engine and thus is represented as a list item feature by the model.

A possible solution for the wrong prediction of headings as list items is to introduce more local features per region such as 'change of indentation', to signify that a region's indentation level has changed. It is common for list items to be slightly right-indented to other parts of a document page such as headings or paragraphs.

B.3 Splitting Ambiguous Labels

After carrying out analysis on the evaluation dataset and some of the predictions made by the model, we deduced that some of the labels defined for our task are either too ambiguous or generally confusing.

To solve label ambiguity, we revised the training and test set documents of the dataset and corrected erroneous annotations in them. It was also discovered that the definition given to the label caption was too general and hence, confusing to both human annotators (see table **??**) and inherently, the model classifier. To solve this confusion, we redefined the caption label and introduced new labels. We provide an example below.

B.3.1 Example

The example in figure B.7 shows a document image with a caption, table and caption as annotated in the evaluation dataset. As highlighted in B.2.2, regions that surround and define items such as tables, figures etc are labeled as *captions*. However, this causes ambiguities as shown in this example. Is region 1 really a caption? Is region 3 also a caption? Does region 3 really define the table?

To deal with these confusing 'caption' labels, we limited the definition of captions to be - "regions that surround an item (e.g., table) and that start with keywords such as 'table', 'figure', 'fig' etc". So in this example, region 1 and 3 do not meet this criteria. Hence, we split the 'heading' label into *section heading* and *item heading*

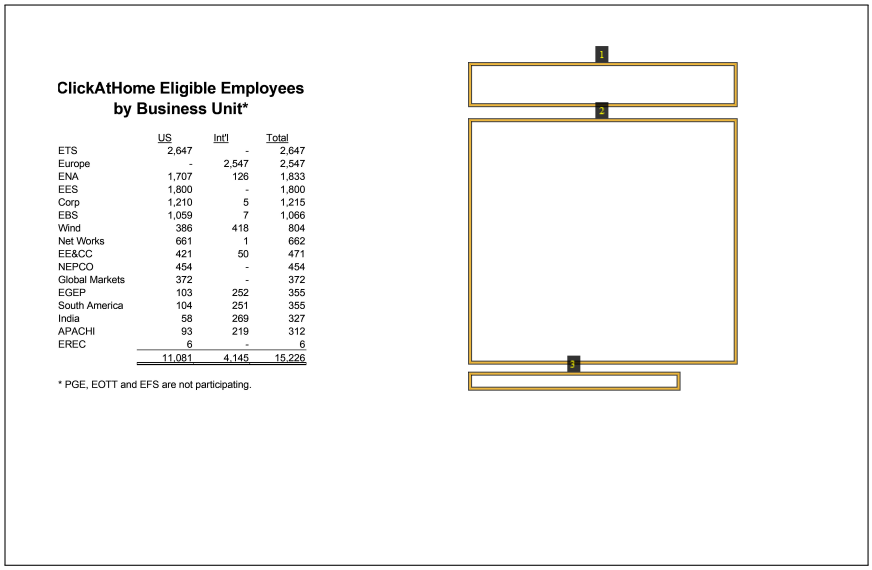


Figure B.7: *A document image and it's segmentation. In the evaluation dataset, region 1 is labeled as a caption, region 2 as a table and region 3, a caption.*

and the footer label into *page footer* and *item footer*. The 'item heading' covers regions such as region 1 which appears at the top of an item and defines it but does not contain any caption keyword. 'Item footer' on the other hand covers regions such as region 3 that lies under an item and makes references to the item but does not contain a caption keyword. In this way, we deal with the ambiguity in the caption label but also simplify the heading and footer labels.

Label category	Top 2 features	Description
Caption	startswithsource	Indicates if text content begins with 'source:'
	has_tablecaption	Indicates if text content contains 'table:'
List item	startswithlistitempattern	Indicates if text starts with bullet item
	prev_textcolon	Indicates if previous text region content ends with colon
Page number	is_top	Indicates vertical position of text content
	next_fontsize	Indicates there is a change in font size from current region to next region
Paragraph	has_multiple_whitespace	Indicates if text content contains consecutive multiple whitespaces.
	end_of_regions	Indicates last region in sequence
Table	has_multiple_whitespace	Indicates if text content contains consecutive multiple whitespaces.
	line_bin_large	Categorizes number of lines in text region into small, medium or large.
Title	line_bin_small	Categorizes number of lines in text region into small, medium or large.
	prev_textcolon	Indicates if previous text region content ends with colon
Footer	has_email_pattern	Indicates if text content contains an email pattern or keyword
	has_digit	Indicates if text content contains digit
Heading	prev_fontsize	Indicates there is a change in font size from previous region to current region
	is_sentence_capitalized	Indicates if text content is all capitalized
Page header	starts_with_digit	Indicates if text content starts with a digit
	close_to_top	Indicates if y axis of text region falls within the top border of document
Email header	endswithcolon	Indicates if text content ends with colon
	has_email_pattern	Indicates if text content contains an email pattern or keyword
Email body text	is_ascii	Indicates if text content contains all ASCII characters
	prev_textcolon	Indicates if previous text region content ends with colon
Email signature	heightratio	Indicates the height ratio of previous region and current region
	avg_font_size	Indicates the average font size of tokens in text region
Email footer	bullet_pattern	Indicates if text region contains unicode bullet characters
	heightratio	Indicates the height ratio of previous region and current region

Table B.3: *Top 2 features learned by the LC-CRF$_2$ model for each label category and their descriptions.*

Appendix C

Additional Experiments

In view of some of the results, conclusions and recommendations from the research work, we carried out a few additional experiments in an attempt to improve the performance of the models.

