



DEGREE PROJECT IN COMPUTER SCIENCE AND ENGINEERING,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2021

Extracting Information From PDF Invoices Using Deep Learning

DIEGO LEON

Extracting Information From PDF Invoices Using Deep Learning

DIEGO LEON

Master in Computer Science

Date: August 17, 2021

Supervisor: Mårten Björkman

Examiner: Olov Engwall

School of Electrical Engineering and Computer Science

Host company: Bontouch AB

Swedish title: Extrahering av information från PDF-fakturor med
hjälp av djupinlärning

Abstract

Manually extracting information from invoices can be time-consuming, especially when managing large amounts of documents. Finding a way to automatically extract this information could help businesses save resources. This thesis investigates the information extraction of semi-structured data from PDF invoices using deep learning methods and comparing them to a rule-based model built as a baseline for comparison. More specifically, an object detection approach based on the Faster R-CNN model is compared with a Natural Language Processing (NLP) approach based on BERT. These models were trained to extract 4 different fields, with a dataset consisting of 899 PDF invoices. These models were tested on how well they extracted each field, and their results were then compared. The NLP approach achieved the highest overall F1 score of 0.911 and attained the highest score in all fields except one. In second place came the rule-based approach, with an overall F1 score of 0.830. In last place came the object detection approach with an overall F1 score of 0.815. It is concluded that the NLP approach is best suited for the task of information extraction from PDF invoices. Because of the small dataset and Faster R-CNN requiring large amounts of data and long training, the object detection approach did not reach its full potential. However, further research is needed to prove if it could outperform the NLP approach with those improvements.

Sammanfattning

Manuell extrahering av information från fakturor kan vara tidskrävande, särskilt om det gäller stora mängder dokument. Att hitta ett sätt att automatiskt extrahera viktig information kan hjälpa företag att spara resurser. Denna avhandling undersöker informationsutvinning av semistrukturerad data från PDF-fakturor med djupinlärningsmetoder och jämför dem med en regelbaserad modell byggd som en basmetod för jämförelse. Mer specifikt jämförs en metod för objekt-detektering baserad på Faster R-CNN modellen med en språkbehandlings-metod baserad på BERT. Dessa modeller tränades för att extrahera fyra olika fält, med ett dataset bestående av 899 PDF-fakturor. Modellerna testades på hur väl de extraherade varje fält. NLP-metoden uppnådde den högsta totala F1 resultatet på 0,911 och hade bäst poängen i alla fält utom ett. På andra plats kom den regelbaserade metoden med F1 resultatet 0,830. På sista plats kom objekt-detekteringsmetoden med F1 resultatet 0,815. Som slutsats är NLP-metoden bäst lämpad för att extrahera information från PDF-fakturor. På grund av den lilla mängden data som användes så nådde inte objekt-detekteringsmetoden sin fulla potential eftersom Faster R-CNN kräver stora mängder data och längre träning. Däremot krävs ytterligare forskning för att bevisa om den kan överträffa NLP-metoden med dessa förbättringar.

Acknowledgments

I would like to thank the host company of this thesis, Bontouch AB, for the opportunity to perform this project. I would also like to give some credit to my KTH supervisor Mårten Björkman and my supervisor at Bontouch AB, Philip Montalvo, for supervision and guidance.

Thank you,
Diego Leon

Contents

1	Introduction	1
1.1	Purpose	2
1.2	Problem definition	2
1.3	Research Questions	3
1.4	Scope & limitations	3
1.5	Outline	4
2	Background	5
2.1	Machine Learning	6
2.1.1	Classification	6
2.1.2	Generalization	6
2.2	Artificial Neural Networks	7
2.3	Natural Language Processing	9
2.3.1	Encoder-Decoder Architecture	10
2.3.2	Transformer	11
2.3.3	BERT	12
2.4	Convolutional Neural networks	13
2.4.1	Convolutional layer	14
2.4.2	Pooling layer	14
2.4.3	Fully connected layer	15
2.5	Object detection	15
2.5.1	R-CNN, Fast R-CNN, and Faster R-CNN	16
2.6	Tools	17
2.7	Related work	17
3	Methods	20
3.1	Dataset	20
3.2	Rule-based	23
3.3	Object detection approach	24

3.3.1	Data Pre-processing	24
3.3.2	Annotation	24
3.3.3	Faster R-CNN	26
3.3.4	Rule-based extraction	26
3.4	BERT	27
3.4.1	Data Pre-processing	27
3.4.2	Annotation	27
3.4.3	BERT model	28
3.5	Test Setup & hardware	29
3.6	Evaluation method	29
3.6.1	F1 score	29
4	Results	31
4.1	Rule-Based	31
4.2	BERT	32
4.3	Object detection	33
4.4	Comparing results	34
5	Discussion	36
5.1	Sources of error	36
5.2	Performance analysis	38
5.3	Societal, Ethical and Sustainability Aspects	41
6	Conclusions and Future Work	43
6.1	Conclusions	43
6.2	Future work and improvements	44
	Bibliography	48

Chapter 1

Introduction

Every day a large number of invoices are sent from businesses to clients. Invoices are important business documents since they enable companies to get paid for their services. With digitization on the rise and IT solutions such as banking apps gaining more traction, PDF invoices are becoming more popular in comparison to the physical variant.

The difference between invoice structures varies, which means that the locations of the fields and information on the invoices are not fixed. Deep learning has proven to be useful since it is specialized in solving problems by generalizing from data. Two common approaches to solve this problem are the computer vision approach and the Natural Language Processing (NLP) approach. Object detection is a common object recognition problem and belongs to the field of computer vision. It has the task of placing bounding boxes around each object of a particular category in an image. NLP in deep learning gives a computer the ability to analyze, understand, and even generate text.

When it comes to automatically extract information from PDF invoices regardless of structure, there are different ways to tackle this problem. An object detection approach would consist of two phases, detection, and recognition. This approach would first detect the text belonging to the searched class and then recognize the detected text by extracting the text from the image with an optical character recognition (OCR) tool. The second phase is dependent on the success of the first phase. For instance, if no price is detected no text can be extracted with OCR. The NLP approach does not need to detect the text for a class in an image. Instead, it directly analyses the textual content from the PDF invoice and extracts the searched class.

1.1 Purpose

To pay an invoice, specific information from it has to be extracted. The process of extracting this information is usually done manually and is time-consuming. The time it takes to extract this information depends on the type of invoice and the amount of invoices that must be managed. This process is also prone to human error.

The host company, Bontouch AB, develops the mobile app for Skandinaviska Enskilda Banken (SEB), which is a leading Nordic financial services group. A core component of the SEB app is its payment feature. Thousands of SEB customers use their phones to make transfers, pay invoices, etc. With the world turning greener, PDF invoices are becoming an increasingly popular alternative to the physical variant. To make life even easier for the users, Bontouch AB would like to explore the possibility of extracting payment information from PDF invoices. The host company has a solution based on heuristics, that extracts Reference, Price, and Receiver for invoices that have a specific structure.

The purpose of this thesis is to investigate the possibility of extracting payment information from PDF invoices using deep learning methods compared to a rule-based method. Results and conclusions drawn from this research could be valuable for further research in this field and developing software that extracts payment information from invoices. Apart from saving companies and users time, it could save companies money that can be spent on more important tasks.

1.2 Problem definition

Automating the extraction process is not trivial since a PDF invoice does not have a fixed structure. Making a general model based on rules and patterns that works for every invoices is therefore a difficult task. An approach that solves this problem should be able to extract information from a PDF invoice regardless of the invoice structure. Other challenges that may arise concern the training data used by the deep learning models. The amount and quality of the data could impact the results as well.

This thesis compares three approaches to extracting information from PDF

invoices, two deep learning approaches and one rule-based approach. The first approach uses the BERT model to focus on the textual content of the invoice, and is based on NLP. The second approach uses object detection and treats PDF invoices as images to learn the spatial and visual features of their fields. This approach is based on Faster R-CNN. The last approach is purely rule-based and serves as a baseline for comparison. These approaches were evaluated in terms of their ability to extract four fields from PDF invoices, specifically: Date, Price, Receiver, and Reference.

Both Faster R-CNN and BERT achieve state of the art results in their respective fields. Comparisons between models that utilize semantic information, and models that utilize visual features are not uncommon. However, there is no study where Faster R-CNN and BERT are directly compared when performing the task of information extraction from PDF invoices. This thesis provides a new insight into which features that the models utilize are important when extracting information from PDF invoices.

1.3 Research Questions

This thesis aims to answer the following questions:

- How does an approach based on object detection compare with a NLP approach, in terms of data extraction from PDF invoices?
- How do both deep learning approaches compare with a completely rule-based approach, in terms of data extraction from PDF invoices?

1.4 Scope & limitations

This thesis covered the process of training two deep learning models. One based on object detection, and the other based on NLP, so that they can be used to extract payment information from PDF invoices. Both models were compared against each other and against a rule-based approach that serves as a baseline. To go from training to extracting information, several steps had to be done for each approach, namely data pre-processing, data annotation, training, validating, testing, extracting text from PDF invoices and images of PDFs using OCR. Data pre-processing, annotation, extracting text from PDF documents and images, are necessary steps for the final result, but are not investigated in this work. This thesis does not take into account the

computational cost of the used approaches and was neither measured nor discussed.