C.1 Experiment 1: 100 additional documents and corrected annotations

We annotated an additional 100 documents in the remaining time left of the research duration and added it to the training set. The annotation was done independently by the main author of this report, focusing on documents containing label categories which were found to be ambiguous in the original dataset e.g. caption, list item, heading. We also inspected the annotations in the original dataset and split ambiguous labels as described in B.3. However, evaluation was still done on the same label categories introduced in the research paper. We ignored the new categories because there were very few instances of them in the training set. It is worthy to note that *section heading* represents *heading* before the label split and *page footer* represents *footer*. In table C.1, we show the new results for only the LC-CRF$_2$ model and thus, compare it with its previous results. It is also noteworthy to mention that the RNN$_2$ model had a similar range of improvement in the overall micro F_1 score (from 0.58 to 0.61), after revising the annotations and including the additional documents in the training set.

	LC-CRF$_2$	LC-CRF$_2new$
Overall Micro F_1	0.830	**0.867**
table	**0.885**	0.906
paragraph	0.754	**0.818**
page number	0.959	**0.971**
list item	0.589	**0.635**
section heading	0.545	**0.769**
page header	0.875	**0.919**
title	0.720	**0.730**
page footer	**0.875**	**0.875**
caption	0.708	**0.800**
email header	0.980	**0.987**
email body text	**0.980**	**0.980**
email signature	**0.974**	**0.974**
email footer	**0.985**	0.969

Table C.1: *Comparative performances among original LC-CRF$_2$ model and the new model after correcting the annotations and adding 100 documents focused on ambiguous labels (LC-CRF$_2new$). Category-specific performance given in F1. Results in bold mark the best model for each category.*

From the results shown above, we see a significant increase in performance of the ambiguous labels e.g. caption, list item and section heading. This indicates a positive effect of the splitting of labels and additional data focusing on those labels. The overall micro F1 also increases by 0.2%.

C.2 Experiment 2: Improving the LSTM Network

We further investigated the reason(s) why the LSTM network was giving unsatisfactory results. Though we concluded that more data is needed to see improving performances and our methods for data augmentation may have been limited, we decided to pursue a more in-depth investigation into the network. The further investigation led us into discovering an issue related to the input data being fed into the network. As stated in appendix A, the input we fed into the network was a transformed feature set of the hand-crafted CRF features into a 3D tensor. However, these tensor values were un-normalized. To understand the problem with un-normalized values, we will briefly recap how training occurs in neural networks and an optimization concept

called **gradient descent**.

When the neural network is fed with inputs/observations, it produces an expected output which is compared to the actual output of the observation. Gradient descent is then used to update the parameters of the model in the direction which will minimize the error (difference between expected output and actual output) that we observe in the model's predictions. In more detailed terms, what gradient descent does is to find the values of each parameter where the loss function is minimized on a loss function surface.[1] Summarily, it's a search for the lowest or minimum value on a loss function topology/surface.

The problem with un-normalized or unscaled values as input into the network is that when the network combines these inputs through a series of linear combinations and nonlinear activations, they don't match the scale of parameter values associated with each input (i.e. they exist on different scales). This causes an awkward loss function surface where the gradients of larger parameters dominate the updates. This is why it is important to normalize the input values. Normalizing/scaling the input values to a standard scale helps the network to learn the optimal parameters for each input node quickly and therefore, quickly find the minimum loss. In addition, it is recommended that the inputs and target outputs are within the typical range of -1 to 1 or else the default parameters for the neural network such as the learning rate, will likely be ill-suited for the data.[2]

To normalize the input values for our network, we used the batch normalization technique [12]. Batch normalization helps to improve the convergence properties of the network and has the effect of accelerating the training process of a neural network, and in some cases improves the performance of the model. We implemented this batch normalization technique by adding a batch normalization layer before the input layer and before every hidden layer in the LSTM network. The batch normalization layer normalizes the activations of the previous layer at each batch and will transform inputs so that they are standardized, meaning that they will have a mean of zero and a standard deviation of one. We carried out another experiment on the LSTM using the revised training set (100 additional documents and corrected annotations) and evaluated on the revised test set. The effect of batch normalization in our network is seen in the experiment results below:

[1] See https://www.jeremyjordan.me/gradient-descent/

[2] See https://www.jeremyjordan.me/batch-normalization/

	RNN_2	RNN_2new
Overall Micro F_1	0.58	**0.86**
table	0.37	**0.87**
paragraph	0.50	**0.78**
page number	0.69	**0.97**
list item	0.26	**0.74**
section heading	0.50	**0.65**
page header	0.66	**0.91**
title	0.41	**0.74**
page footer	0.72	**0.87**
caption	0.07	**0.67**
email header	0.70	**0.97**
email body text	0.79	**0.98**
email signature	0.85	**0.96**
email footer	0.76	**0.98**

Table C.2: *Comparative performances among original RNN_2 model and the model after including batch normalization layers before the input and hidden layers (RNN_2new). Category-specific performance given in F1. Results in bold mark the best model for each category.*

From the results shown above, we see the vast and significant difference that normalizing the inputs makes. There is a 28% increase in the overall micro F_1 score and significant increase in all the labels including the ambiguous ones. The overall F_1 of 0.86 shows a match in performance with the LC-CRF_2new model, indicating the critical importance of the batch normalization step.