1.5 Outline

The thesis consists of 6 chapters each important to get a full understanding of the project, and follows an academic structure. The first chapter, *Introduction*, explains the problem, answers why the thesis is done, what the challenges are, what is going to be done, and what is not going to be done. Secondly, the chapter *Background* explains the necessary background information for the work, such as object detection, natural language processing, and in the end, related work is introduced. The third chapter, *Methods*, describes the methodology and experiments that will be conducted. The fourth chapter, *Results*, presents the results from the experiments. Chapter five, *Discussion*, contains a discussion about the results, ethics, and sustainability. Finally, chapter six, *Conclusions*, contains suggestions for future work, and ending remarks.

Chapter 2

Background

A PDF invoice is a semi-structured document containing information about products and services a business provides to a client. This document establishes an obligation from the clients' side to pay for those products and services provided by the business. In order to pay, specific information has to be extracted from an invoice. There are several approaches to information extraction from PDF invoices, but in this thesis, two approaches will be investigated. The first approach is based on Natural Language Processing (NLP) and tries to understand the textual content of a document in order to extract the desired fields. The second approach is based on object detection and tries to learn the spatial and visual features of an invoice in order to extract the desired fields.

This chapter contains the necessary background information for this thesis. Initially, the concepts: Machine learning (2.1), and Artificial Neural Networks (2.2) are explained. These concepts are important since many of the architectures and concepts described in this chapter are based on them. It's therefore important to understand both concepts before reading further about NLP and object detection. NLP (2.3) is then introduced along with the BERT architecture which the NLP approach is based on, and other architectures that BERT consists of. Furthermore, Convolutional Neural Networks (CNNs) (2.4) are explained. CNNs are important for understanding the Faster R-CNN architecture. Object detection (2.5) is then introduced along with Faster R-CNN which is the model that the object detection approach is based on. Tools (2.6), and Related work (2.7) are introduced at the end of this chapter.

2.1 Machine Learning

Machine learning (ML) is a field within computer science. It evolved through computational learning theory and pattern recognition subfields in artificial intelligence. ML has its focus on algorithms that can learn and make predictions based on data. These types of algorithms construct a model by receiving input with the purpose of making data-driven predictions and decisions, as opposed to following rules or instructions. The process where the model learns from the data is called "training". Inference refers to the process of using the trained model to make predictions based on data [1].

2.1.1 Classification

Classification tasks deal with predicting the discrete value of targets based on features/attributes. Based on previously correctly labeled data during training, the model learns to correctly label new unseen data. An example of a classification problem could be about predicting if an animal in a picture is a cat. The set of values from a class c for this problem could be $\{yes, no\}$, where yes represents positive decisions and no represents negative. The input would be images of cats and other animals so that the model learns the different characteristics of a cat [2].

2.1.2 Generalization

What makes machine learning stand out is its ability to generalize by creating sensible output from unseen data. An advantage of this is its ability to deal with noisy data, which refers to data containing small inaccuracies. Since the purpose is to classify unseen data correctly, we must use a test set to test the ML model instead of using the training set. This set is composed of inputs and targets, where the input is feed to the model and the output is compared with the target. Doing this will tell how well the model has learned [3].

It is important to also monitor how well the model generalizes during the learning phase since there is a danger in training for too long and not training enough. Training a model for too long leads to overfitting (memorizing) the data. This means that the model not only learned the desired attributes from the data but also the noise and inaccuracies as well. Figure 2.1 is an example of a model overfitting. It will be hard for this model to generalize, and the difference between training error and testing error will be large. Underfitting

occurs when the training error is too large, in this case the model barely learned any attributes at all. Overfitting can be avoided by terminating the training before the generalization ability is decreased. This is done by monitoring the generalization performance at each timestep using a separate set called validation set [3].

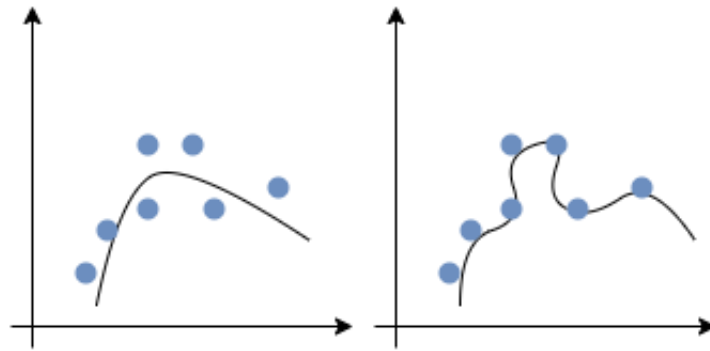


Figure 2.1: Overfitting causes the model to perfectly match the input (on the right), instead of finding the generating function (on the left)

2.2 Artificial Neural Networks

An artificial neural network (ANN) is a system that is inspired by the functioning processes of the human brain. Just like our brain, it processes input signals and transforms them into output signals. It can adapt and modify its internal structure depending on a function objective and is well suited for solving nonlinear problems. An ANN is also useful when the underlying function is unknown [4].

The human brain contains approximately 85 billion neurons. Each one of these neurons receives input signals from other neurons or the environment. Those signals are then processed and the output is then sent to other neurons. In the brain, learning occurs when the strength of synaptic connections between neurons is modified and new connections are created. However, the science of how exactly these mechanisms work has not been fully explored [3].

Artificial neurons

An artificial neuron works similarly to a neuron in our head. McCulloch and

Pitts modeled a neuron as a mathematical model, where only the important features were extracted to accurately represent a neuron [3]. Rosenblatt (1958) later proposed the Perceptron, which is built around this mathematical model. The Perceptron consists of a single neuron, binary inputs, adjustable weights, bias, and binary output, see Figure 2.2. Rosenblatt proved that his model converges when the patterns used for training are drawn from two linearly separable classes [5].

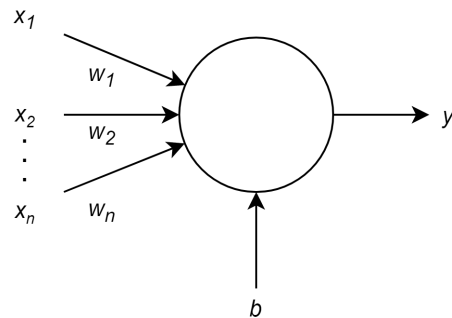


Figure 2.2: A picture of Rosenblatts Perceptron. The inputs x_i are multiplied by the weights w_i . These values are then summed and added with the bias b , then thresholded. If the value is greater than the threshold θ then y is 1, otherwise y equals 0.

If thresholding produces output 0, then the neuron fires, and if the output is 1 it does not fire. The function that makes this decision is known as an activation function.

Architecture

Artificial neural networks consist of 3 types of layers. The first type of layer receives the incoming data and is called the "input layer". The last one is called the "output layer", and the layers in between are called "hidden layers".

There are two main types of network architectures when it comes to neural networks, feed-forward neural networks, and recurrent neural networks (RNN). Which one to choose depends on how the neurons are connected. The first type only allows signals to travel from input to output. The output from one neuron is passed as input to neurons in the next layer, no loops are allowed. This means that the output from one layer does not impact the same layer [6]. The latter type allows signals to travel in both directions by enabling loops in the

network [7].

Learning

As described earlier, the model learns during the training phase. Initially, all network weights and biases are set with random values. An epoch represents the model passing through the training data. At the end of each epoch, the network weights and biases are adjusted with the end goal of improving the network predictions.

When it comes to feed-forward neural networks there are two main ways of learning, online learning, and offline learning. During training, online learning ingests data one instance at a time, while offline learning ingests all the data at one time to build the model [3].

Transfer learning

Humans tend to transfer knowledge gained doing one task and applying it to another task. The purpose of transfer learning is to use the knowledge gained from one task and using it to help solve another task. Transfer learning can therefore bring an initial performance boost, before using the training data, compared with a model that has no prior knowledge. The time gained by using transfer learning to train the model compared to training the model from scratch can be critical. Transfer learning can impact the final performance in a positive way [8].

Deep neural networks

The depth of a neural network is given by the number of layers it is made up of. A network with more than one hidden layer qualifies as a "deep" network [9]. By using multiple layers we make the model more complex and enable it to solve more difficult problems. Deep neural networks can be used in several areas, such as text recognition, image recognition, and speech recognition [10].

2.3 Natural Language Processing

Natural Language Processing (NLP) is a field of Artificial Intelligence that gives a computer the ability to analyze and understand human language in speech or text. Some of the challenges involve natural language generation,

speech recognition, natural language understanding, and named entity recognition [11]. Named Entity Recognition is a form of NLP and a sub-task of information extraction. It has the task of identifying and classifying key entities in text [12]. One of the methods to be investigated is based on NLP.

2.3.1 Encoder-Decoder Architecture

An encoder-decoder architecture solves the problem with handling the case where input and output are both of variable length sequences, see Figure 2.3. It contains two components, an encoder, and a decoder. In the usual case, each component consists of a recurrent neural network (RNN). The encoder transforms (encodes) the input sequence into an encoded vector. The purpose of this vector is to encapsulate the information for all input elements to aid the decoder in making accurate predictions. The encoding is then fed to the decoder which decodes the vector and outputs a sequence that can differ in size compared to the input size [13].

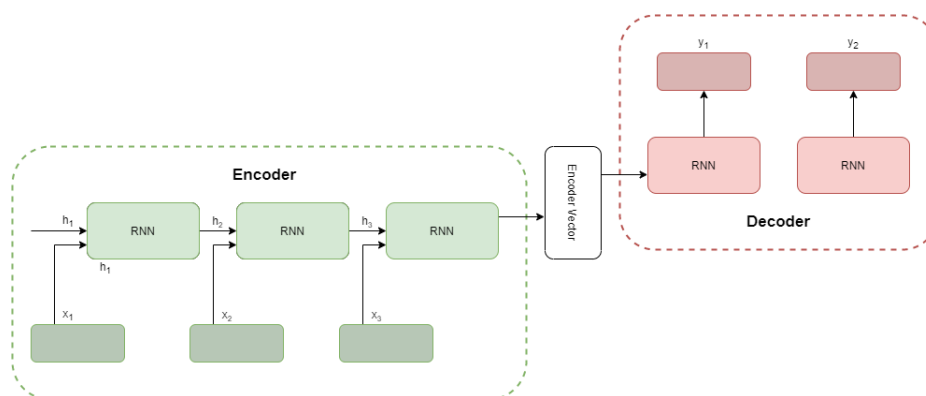


Figure 2.3: An overview of the Encoder-Decoder architecture

Attention

The attention mechanism was proposed to help neural networks handle long sentences. This improvement is done by feeding a context-vector to the decoder along with the encoded vector. This vector helps the decoder focus on the relevant parts of the original sequence at each step [14]

2.3.2 Transformer

In the paper 'Attention Is All You Need' a novel architecture called transformer is introduced [14]. This architecture is based on the encoder and decoder architecture and uses the attention mechanism. Figure 2.4 shows an overview of this architecture. However, it does not use RNNs. This paper proves that only using the attention mechanism can reach a new state of the art in translation quality.

To start, the sequence is fed into the encoder where each element gets an attention vector. This vector is calculated by equation 2.1. The matrix Q contains a vector that represents an element in the sequence. V is a vector that represents all the words in the sequence, and this sequence is matched to the keys in K . The length of the keys in K is represented by d_k . The output is the attention vector, and is called Dot-Product Attention and contains information about the most relevant parts of the input depending on this element.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2.1)$$

The paper also showed that it is better to calculate multiple attention operations in parallel. The attention vectors are then fed into a feed-forward layer which produces the final output and passes it to the decoder.

The decoder (right side of the architecture) has two attention layers that work differently. The first attention layer is a masked layer called Masked Multi-Head Attention. This layer is responsible for calculating multiple attention operations. This layer is necessary to prevent the model from looking into future words. It could otherwise get accustomed to copying the input from the decoder. The next step involves another Multi-Head Attention layer. The output from the Masked Multi-Head attention layer is fed into this layer as the query vector, and key and value are received from the encoder output. The output of this operation is sent to a feed-forward layer.

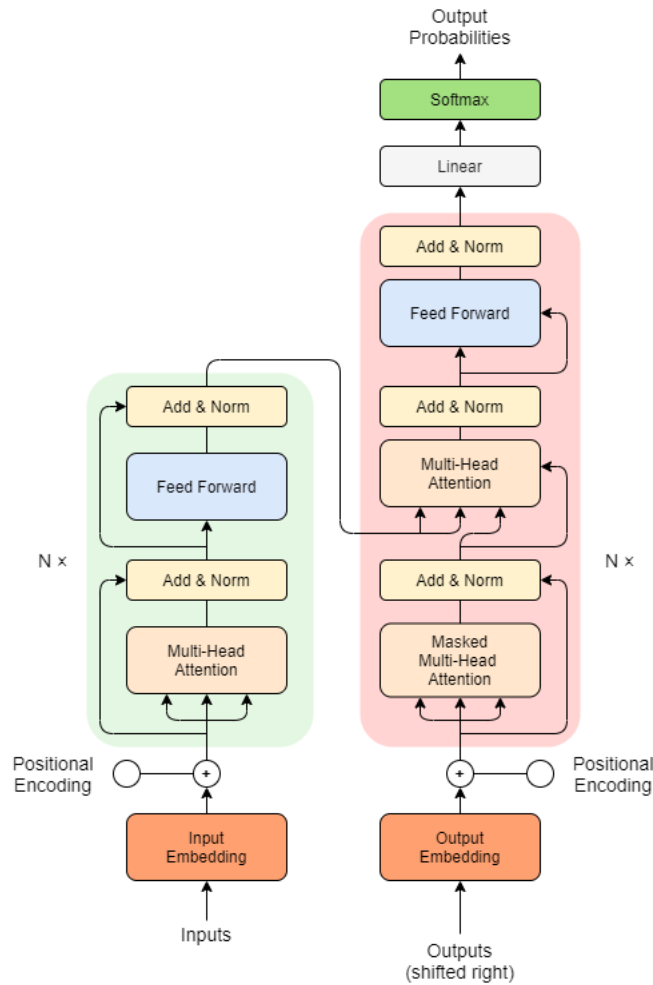


Figure 2.4: An overview of the Transformer architecture based on [14]

2.3.3 BERT

Bidirectional Encoder Representations from Transformers (BERT) is a model based on the transformer architecture and is pre-trained on bidirectional representations. This means that BERT can interpret representations from unlabeled text from both directions.

The training of BERT is first done with a pre-training phase, where the model is trained to understand language and context by training on two unsupervised tasks: Masked language modeling (MLM) and Next Sentence Prediction (NSP). BERT can then be fine-tuned to learn a specific task. Fine-tuning has

enabled BERT to achieve state-of-the-art results on several NLP tasks [15]. See Figure 2.5 for an overview of the pre-training and fine-tuning phases.

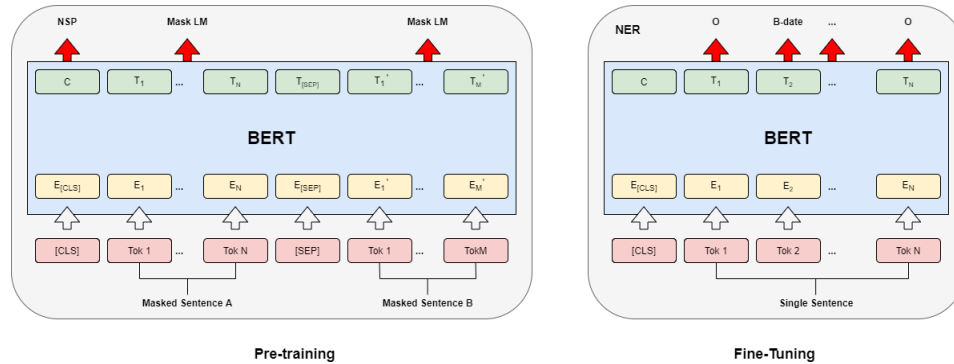


Figure 2.5: An overview of the BERT model based on [15]

2.4 Convolutional Neural networks

A convolutional neural network is a deep network with learnable weights and biases, that can receive images as input. It is based on convolution, which is the application of linear shift-invariant filters to the input, with an activation on the output. The result of this process is a feature map, which maps activations by saving the locations and strength of the detected features [16].

Traditional neural networks manage the interaction between each input unit and output unit, while convolutional networks have sparse interactions, which means that they can handle an image with millions of pixels as input by only focusing on meaningful features that occupy thousands of pixels. As a result, fewer parameters are stored, and fewer operations are needed to compute the output.

The convolutional neural network is based on three main layers: the convolutional layer, the pooling layer, and the fully connected layer, which are stacked to form a full CNN architecture. Unlike a traditional neural network, the layers of a CNN have neurons arranged in a three-dimensional way: width, height, depth [10].

2.4.1 Convolutional layer

The convolutional layer contains elements called kernels/filters which are responsible for extracting high-level features such as edges, from the input image. The kernel's height and width are smaller than the input image, but both have the same depth (eg. RGB). When each image passes through (convolutes) the layer, each filter is used and slides across the whole image. Dot product operations are performed between the filters and the region of the input image it is currently on top of (receptive field). The purpose of this operation is to form a matrix containing the values of each operation, which then forms the feature map [17], see Figure 2.6.

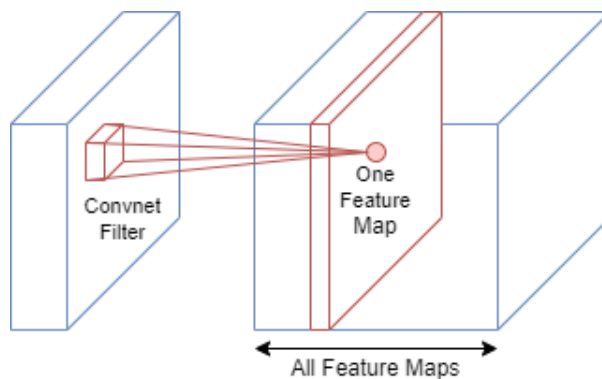


Figure 2.6: The convolutional layer outputs the result of applying a convolution to a subset of the previous layer's neurons.

2.4.2 Pooling layer

The purpose of this layer is to reduce the spatial size of the feature maps so that the number of parameters and computational power used to process the data is reduced. Doing this helps to extract dominant features. It is common to insert a Pooling layer in-between convolutional layers in a CNN architecture.