Appendix D

User Guide

Usage

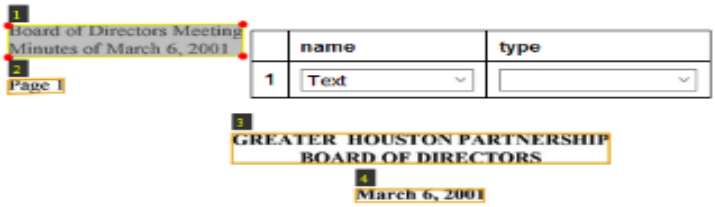
1. To begin using the application, open the **'via.html'** file with a Chrome or Firefox browser. If any of these browsers are not your default browser then right click on the file and select 'Open With' to choose either browser option. The opened file will look as shown in the figure below.

2. Click on the **'Project'** menu and click **'Load'** (i.e. Project > Load). This will open your file explorer, navigate to the main folder where the 'via.html' file resides. Select any of the **Pro_GroupXY.json** files in the directory of the application. (where XY represents a number from 1 - 40). This will load the project containing the segmented files including all project settings.
   NB – Only incomplete project group numbers should be worked on. Hence, open this link to confirm incomplete groups
   https://docs.google.com/document/d/1kd0S84fE4DwvbQ3PZAN6W19Srsm7-cfckQ7DN1BjQHc/edit?usp=sharing

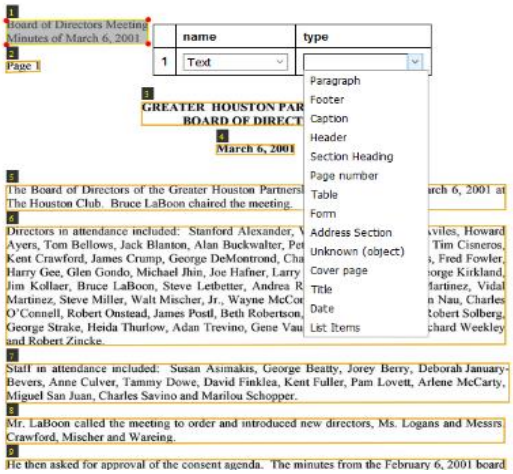
3. Each project contains 10 files. The task is to select a type attribute for every region in each file in the project.

Labeling Steps

1. To label a region, click within the bounding box of the region and a **Toggle Annotation Editor** pops up to the right side of the region as shown below:



2. Two attributes are shown in the Annotation Editor – name and type. The name attribute is pre-filled because most regions contain Text (A more detailed explanation on the attribute values is presented from page 4). The type attribute also needs to be filled. To do this, click on the drop-down and select an appropriate option from the list.



3. When all regions in **each file** have been labeled appropriately, the annotations need to be saved. To do this, hover on the **'Annotation'** menu and click on **'Export Annotations as CSV'.** Save the csv file that has been automatically generated.

4. Rename the saved **.csv file** as **GroupXY** (where XY is a number from 1-40). This indicates the project group number that you have worked on (See page 2). Please send this .csv file to senendu5@yahoo.com then fill in your name in the group row you've worked on using this link > https://docs.google.com/document/d/1kd0S84fE4DwvbQ3PZAN6W19Srsm7-cfckQ7DN1BjQHc/edit?usp=sharing

Name and Type Attributes

The name attribute in the Toggle Annotation Editor contains 4 options to select from while the type attribute contains 13. This section will help to guide you in making a choice for the appropriate 'name' value and 'type' value.

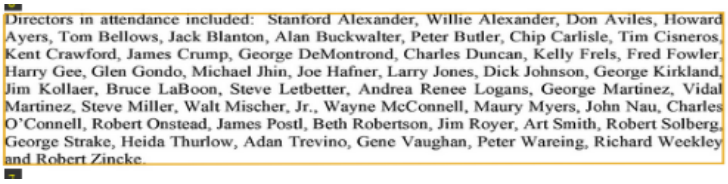
Name

The annotation task is to select a 'type' attribute. However the annotator is allowed to override the 'name' attribute when it may be necessary to do so.

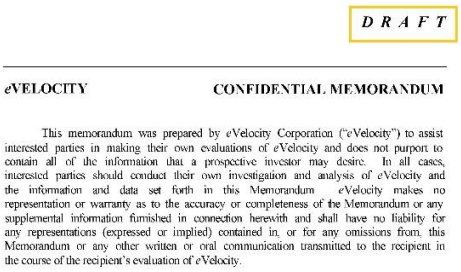
- **Text (Default)** – This is the default selection. Most regions contain textual content hence why it is selected by default. Text should be selected whenever the region being labeled is a text region or dominantly contains text.
- **Graphic** – Graphic should be selected whenever the region is a graphic such as (charts, plots). When Graphic is chosen, the type attribute **MUST BE** 'Unknown (object)'.
- **Image** – Image should be selected whenever the region is an image (other than charts or plots e.g. logos). When Image is chosen, the type attribute **MUST BE** 'Unknown (object)'.
- **Line/Separator** – When Line/Separator is chosen, the type attribute **MUST BE** 'Unknown (object)'. Line/Separator should be selected if the region is:
    - A line
    - A separator that separates sections
    - A line that signifies that start of a table
    - A line within a table or form

Type

1. **Paragraph –** This should be selected if the region is a paragraph. A paragraph typically looks as thus:


Directors in attendance included: Stanford Alexander, Willie Alexander, Don Aviles, Howard Ayers, Tom Bellows, Jack Blanton, Alan Buckwalter, Peter Butler, Chip Carlisle, Tim Cisneros, Kent Crawford, James Crump, George DeMontrond, Charles Duncan, Kelly Frels, Fred Fowler, Harry Gee, Glen Gondo, Michael Jhin, Joe Hafner, Larry Jones, Dick Johnson, George Kirkland, Jim Kollaer, Bruce LaBoon, Steve Letbetter, Andrea Renee Logans, George Martinez, Vidal Martinez, Steve Miller, Walt Mischer, Jr., Wayne McConnell, Maury Myers, John Nau, Charles O'Connell, Robert Onstead, James Postl, Beth Robertson, Jim Royer, Art Smith, Robert Solberg, George Strake, Heida Thurlow, Adan Trevino, Gene Vaughan, Peter Wareing, Richard Weekley and Robert Zincke.

2. **Page header –** A page header is typically found at the top of document pages. Select header if the region lies at the top left or right corner of the document image. There may also be other reasons for choosing header. The annotator should do this at their discretion. The image below shows an example of a page header, surrounded by the yellow-colored rectangle.



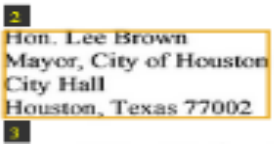
D R A F T

*e*VELOCITY CONFIDENTIAL MEMORANDUM

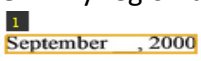
This memorandum was prepared by *e*Velocity Corporation ("*e*Velocity") to assist interested parties in making their own evaluations of *e*Velocity and does not purport to contain all of the information that a prospective investor may desire. In all cases, interested parties should conduct their own investigation and analysis of *e*Velocity and the information and data set forth in this Memorandum. *e*Velocity makes no representation or warranty as to the accuracy or completeness of the Memorandum or any supplemental information furnished in connection herewith and shall have no liability for any representations (expressed or implied) contained in, or for any omissions from, this Memorandum or any other written or oral communication transmitted to the recipient in the course of the recipient's evaluation of *e*Velocity.

3. **Title –** Title represents document titles. Titles are likely to be found at the top center of the document.

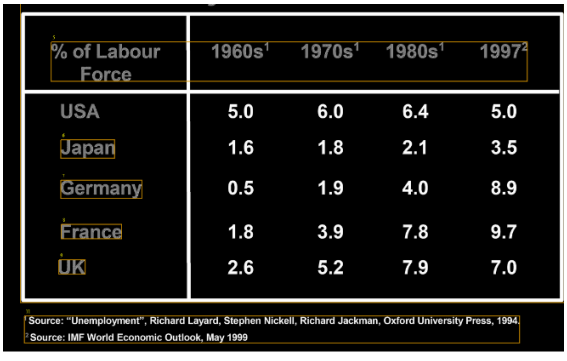
4. **Address Section –** This region will mainly be found in letter documents. This represents the sender or receiver's address and should be labeled as such.



5. **Date –** Any region that contains only a date should be labeled as Date



6. **Caption –** A caption label represents a caption for either an image, table, graph, chart or plot. That is, any text that is used to 'name' any of these items. Two captions can be seen in the figure below, beginning with the word, 'Source'.

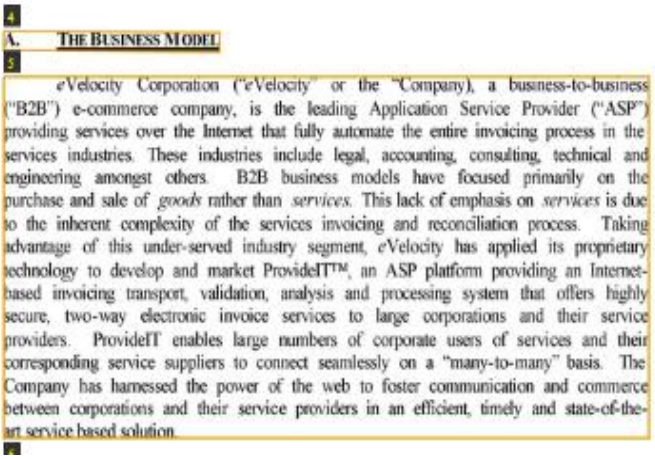


% of Labour Force	1960s[1]	1970s[1]	1980s[1]	1997[2]
USA	5.0	6.0	6.4	5.0
Japan	1.6	1.8	2.1	3.5
Germany	0.5	1.9	4.0	8.9
France	1.8	3.9	7.8	9.7
UK	2.6	5.2	7.9	7.0

[1] Source: "Unemployment", Richard Layard, Stephen Nickell, Richard Jackman, Oxford University Press, 1994.
[2] Source: IMF World Economic Outlook, May 1999

7. **Section Heading –** A section heading represents a heading that is used to begin a new section or item. This also includes table or form headings. An example is shown below. The region containing the text 'THE BUSINESS MODEL' is a section heading. Also the

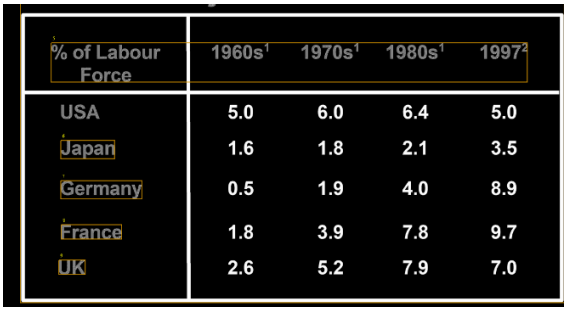
table heading in the figure above (i.e. '% of Labour Force, 1960s, 1970s etc')represents a section heading.