The two main types of Pooling used in CNNs are Max Pooling and Average Pooling. See Figure 2.7 for an overview of both processes. Max Pooling, the most used type of Pooling, returns a matrix with the maximum value in each receptive field, while Average Pooling returns a matrix with the average value [18].

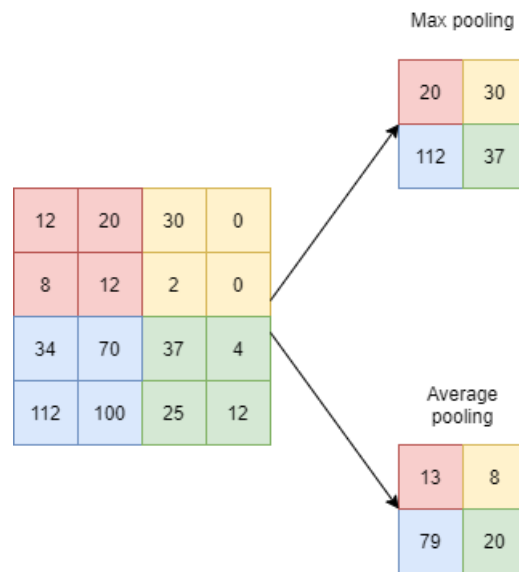


Figure 2.7: Max Pooling and Average Pooling

2.4.3 Fully connected layer

The purpose of this layer is to produce the final classification of the input image. All neurons in this layer have full connections to all activations in the previous layer. Fully connected layers are usually used as the last layers in the CNN architecture.

2.5 Object detection

Unlike image classification with CNNs explained before, this time the focus is on classifying objects within a certain image. Object detection is a common object recognition problem that has the task of placing bounding boxes around each object of a particular category in an image. The problem definition is to determine the location of the object in an image and determine which category each object belongs to.

A traditional object detection method to find specific objects in an image involves sliding a window of different sizes over each part of the image finding all the possible positions of the objects. Each window is then classified with the end goal of finding the location of the objects. This method is computationally heavy due to a large number of candidate windows [19].

2.5.1 R-CNN, Fast R-CNN, and Faster R-CNN

Regions with CNN features, R-CNN, was proposed by Ross Girshick to solve the issue with selecting all possible regions. The number of possible regions was set to 2000, and selective search was used to extract these regions, which are called region proposals [20]. With selective search, the structure of the image is taken into account when chosen the regions [21].

The extracted region proposals are wrapped into a square and propagated into a CNN that produces a 4096-dimensional feature vector. Each region is then classified by feeding the feature vector into a class-specific linear Support Vector Machine (SVM) that does the classification task [20], see Figure 2.8.

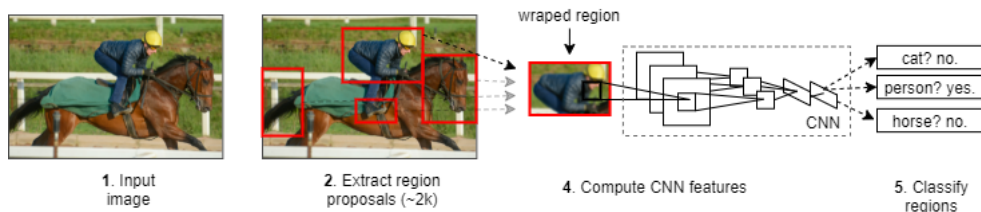


Figure 2.8: An overview of the object detection system based on [20]. (1) The R-CNN receives an image as input, (2) extracts around 2000 region proposals, (3) extracts the features for each proposal using a CNN, (4) classifies each region using an SVM.

Ross Girshick improved R-CNN and built a faster version named Faster R-CNN. The difference lies in that the input image is directly fed into a CNN to generate feature maps which are then used to identify region proposals. These region proposals are then wrapped into squares and reshaped using a RoI Pooling layer, which is similar to Max Pooling. A softmax layer is then used with the RoI feature vector to classify the region proposals and the offset values for the bounding box. What makes Fast R-CNN faster than R-CNN is that it only feeds one image to the CNN every time, instead of 2000 every time. Convolution is therefore done once per image [22].

Both R-CNN and Fast R-CNN use selective search to find the region proposals. The downside of using this procedure is that it is computationally expensive. To solve this issue Faster R-CNN was created and uses a Region Proposal Network (RPN) instead of doing a selective search. The RPN has the task of predicting the region proposals using feature maps [23].

2.6 Tools

When conducting Machine learning related experiments, there are a number of tools that can be used to fast and easily start executing these experiments, one of them is TensorFlow, and is presented below.

TensorFlow

TensorFlow is an end-to-end open-source platform for machine learning, developed and used by Google. It aids you in building and training ML models. By using Python it can provide a convenient front-end API for building applications with the framework [24].

PyTorch

PyTorch is an open-source deep learning library primarily developed by Facebook and is usually used for computer vision and natural language processing [25]. It provides Tensor computation utilizing GPU acceleration.

2.7 Related work

Most of the work regarding extracting information from billing documents has been based on extracting data from invoices. Compared with receipts, extracting information from PDF invoices does not carry the issues with handling imperfect images. For instance, blurred, damaged or folded images of billing documents. Since these documents are electronic, the textual content can be directly extracted, and OCR tools are not needed as long as the documents are not converted to images. Most of the papers mentioned in this section use F1 score to evaluate performance, which is a commonly used metric for information extraction.

CloudScan is an invoice analysis system created to learn a single global model of invoices that generalizes to unknown invoices [26]. It uses a recurrent neural network based on the Long Short Term Memory (LSTM) which is good at modeling long term dependencies. A dataset of 326,471 invoices were used to train the network in order to extract 8 different entities. The system can be seen as a NER system in which named entities, such as persons or locations are extracted from the text. Cloudscan achieved an overall F1 score of 0.891.

This model was compared to a baseline model based on logistic regression, which achieved an F1 score of 0.887.

The BERT model was used in a earlier Master's thesis in which three different machine learning models, GCN, BERT, BiLSTM, were used to retrieve 7 entities from receipts [27]. The dataset used contained 790 receipts. For comparison, a rule-based model was used. The BERT model had the highest accuracy with an F1 score of 0.455 but was outperformed by a rule-based model. The project concluded that there is potential in using natural language processing for this problem, but since BERT achieved a lower score than the rule-based model more research is needed.

A recently proposed approach is to map the text in an invoice to a grid. In the research paper, Chargrid: Towards Understanding 2D Documents [28], each invoice page is encoded as a two-dimensional grid of characters. This grid-based approach consists of a fully convolutional encoder-decoder network, where one encoder is used for semantic segmentation and two decoders are used for bounding box regression. The following fields were extracted: date, invoice number, amount, vendor name, and vendor address. The dataset consisted of 12 000 scanned invoices from a large variety of different vendors and languages. Chargrid outperformed approaches based on sequential text or document images.

The topic of information extraction from templatic documents has also been researched at Google. One of their latest contributions can be found in the paper "Representation Learning for Information Extraction from Form-like Documents" [29], where a novel approach using representation learning is introduced to automatically extract structured data from templatic documents. The solution generates extraction candidates by using the knowledge of the field types, then based on neighboring words to these candidates, a neural network learns the dens representation for each candidate. This approach achieved an F1 score of 0.878 when tested with unseen invoices. Scores for each field can be seen in Figure 2.9.

Field	F1 Score
<i>amount_due</i>	0.801
<i>delivery_date</i>	0.667
<i>due_date</i>	0.861
<i>invoice_date</i>	0.940
<i>invoice_id</i>	0.949
<i>purchase_order</i>	0.896
<i>total_amount</i>	0.858
<i>total_tax_amount</i>	0.839

Figure 2.9: F1 scores for each invoice field

In an attempt to automate the task of extracting information from PDF invoices, a study was conducted where a graph-based approach was introduced [30]. Text from PDFs was represented as graphs and their content was extracted using graph convolutional neural networks (GCN). The key items extracted were invoice number, total amount, and invoice date. The dataset used was composed of 1129 English invoices from 277 different vendors. The proposed model extracted these key items from different invoices with an F1 score of 0.875.

In a study conducted by researchers at Alibaba Group [31], a bidirectional LSTM network with a Conditional Random Field (CRF) layer (BiLSTM-CRF) was combined with Graph Embeddings to investigate if the model could attain a higher score than using BiLSTM-CRF alone to extract information from scanned invoices. Invoice Number, Vendor Name, Payer Name, and Total Amount were extracted. The dataset used in this paper consisted of 3000 user-uploaded pictures of invoices. The results showed that the BiLSTM-CRF combined with a GCN performed the best with an F1 score of 0.873 when extracting the key items.

The research paper, Data-Driven Recognition and Extraction of PDF Document Elements [32], compared object detection (Faster R-CNN) and semantic segmentation in terms of detection accuracy. This study found that an object detection-based approach yields superior results. This research paper only focused on analyzing unstructured PDF documents. In contrast, this project will handle semi-structured PDF invoices, and compare the impact that an object detection approach has on information extraction. For future work, the authors considered applying NLP approaches to extract semantic information.

Chapter 3

Methods

In this chapter, the methods, experiments, and evaluation method used will be described. As a start, the datasets, data pre-processing steps, and annotation for each approach will be presented. Lastly, the rule-based approach and the implementation of each model is described.


3.1 Dataset

The dataset in this thesis is provided by Bontouch AB employees and other volunteers. It consists of 899 PDF invoices which are in Swedish. These invoices are generated from 73 PDF invoices collected through crowdsourcing. Each of the 73 PDFs is used as a template where the entities of interest (Date, Price, Reference, Receiver) is randomly generated for each PDF. To convert each PDF to a template, the tool Adobe Acrobat DC is used to remove old entities and to place labeled fields where the new entities are being placed. A script is then executed to generate PDF invoices with new entities from invoice templates. Each new entity is randomly generated and has the same format as the original entity that was removed. The format for each field varies depending on the type of invoice, see table 3.1.

Fields	Formats
Date	The random entity generated for the due date has the international format yyyy-mm-dd.
Price	For Price, a random number that can be 1 up to 9 digit long with two decimal points at the end, as such: ".xx".
Receiver	This field can have 3 different formats depending on if the receiver type is Plusgiro, xx xx xx-x or xxxxxx-x, or Bankgiro xxx-xxxx, where x is a number.
Reference	When it comes to Reference, the format varies as well depending on if the reference type is invoice reference number (5-digit number) or OCR number (16-digit number).

Table 3.1: The format for the randomly generated entities for each field

Out of these generated PDFs, 646 are used for training, 63 for validation, and 190 are used for testing the models. This division is done randomly. An example of a PDF invoice can be seen in figure 3.1. Every PDF invoice contains the following fields which are going to be extracted: Price, Date, Receiver, and Reference.



Sida 1/1

Faktura

Datum 2020-12-15 **Fakturanummer** 4642 1530 6999 9764

Namn Namnsson
Gatuvägen 11
111 11 Stadsby
Sverige

Orderdatum: 2020-12-15
Ordernummer: 40210490
Leveransadress: Namn Namnsson, Gatuvägen 11,
111 11 Stadsby, Sverige

Artnr	Beskrivning	Antal	Å-pris (inkl. moms)	Moms	Summa
	Köp hos Apotea AB				179,00
				Nettobelopp	159,82
				MOMS (12 %)	19,18
					2903.00

Om du loggar in på klarna.se kan du se information om dina köp. Välkommen in!

Klarna Bank AB (publ) har övertagit rätten att få betalt. Klarna Bank AB (publ) har överlåtit fordran till Nordea Finans Sverige AB (publ). Betalning ska ske till Nordea Finans Sverige AB (publ) på nedanstående konto. Vid utebliven likvid debiteras påminnelseavgift fn, upp till 60,00 SEK, samt dröjsmålsränta enligt nu gällande referensränta med tillägg av 24,00 %.

Postadress Apotea AB Sveavägen 168 11346 Stockholm Sverige	Kontaktinformation Telefon: 08-7509220 E-post: info@apotea.se Hemsida: www.apotea.se	Org.nr/F-skatt 556651-6489 Momsreg.nr SE556651648901
---	--	---

PlusGiro

Vill du betala med ett klick?

Vi vill göra det så smidigt och enkelt som det bara går. Logga in på klarna.se för att betala med ett klick. Betala senast 1980-04-27

Du kan också betala med banköverföring:

Att betala: **2903.00**

Betala senast: **1980-04-27**

Till plusgiro: **46 97 62-4**

Med OCR: **6275956952909972**

INBETALNING/GIRERING CK

Till PlusGirokonto nr. 46 97 62-4	Avgift	Kod 1 Kassastämpel
Betalningsmottagare (endast namn) Klarna.		
Avsändare (namn och postadress) Namn Namnsson Gatuvägen 11 111 11 Stadsby Sverige		
Eget kontonr. vid girering		
Svenska kronor 2903 öre 00		

Meddelande till betalningsmottagare kan inte lämnas på denna blankett

#
6275956952909972 #
46 97 62-4 # 16 #

Figure 3.1: An example PDF invoice form the testset

3.2 Rule-based

The rule-based approach is created to compare the deep learning models with a trivial approach and could indicate if it is worth investing time in using these complex models. The textual content of the PDF is first extracted using the Python package *pdfplumber* [33]. Then each entity is searched and extracted using rules. Information about rules for each field can be found below.

Date

Many dates may be found in an invoice. The date that is being searched for is the due date. A list of strings that usually appear before the wanted date is therefore created to increase the likelihood of finding the correct date. For instance: "förfallodatum", "förfallodag", "oss till handa", and "betala senast". Firstly, each of these strings are search for, and if one is found in the PDF, then Date is searched for in the same row or the row underneath using regex. If more than one candidate is found, the most common is picked.

Price

Like Date, there are several strings with this format in a PDF invoice. A list of strings that usually appears before the total price is used since the field is most likely to appear close by. Each string in this list is searched using regex. If one is found, the field is searched in the same row or the row below using regex. If more than one candidate is found, the most common is picked.

Receiver

This field can have different formats depending on if the receiver type is Plusgiro or Bankgiro. Both "bankgiro" and "plusgiro" were searched for since they usually appear before the entity. If found, the field is searched in the same row or the one underneath using regex. If more than one candidate is found, the most common is picked.

Reference

Like Receiver, Reference has more than one format. It can either be the OCR number or the invoice number. Strings such as "ocr", "fakturanummer", "fakturanr", and "#" usually appear before the actual entity, and are searched.

If one is found, the field is searched in the same row and the next row using regex. If more than one candidate is found, the most common is picked.

3.3 Object detection approach


The object detection approach is based on the Faster R-CNN model which has the task of detecting and classifying bounding boxes for each field in an invoice image. However, this does not cover the process of extracting the text from each bounding box. For this reason, a rule-based extraction is implemented on top of Faster R-CNN.

3.3.1 Data Pre-processing

Since the Faster R-CNN model handles images, all PDFs are converted to png format. To ensure uniform size, all input images were scaled down and padded with whitespace to 1024 x 1024 pixels. This is done because the Faster R-CNN model in use only handles images of size 1024x1024. These pre-processing steps are done with the help of the Python package OpenCV.

3.3.2 Annotation

The data annotation is done by placing bounding boxes on top of entity fields that the model will later recognize, see figure 3.2. To make this step easier *labelImage* is used, which is a graphical annotation tool [34]. Bounding boxes are placed and one XML file is generated for each PDF, following the PASCAL VOC format.



Sida 1/1

Faktura

Datum 2020-12-15 **Fakturanummer** 4642 1530 6999 9764

Namn Namnsson
Gatuvägen 11
111 11 Stadsby
Sverige

Orderdatum: 2020-12-15
Ordernummer: 40210490
Leveransadress: Namn Namnsson, Gatuvägen 11,
111 11 Stadsby, Sverige

Artnr	Beskrivning	Antal	Å-pris (inkl. moms)	Moms	Summa
	Köp hos Apotea AB				179,00
				Nettobelopp	159,82
				MOMS (12 %)	19,18
					8888,00

Om du loggar in på klarna.se kan du se information om dina köp. Välkommen in!

Klarna Bank AB (publ) har övertagit rätten att få betalt. Klarna Bank AB (publ) har överlåtit fordran till Nordea Finans Sverige AB (publ). Betalning ska ske till Nordea Finans Sverige AB (publ) på nedanstående konto. Vid utebliven likvid debiteras påminnelseavgift fn. upp till 60,00 SEK, samt dröjsmålsränta enligt nu gällande referensränta med tillägg av 24,00 %.

Postadress Apotea AB Sveavägen 168 11346 Stockholm Sverige	Kontaktinformation Telefon: 08-7509220 E-post: info@apotea.se Hemsida: www.apotea.se	Org.nr/V-skatt 556651-6489 Momsreg.nr SE556651648901
---	--	---

PlusGirot

Vill du betala med ett klick?

Vi vill göra det så smidigt och enkelt som det bara går. Logga in på klarna.se för att betala med ett klick. Betala senast 2020-12-29

Du kan också betala med banköverföring:

Att betala: **8888,00**

Betala senast: **1990-04-29**

Till plusgiro: **36 68 41-7**

Med OCR: **9658627992414368**

INBETALNING/GIRERING CK

Till PlusGirokonto nr. 36 68 41-7	Avgift	Kod 1 Kassastämpel
Betalningsmottagare (endast namn) Klarna.		
Avsändare (namn och postadress) Namn Namnsson Gatuvägen 11 111 11 Stadsby Sverige		
Eget kontonr. vid girering		
Meddelande till betalningsmottagare kan inte lämnas på denna blankett		
Svenska kronor öre 8888 00		

9658627992414368 # **36 68 41-7** # 1 6

Figure 3.2: An example pdf invoice with bounding boxes

3.3.3 Faster R-CNN

A pre-trained Faster R-CNN model trained on COCO 2017 dataset with training images scaled to 1024x1024 is used [35]. COCO is a large-scale object detection, segmentation, and captioning dataset containing 123,287 images, 886,284 instances [36]. The model uses Resnet-101 as a backbone, which is a deep CNN with 101 layers. The training data was split and a validation set was created consisting of 10% of all the PDFs using *Roboflow* which is a tool that simplifies data preparation [37]. The model was trained for 52 epochs with a learning rate of $4 * 10^{-2}$. The model with the lowest loss was saved as the final model.