A. THE BUSINESS MODEL

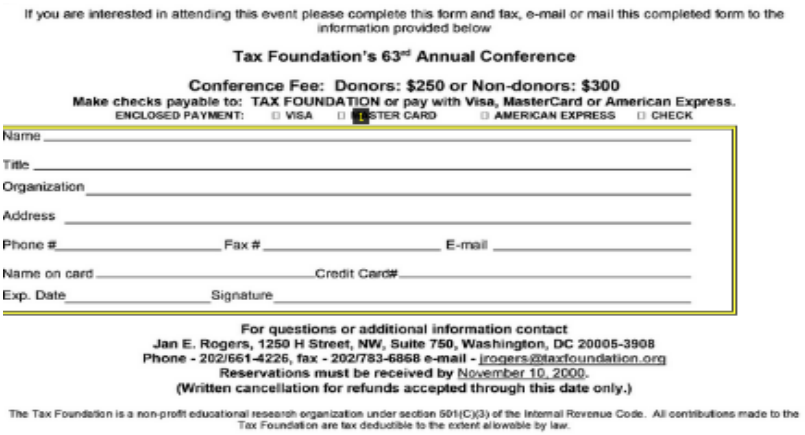
*e*Velocity Corporation ("eVelocity" or the "Company"), a business-to-business ("B2B") e-commerce company, is the leading Application Service Provider ("ASP") providing services over the Internet that fully automate the entire invoicing process in the services industries. These industries include legal, accounting, consulting, technical and engineering amongst others. B2B business models have focused primarily on the purchase and sale of *goods* rather than *services*. This lack of emphasis on *services* is due to the inherent complexity of the services invoicing and reconciliation process. Taking advantage of this under-served industry segment, *e*Velocity has applied its proprietary technology to develop and market ProvideIT™, an ASP platform providing an Internet-based invoicing transport, validation, analysis and processing system that offers highly secure, two-way electronic invoice services to large corporations and their service providers. ProvideIT enables large numbers of corporate users of services and their corresponding service suppliers to connect seamlessly on a "many-to-many" basis. The Company has harnessed the power of the web to foster communication and commerce between corporations and their service providers in an efficient, timely and state-of-the-art service based solution.

8. **Table –** Quite straightforward, a region that houses a table should be labeled as a Table.



% of Labour Force	1960s[1]	1970s[1]	1980s[1]	1997[2]
USA	5.0	6.0	6.4	5.0
Japan	1.6	1.8	2.1	3.5
Germany	0.5	1.9	4.0	8.9
France	1.8	3.9	7.8	9.7
UK	2.6	5.2	7.9	7.0

9. **Forms –** A region should be labeled as a Form if it bounds a form or what looks like a form to the annotator.



If you are interested in attending this event please complete this form and fax, e-mail or mail this completed form to the information provided below

Tax Foundation's 63rd Annual Conference

Conference Fee: Donors: $250 or Non-donors: $300

Make checks payable to: TAX FOUNDATION or pay with Visa, MasterCard or American Express.