3.3.4 Rule-based extraction

The trained Faster R-CNN model detects the entity fields, classifies them correctly and returns a set of bounding boxes. Each bounding box is then cropped from the invoice image to extract its content using an optical character recognition (OCR) tool, see figure 3.3. To improve the tool's ability to detect the text, each bounding box is scaled up by 300% using OpenCV. Python-tesseract is used to recognize and "read" the text inside each bounding box. After these steps, there should be at least one candidate for each entity. Choosing one of the candidates for each entity is done by applying the same rule-based approach described in section 3.2. The most common candidate that matches the entity format is chosen as the final entity. The extraction phase depend heavily on the detection phase. The use of OCR tools to extract the text from each bounding box is not a error free process and can impact the final score.

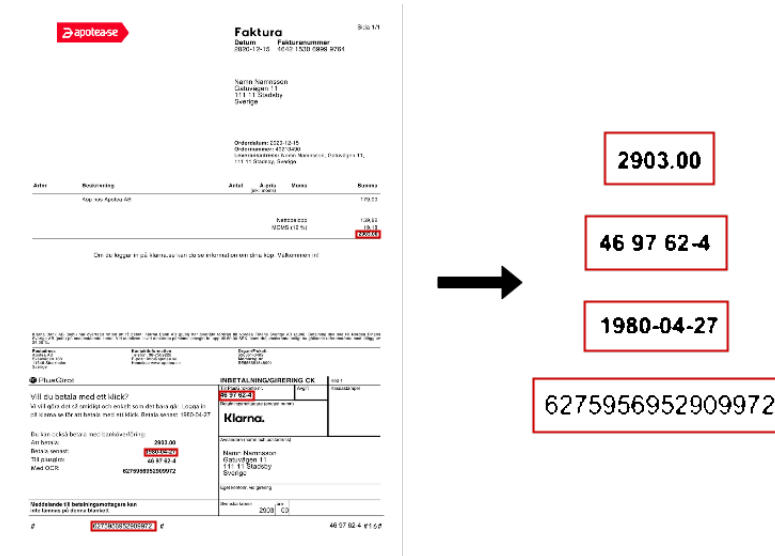


Figure 3.3: Cropped bounding boxes from a pdf invoice

3.4 BERT

In this section the data pre-processing, annotation, training and validation for the BERT model is explained.

3.4.1 Data Pre-processing

In contrast to the previous method which treated PDFs as images, BERT handles text. The textual content of each PDF is extracted using *pdfplumber* in the same way as for the rule-based approach, and saved row-wise where each row represents a sentence. The extracted text can sometimes contain words or numbers that have been divide by spaces during the extraction process, as well as sentences divided into two rows, and even distorted words. To combat this, the ground true labels for each field in each PDF are used to check if the fields exist. If a entity has been merged with another word or number, then space is added to separate it from that word.

3.4.2 Annotation

In order to feed the data to BERT, it has to be converted into CoNLL format, see figure 3.4. Each row consists of a token and a label column separated by

white space. Sentences are separated by an empty line. Each token is tagged using the IOB (inside, outside, beginning) tagging format. The B- prefix shows that the token is the beginning of a chunk, the I- prefix indicates that the token is inside a chunk. The O tag states that the token does not belong to a chunk. Since the sum class is not divided in several tokens, only the prefix B- is used for this class. The same applies to Date, and Reference. When it comes to Receiver, it can in some cases be divided in several parts placed after each other. In this case, the first token is labeled "B-receiver", and the remaining parts "I-receiver".

```
Att 0
betalä: 0
4764.00 B-sum
ange 0
OCR-nr: 0
4652713359266145 B-reference
Till 0
plusgiro: 0
98 B-receiver
02 I-receiver
09-2 I-receiver
```

Figure 3.4: CoNLL file format with tokens tagged using the IOB format

3.4.3 BERT model

A pre-trained, uncased BERT-Base model with 12 layers, 768 hidden nodes, 12 heads, and 110M parameters is implemented. This model is pre-trained on a large corpus of English data, specifically BookCorpus, which is a data set consisting of 11,038 unpublished books, and English Wikipedia [38]. By using the python library, Transformers, it takes a few amount of steps to load this pre-trained model as well as fine-tuning it. The NER data described earlier is tokenized using WordPiece. The model is fine-tuned to classify tokens to one of the four entities. The model is trained for 30 epochs with a learning rate of 10^{-5} , batch size 16, and dropout of 10^{-1} . The validation set consists of 10% of the training set. The model with the lowest validation loss is kept as the final model.

3.5 Test Setup & hardware

Google Colab is a free cloud-based platform that allows you to execute python code on your browser and is used to train both BERT and Faster R-CNN [39]. Training and testing is done utilizing the free GPU access that Google Colab provides. The GPUs available varies over time and include Nvidia K80s, T4s, P4s, P100s. The amount of memory varies over time as well but is usually 12 GB.

3.6 Evaluation method

In order to compare results and draw conclusions from the experiments, a performance metric has to be used. The metric used in this project is presented below.

3.6.1 F1 score

A performance metric is needed to evaluate the performance of each approach. F1 score, which is commonly used in related work, will be used as the performance metric. This metric depends on the precision and recall of the tests. Precision p is the ratio of the correctly identified positive cases, while Recall r is the ratio of the correctly identified positive cases from all the actual positive cases. This can be expressed using the following terms: true positives (TP) false positives (FP), true negatives (TN), and false negatives (FN), which are used in equation 3.1 and 3.2 [40].

$$p = \frac{TP}{TP + FP} \quad (3.1)$$

$$r = \frac{TP}{TP + FN} \quad (3.2)$$

F1 score is the harmonic mean of precision and recall, and is used for analyzing binary classification. See equation 3.3 for the mathematical definition of F1 score. One way of calculating the F1 score is to take the arithmetic mean of each class F1 score.

$$F_1 = 2 \times \frac{p \times r}{p + r} \quad (3.3)$$

This performance metric can be expanded to evaluate multi-class classification

by summing the TP, FP, and FN for each class to compute precision, recall and F1 score (micro averaging). An additional option is to average all the F1 scores of each class and calculate the arithmetic mean F1 score (macro averaging).

Chapter 4

Results

In this chapter, the results for the rule-based approach, NLP approach, and the object detection approach are presented. As a start, the results for each approach are presented in terms of precision, recall, and F1 score. In the end, the results from each approach are compared. Each PDF invoice contained at least one data field from each class.

4.1 Rule-Based

The final results from the rule-based model can be seen in table 4.1. The class with the highest F1 scores was Reference with a score of 0.902, followed by Price and Receiver, which had the same F1 score of 0.845. The class with the lowest F1 score is Date with a score of 0.712. The overall performance of the model is 0.830 in micro average and 0.826 in macro average. As can be seen in Table 4.1, the precision for each field equals 1. The reason for this is that the rule-based approach only classifies a prediction as correct if it strictly equals the ground truth. See section 5.2 for a discussion about this.

Class	Precision	Recall	F1
Date	1.0	0.553	0.712
Price	1.0	0.732	0.845
Receiver	1.0	0.732	0.845
Reference	1.0	0.821	0.902
Micro avg	1.0	0.709	0.830
Macro avg	1.0	0.709	0.826

Table 4.1: The results from the rule-based model.

4.2 BERT

Table 4.2 shows the results from the BERT model. The class with the highest F1 scores was Price with a score of 0.982, followed by Receiver, which had an F1 score of 0.966. The class with the lowest F1 score is Reference with a score of 0.788. The overall performance of the model is 0.911 in micro and 0.901 in macro average. The training and validation loss for each training epoch can be seen in Figure 4.1. BERT reached its lowest loss value of 0.027 on epoch 3. After that the loss value increases. The reason for the model reaching its lowest loss already at epoch 3 is discussed in section 5.2.

Class	Precision	Recall	F1
Date	0.824	0.919	0.869
Price	1.0	0.966	0.982
Receiver	0.948	0.985	0.966
Reference	0.765	0.812	0.788
Micro avg	0.892	0.930	0.911
Macro avg	0.884	0.920	0.901

Table 4.2: The results from the BERT model.

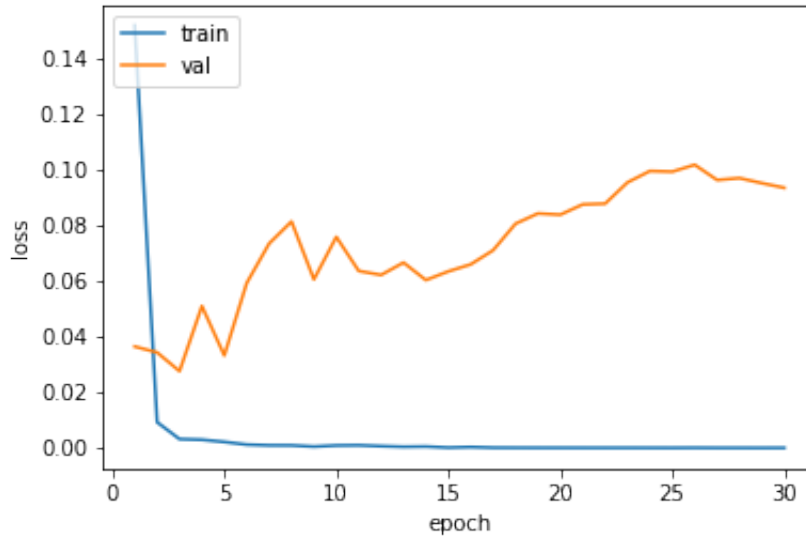


Figure 4.1: Training and validation loss for the BERT model.

4.3 Object detection

The final results from the Faster Object detection approach can be seen in table 4.3. The class with the highest F1 scores was Price with a score of 0.902, followed by Reference which had an F1 score of 0.841. The class with the lowest F1 score is Receiver with a score of 0.555. The overall performance of the model is 0.779 on both micro and macro average. As in table 4.1, table 4.3 shows that the precision for each field equals 1. The same explanation can be applied for this result.

Class	Precision	Recall	F1
Date	1.0	0.663	0.797
Price	1.0	0.889	0.942
Receiver	1.0	0.474	0.643
Reference	1.0	0.762	0.841
Micro avg	1.0	0.688	0.815
Macro avg	1.0	0.688	0.806

Table 4.3: The results from the object detection approach.

The training and validation loss for Faster R-CNN for each training epoch can be seen in Figure 4.2. Faster R-CNN reached its lowest loss of 0.250 near the end of the training phase at epoch 46.

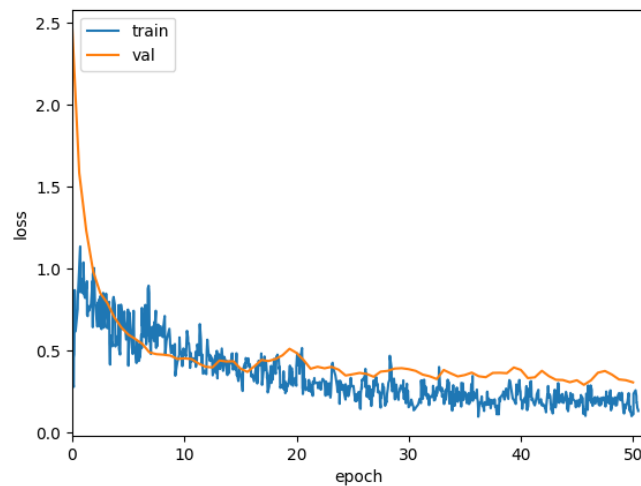


Figure 4.2: Training and validation loss for Faster R-CNN.

The Box detection precision and recall for Faster R-CNN for each epoch of the validation phase can be seen in Figure 4.3. The model reached its highest precision of 0.538 at epoch 48, and its highest recall of 0.651 at epoch 49.

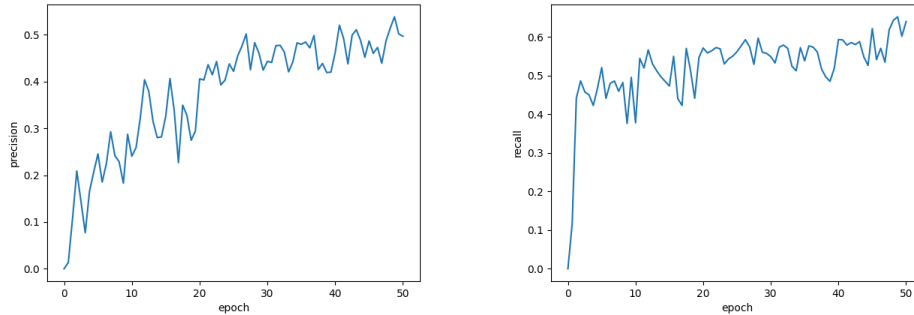


Figure 4.3: Box detection precision (on the left) and recall (on the right) for Faster R-CNN during the validation phase.

4.4 Comparing results

Table 4.2 shows the F1 scores for each class, and for each model, highlighting the highest score with bold font style. The highest score overall belongs to the BERT model which has an F1 score of 0.911 micro and 0.901 macro average. The BERT model achieved the highest score for all classes except for Reference where the highest score was achieved by the rule-based model. The model with the lowest score is the object detection approach, which had 0.815 micro average and 0.806 macro average.

Class / Model	Rule-Based	BERT	Object Detection
Date	0.712	0.919	0.797
Price	0.845	0.966	0.942
Receiver	0.845	0.985	0.643
Reference	0.902	0.812	0.841
Micro avg	0.830	0.911	0.815
Macro avg	0.826	0.901	0.806

Table 4.4: Comparison of each models final F1 scores.

The model that achieved the highest F1 score for Date was BERT, which achieved a score of 0.919. The rule-based model achieved the lowest score

of 0.712. When it comes to Price, BERT achieved the highest score of 0.966. The lowest score for this data field was 0.845 and was achieved by the rule-based model. The highest score for Receiver was 0.985 and was achieved by BERT. The object detection approach achieved a score of 0.643, which was the lowest F1 score for this data field. The rule-based model achieved an F1 score of 0.902 when evaluating Reference, which was the highest score achieved for this data field. The lowest F1 score was 0.812 and was achieved by BERT.

Chapter 5

Discussion

In this chapter, potential sources of error will be presented, and the results from the previous chapter will be discussed and analyzed. Lastly, the Societal, Ethical and Sustainability aspects of this thesis will be discussed.

5.1 Sources of error

A thesis like this contain a long list of steps to get from start to finish. Considering the time constraint, manual work and the amount of work required to get results, errors can happen and should be discussed.

To create the dataset, the original entities from each invoice are replaced with labeled fields where new random entities can be placed. Since this is a manual task, it is possible that a field may have been mislabeled or placed differently than the original entity. For instance, using the field for another class with a new class without changing the label resulting in a missing field. Additionally, an old entity can be forgotten and not deleted, so that old and new values for an entity exist in the same invoice. These scenarios could impact the results for all the models in a negative way. Making the fields too small for the new values could lead to a value being cut for being too large, which impacts the Faster R-CNN model since it converts the PDF to an image.

Some entities can be harder to extract/learn than others. Receiver can have 3 different formats and 2 names, Bankgiro, and Postgiro. However, only one of these formats is present in an invoice at a time. Similar to Receiver, Reference has two formats and two names. One is the invoice number (usually exists in every invoice), and the other is the OCR number which is not as common

and can exist in the same invoice as the first type. If both reference types are present in the same invoice, the OCR number is chosen as the correct value for this field. Having both formats in the same invoice may confuse the deep learning models in picking the invoice number instead of the OCR number and result in an incorrect prediction.

When it comes to the object detection approach, it has to first find the text corresponding to the correct field, then extract the text from the image. In some cases, the Faster R-CNN model does not manage to find the field in the image and does not move on to the OCR step. In this case, it is labeled as misclassification. Even if the field is found, the OCR tool has to correctly extract the text. Issues with OCR can also lead to misclassification.

Human error can occur in the bounding box labeling process of the Faster R-CNN training data and can negatively impact the results. For instance, if a ground truth bounding box is placed with the wrong label. Furthermore, less than 1000 training images were used to train the Faster R-CNN model. Although transfer learning was used to compensate for the small dataset, a larger dataset in combination with longer training could improve the performance.

Extracting the textual content from a PDF invoice is a process not completely absent of errors. When preparing the data for BERT, the extracted text from some PDFs contained words that were split in half multiple times (separated by spaces). Some entities were not properly aligned with the rest of the text in the same row, and were moved up or down one row, probably because of the placement of the field as explained before. Distorted entities could also be found, such as one occurrence of Date: "2021-12-as". This can impact the learning process for BERT in a negative way.

Something that distinguishes the approaches from each other is the amount of steps they require to extract the entities. For instance, the rule-based approach first searches for a word that is usually placed close to the entity in question, then searches for an entity that has the correct format and is close to that word. It is made up of two search phases. Similarly, the object detection approach also consists of two phases. In the first phase, Faster R-CNN places bounding boxes and classifies them. Since the model can return more than one candidate for each class, the next step is to pick a candidate. In contrast, the NLP approach only consists of one step, and can go through the entire text and instantly guess what class an entity belongs to. When there are several

steps that have to be fulfilled to extract information, it can end up costing the performance. In this case, it was at the expense of precision for the rule-based approach and the object detection approach, which has the value 1 as can be seen in the precision column in Table 4.1 and 4.3. The reason for this is that these approaches classify a prediction as incorrect if the value does not exactly match the ground truth, which means that there are no false positives, and precision is therefore equal to 1.

5.2 Performance analysis

The NLP approach outperformed both the rule-based approach and the object detection approach on all fields except for Reference. With a superior F1 score of 0.911, BERT showed the power that pre-trained language knowledge and understanding of context has when solving this problem. In this case, it proved to be more powerful than the spatial and visual features of the fields, which the Faster R-CNN utilizes. In second place came the rule-based approach, which achieved an F1 score of 0.830, followed by the object detection approach with an F1 score of 0.815. The reason for the rule-based approach achieving a decent score is that most fields follow structures that can be seen across most invoices. For instance, a field being placed with the same word next to or on top of it most of the time. Figure 4.1 shows that BERT overfits after epoch 3, which can be seen by the increasing validation loss and the decreasing training loss. A reason for this could be the small dataset used and BERT memorizing it. A larger training set could therefore lead to increased performance.

When it comes to the object detection approach the low score compared to the NLP approach can be explained by it not taking into account the semantics and only relying on the visual features. Due to the complexity of Faster R-CNN and its purely visual approach, it requires more training data compared to BERT. Figure 4.2 shows that the validation loss first decreases rapidly, then slows down but does not converge. Additionally, the validation loss achieved its lowest value close to the end of the training phase. Figure 4.3 show that both precision and recall are still increasing towards the end of the training phase. This indicates that longer training time in terms of epochs could also yield an increase in performance.