ENCLOSED PAYMENT: ☐ VISA ☐ MASTER CARD ☐ AMERICAN EXPRESS ☐ CHECK

Name _____

Title _____

Organization _____

Address _____

Phone # _____ Fax # _____ E-mail _____

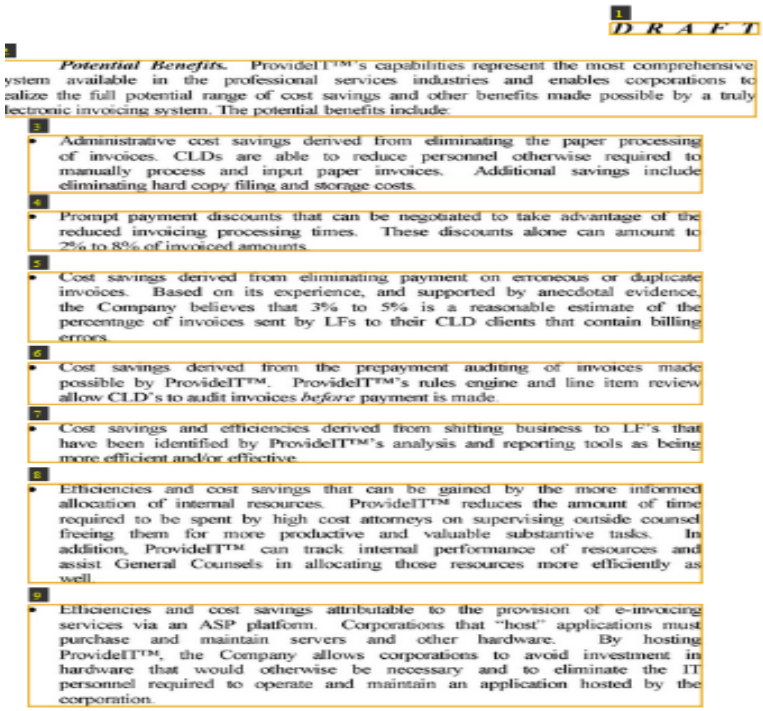
Name on card _____ Credit Card# _____

Exp. Date _____ Signature _____

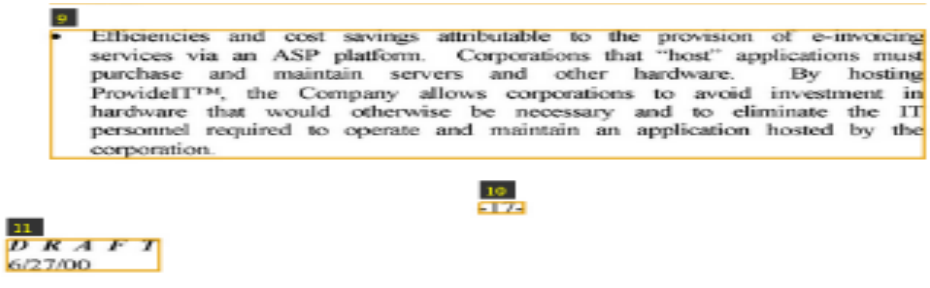
For questions or additional information contact
Jan E. Rogers, 1250 H Street, NW, Suite 750, Washington, DC 20005-3908
Phone - 202/661-4226, fax - 202/783-6868 e-mail - jrogers@taxfoundation.org
Reservations must be received by November 10, 2000.
(Written cancellation for refunds accepted through this date only.)

The Tax Foundation is a non-profit educational research organization under section 501(C)(3) of the Internal Revenue Code. All contributions made to the Tax Foundation are tax deductible to the extent allowable by law.

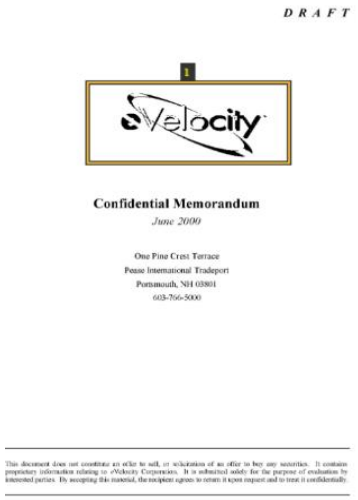
10. **List Items –** A region should be labeled as a list item if it contains text arranged as a list whether ordered or unordered (bullet lists, numbered lists etc).



11. **Footer -** A footer will likely appear separately from the main text in a page, at the bottom of the document. The region at the bottom left of the figure below can be considered a Footer region.



12. **Unknown (object) –** Use this label when the region is unidentifiable or cannot be interpreted as any of the other regions. This will usually be the case for a graphic or image **‘name’** attribute. The region with the ‘Velocity’ logo in the figure below should take the unknown label.



13. **Page number –** Select page number when the region represents the number of the page. This could be just a number as in ‘5’ or could be represented in other ways such as ‘Page 5’ or ‘Page 5 of 10’ etc.

14. **Email Header –** This region is specific for email document images. It is the region that is normally found at the top of an email containing the addresses (i.e. From, Sent, To) and subject of the email. Email headers can be also be found at the center or end of the page, its main property is that it contains the addresses and/or subject of the email.

15. **Email salutation –** This region is specific for email document images. It is the region that is normally used to begin an email. (i.e. Dear/Hello/Hi).

16. **Email body text –** This region is specific for email document images. It is the region that contains the body of the email i.e. the email message.

17. **Email Signature –** This region is specific for email document images. It is the region that is normally found at the end of an email message. It includes the signature of the email sender (i.e. personal name/company details).

18. **Email Footer –** This region is specific for email document images. It is the region that is normally found at the bottom of the email document image. It usually signifies

False and Overlapping Regions

In some of the segmented document images, there are regions that overlap one another. This is especially common for documents that contain tables or forms. Specific contents within the tables or forms are also segmented thereby causing overlaps and unnecessary regions. An example is shown in the figures below to the left:

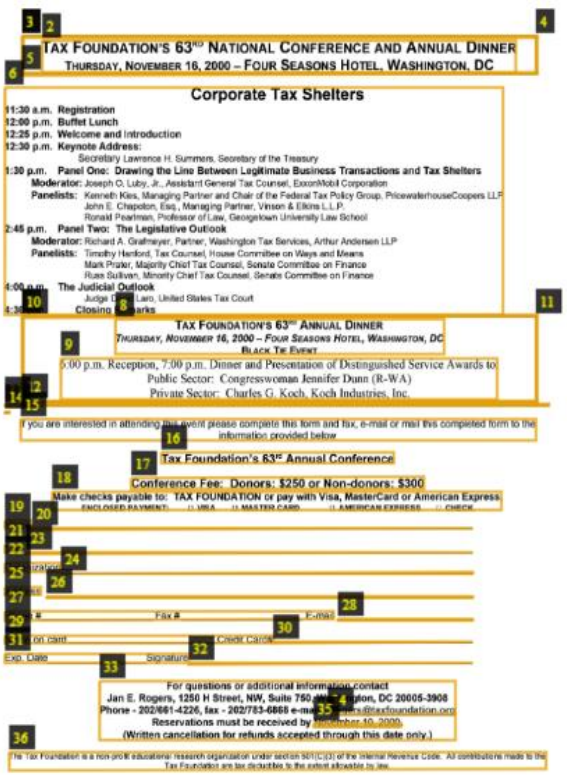


Figure 1: Overlapping Regions

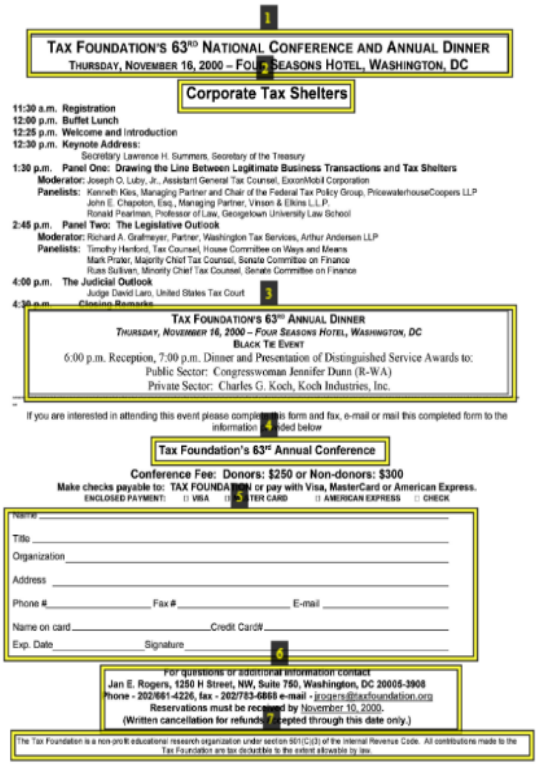


Figure 2: Corrected Segmentation

We want to get rid of such overlapping segmented regions and hence turn figure 1 into what figure 2 looks like. Note that it is sufficient to have one bounding box that bounds the entire table or form (region 5 in figure 2), there is no need to segment the contents within them. However for tables, the annotator should include bounding boxes for table headers as shown in figure 3 below and the type attribute should be 'Section heading'.

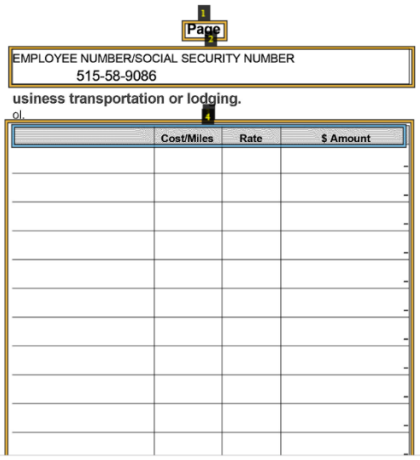
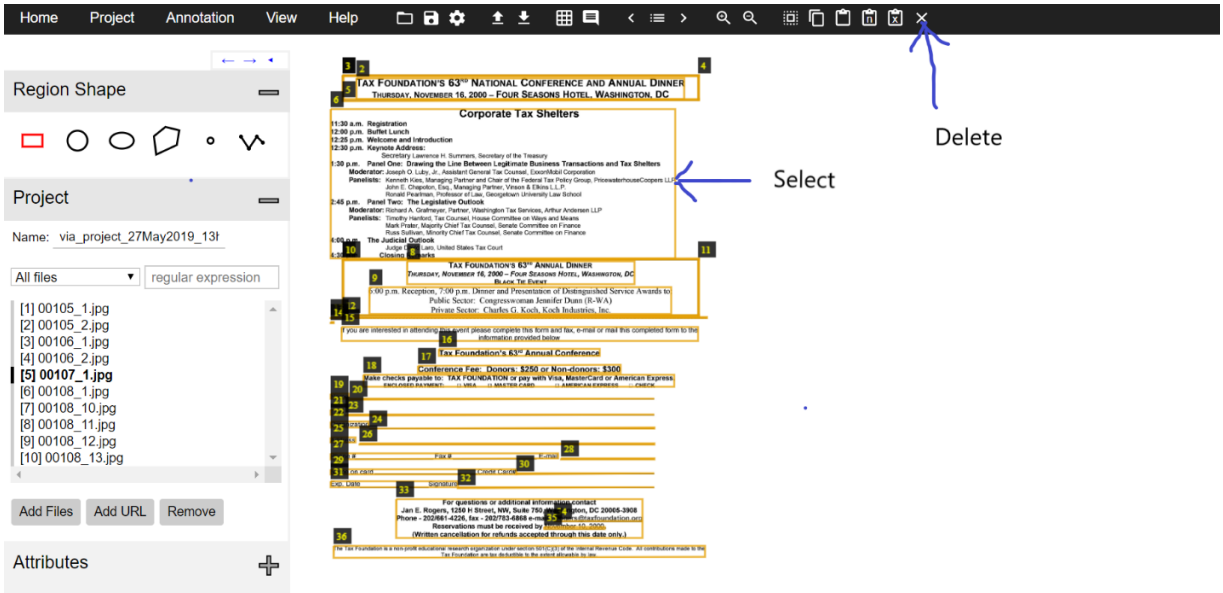


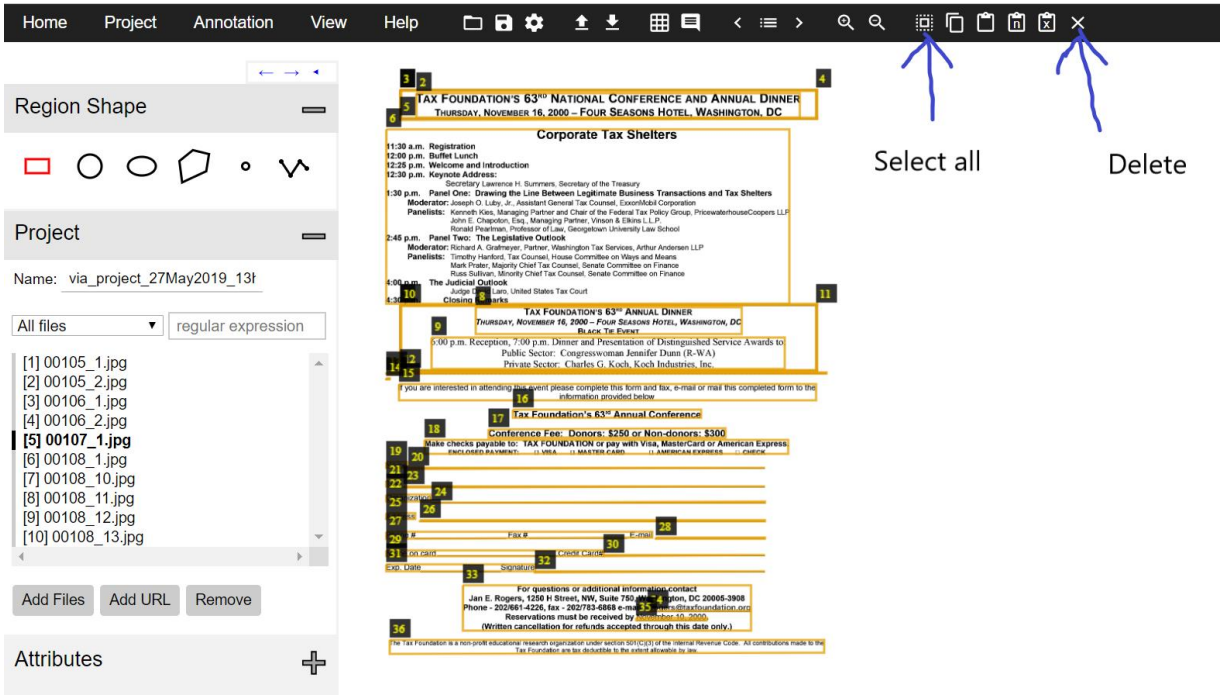
Figure 3: Table header bounded with a bounding box. (Region should be labelled as section heading)

The easiest way to deal with this situation is to undo all or some of (the overlapping) segmented regions and for the annotator(s) to manually segment the document themselves before labelling. This can be achieved by carrying out the following steps:

1. **Delete a Region**: In case the annotator wants to delete just a region, select the region then click the icon to the farthest right 'Delete Region'. This is illustrated in the figure below



2. **Delete all Regions:** In case the annotator wants to delete just all regions, click the 'Select all regions' icon and then the 'Delete Region' icon. This is illustrated in the figure below



3. **Manually drawing regions:** Regions can be drawn by clicking and dragging across the document image. The rectangular shape is selected by default and should be used when drawing regions manually.

Submission and Finalization

1. When all regions in **each file** have been labeled appropriately, the annotations need to be saved. To do this, hover on the **'Annotation'** menu and click on **'Export Annotations as CSV'.** Save the csv file that has been automatically generated.

2. Rename the saved **.csv file** as **GroupXY** (where XY is a number from 1-40). This indicates the project group number that you have worked on (See page 2). Please send this .csv file to senendu5@yahoo.com then fill in your name in the group row you've worked on in this link https://docs.google.com/document/d/1kd0S84fE4DwvbQ3PZAN6W19Srsm7-cfckQ7DN1BjQHc/edit?usp=sharing

3. If you have any questions, please send an email to senendu5@yahoo.com

Bibliography

[1] F. Peng and A. McCallum, “Accurate information extraction from research papers using conditional random fields,” in *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2004, Boston, Massachusetts, USA, May 2-7, 2004*, 2004, pp. 329–336.

[2] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML ’01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 282–289. [Online]. Available: http://dl.acm.org/citation.cfm?id=645530.655813

[3] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[4] A. Adnan and S. Ricky, “Page segmentation and classification utilizing bottom-up approach,” *International Journal of Image and Graphics*, vol. 01, 2011.

[5] K. Kise, A. Sato, and M. Iwata, “Segmentation of page images using the area voronoi diagram,” *Comput. Vis. Image Underst.*, vol. 70, no. 3, pp. 370–382, 1998. [Online]. Available: http://dx.doi.org/10.1006/cviu.1998.0684

[6] A. Antonacopoulos, C. Clausner, C. Papadopoulos, and S. Pletschacher, “Icdar2015 competition on recognition of documents with complex layouts - rdcl2015,” *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1151–1155, 2015.

[7] C. Clausner, A. Antonacopoulos, and S. Pletschacher, “Icdar2017 competition on recognition of documents with complex layouts - rdcl2017,” in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, 2017, pp. 1404–1410.

[8] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. Shamima Nasrin, M. Hasan, B. C. Van Essen, A. Awwal, and V. Asari, "A state-of-the-art survey on deep learning theory and architectures," *Electronics*, vol. 8, p. 292, 03 2019.

[9] X. Yang, E. Yumer, P. Asente, M. Kraley, D. Kifer, and C. L. Giles, "Learning to extract semantic structure from documents using multimodal fully convolutional neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4342–4351.

[10] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1064–1074. [Online]. Available: https://www.aclweb.org/anthology/P16-1101

[11] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *ArXiv*, vol. abs/1712.04621, 2017.

[12] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167