Compared to results from previous work, the NLP approach based on BERT attained the highest overall F1 score of 0.911. The model with an F1 score closest to the NLP approach is the invoice analysis system, CloudScan, with

an F1 score of 0.891. Comparing both models is challenging since they used different datasets. CloudScan trained with a dataset consisting of 326 471 invoices and extracted 8 entities, as opposed to BERT which was trained with 646 invoices and extracted 4 entities. Something noticeable is that both BERT and CloudScan did not take any image features into account, which shows the importance of utilizing semantic information when solving the problem of information extraction from PDF invoices.

When comparing the best-performing model with related work, approaches based on NLP achieved the best results. Only taking into consideration related work alone, CloudScan was the model with the highest F1 score. As a result, suspicions grew that the NLP approach was superior. Even if the object detection approach achieved the worst score of the 3 approaches, it didn't perform badly. The visual features utilized by this approach are therefore not worthless and could be useful to attain higher performance.

The NLP approach outperforming the rule-based approach and the object detection approach, is promising. However, it is not enough for the model to be used in production. Even by attaining an F1 score of 0.911, the NLP approach would need to be trained and tested on a larger dataset with a great variety of PDF invoices since being trained with too little data can result in poor generalization.

Date

The model that performed highest on Date was BERT. It achieved an F1 score of 0.919, followed by the object detection approach, and the rule-based approach, which achieved 0.797 and 0.712 respectively. Date can be difficult to find since there are more fields apart from the due date that has the same format in an invoice. For instance, the date when the order was placed and the date when the invoice was sent. However, there are specific words that are placed next to the field, and this field can be found in similar places. Having an understanding of language and context gave BERT the advantage and therefore attained a higher score. Faster R-CNN can utilize the spatial and other visual properties, which gave the object detection approach the edge over the rule-based approach, but by not taking into account the semantic features, it was outperformed by BERT. It is difficult to know the impact that OCR had on the results from the object detection approach.

The fact that there are multiple fields following the same format as Date makes

the search more difficult for the rule-based approach since it searches for specific formats. This may explain why it ended up in last place. However, the rule-based approach utilized the fact that certain words appear close to the field, and this helped its performance.

Price

BERT had the highest F1 score for Price, attaining a score of 0.966. In second place came the object detection approach with an F1 score of 0.942, which is not much behind BERT. The rule-based approach achieved a lower F1 score of 0.845. Similar to Date, an invoice contains multiple values that have the same format as Price, which can make it more difficult to find.

The fact that BERT has an understanding of language and context proved to make it a better fit for Price, compared to Faster R-CNN and its ability to learn spatial and visual properties. Since an invoice can contain several price values, finding the correct one with a rule-based approach proved to be more difficult since it searches for a format found in other fields.

Receiver

The model with the highest F1 score for Receiver was BERT. It Achieved a score of 0.985. In second place came the rule-based approach then the object detection approach with 0.845 and 0.643 respectively. In contrast to the previously discussed fields, Receiver can have 3 different formats depending on the type of receiver. Since only one of each type can be present in an invoice at a time, the training data for the receiver field is divided by three, which means less data to learn the receiver formats. This, combined with the small size of the training data could have impacted the score negatively. These factors impacted Faster R-CNN more since it requires a larger dataset. Since the length of this field is larger than the previous one it makes it more prone to misreadings from the OCR engine and could impact the results negatively. Similar to previously presented fields, Receiver is accompanied with words revealing the type of receiver (format) it has. Both BERT and the rule-based model take advantage of this, but BERT with its pre-trained knowledge of the language and context achieves a better score.

Reference

The rule-based approach attained an F1 score of 0.902, and was the highest score for this field. What makes this field stand out from the rest, is that it

can have two different formats and both formats could be present in the same invoice. When this happens one of the formats (OCR number) is prioritized and seen as the correct one. To avoid the scenario of picking the wrong value, the rule-based approach searches first for the OCR number then the invoice number. BERT and Faster R-CNN have a harder time knowing which formats is correct.

5.3 Societal, Ethical and Sustainability Aspects

One of the expected benefits of developing models that automate manual work is that it liberates people working with manual tasks, leaving them with more time that can be used for more important tasks. A model like BERT could be further developed and used as an internal tool for a company, or in an app where it serves its users. From the perspective of the user, it could for instance save people time by filling in forms automatically. This could give users a good impression of the company developing the app. When it comes to the worker that doesn't need to manually extract the information anymore, that person could focus on a more important task, which could benefit the company through increased productivity and higher income.

A commonly heard drawback of automation is that it sometimes leads to people losing their jobs. For instance, the person responsible for extracting information from PDF invoices might only be responsible for that task and could end up without a job. This brings up the question, is automation ethical?

When it comes to environmental sustainability, both the training and testing phase impacted the environment in a negative way. The reason for this is that training and testing a machine learning model can be a computationally heavy task. A study conducted by Åsa Moberg [41] examined the potential impacts of a change from paper invoices to electronic invoices in Sweden. This means changing 1.4 billion invoices per year from paper to electronic distribution. The results indicated that greenhouse gas emissions and energy demand could decrease. A total energy saving of around 1 400 TJ-equivalents/year and a decrease of greenhouse gas emissions by 39 000 to 41 000 ton CO_2 -equivalents/year could be achieved. The main potential negative impact of transitioning to only using electronic invoices was due to the electricity used by servers.

When it comes to information extraction from PDF invoices, making the

process faster and more efficient could lead to people preferring this technology and stop using invoices printed on paper that contain chemicals and requires sacrificing trees that clean the air, to produce the paper.

The data used in this project was collected through crowdsourcing where workers from the host company voluntarily send PDF invoices. Personal information in the invoices was replaced with made-up information. The rest of the information existing in the invoice is public information.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this thesis, an object detection approach was compared with a natural language processing approach, and a rule-based approach, in terms of information extraction. The object detection approach was based on Faster R-CNN and rule-based extraction, and the NLP approach was based on BERT. The fields that were extracted are Date, Price, Receiver, and Reference. The results showed that the approach with the best performance was the approach based on BERT. It attained a micro average F1 score of 0.911. In second place came the rule-based approach achieving 0.830 in micro average F1 score, followed by the object detection approach with the micro average of 0.815. The NLP approach achieved the highest F1 score on all fields except for Reference, where the rule-based approach performed the best with a F1 score of 0.902. The reason for this was that Reference could have two different formats in the same invoice, which confused the machine learning models. The reason the object detection approach is believed to perform the worst is that it needed more data and longer training.

To conclude this thesis, the NLP approach proved to be better at extracting information from PDF invoice than the object detection approach, and showed the advantage that BERT has by understanding language and context. Although the object detection approach attained the lowest F1 score, it has the potential to achieve higher performance in the future if more data is provided. Since the NLP approach outperformed the rule-based approach, it is concluded that a deep learning approach is better suited for the task of extracting payment information from PDF invoices.

6.2 Future work and improvements

Future work can be based on several aspects of the thesis. Since the OCR engine is an important tool that the extraction depends on in the object detection approach, it would be interesting to know how the OCR error impacted the final score and find a way to fix the issue or replace it with another tool.

Before labeling the fields for the data used by BERT, the entire text from each PDF was extracted using a python library as described in section 3.4.1. In some cases, words from the extracted text became unintentionally split into several parts, and even merged with other words from the text. No clear explanation could be found and this happened to the same invoices every time. Since this issue appeared before the annotation process, it's possible that a field affected by this could have been left without being annotated. Investigating how this impacted the final score could also be part of future work. Another option would be to replace the used python library with another one or manually correct the mistakes in the extracted text. It is an important problem to solve since the text extraction library is not only used for extracting text for data annotation, but also used for extracting text for inference. A relevant question to ask is, given the mistakes by the extraction library, how much manual correction is needed, or if it is worth using a deep learning approach given the fact that it sometimes needs manual correction. For companies interested in taking this approach for information extraction, it is also relevant to research the time and cost of making corrections.

Since the Faster R-CNN model requires a large dataset, expanding the dataset as well as increasing the training time could boost performance. Adding more PDF invoices to the dataset could also solve the problem of overfitting for BERT. More data can be generated through data augmentation. Furthermore, hyperparameter optimization can be done to increase performance.

Labeling the training data for Faster R-CNN in another way that utilizes the fact that many fields often appear close to each other. In this case, the invoice could be divided into sections that are labeled instead of the current approach that labels each field individually. Since BERT achieved the best performance overall, an interesting experiment would be to combine it with the object detection approach. In that case, if an invoice is labeled and detected section-wise with Faster R-CNN, then BERT could extract the fields from each section.

Bibliography

- [1] Wikipedia, “Introduction to machine learning.” [Online]. Available: <http://www.datascienceassn.org/sites/default/files/Introduction%20to%20Machine%20Learning.pdf>
- [2] A. Ławrynowicz and V. Tresp, *Introducing Machine Learning*, 01 2014, vol. 18, pp. 35–50.
- [3] S. Marsland, *Machine Learning: An Algorithmic Perspective, Second Edition*, 2nd ed. Chapman Hall/CRC, 2014. ISBN 1466583282
- [4] Z. Zhang, “A gentle introduction to artificial neural networks,” *Annals of Translational Medicine*, vol. 4, pp. 370–370, 10 2016. doi: 10.21037/atm.2016.06.20
- [5] F. Rosenblatt, “The perceptron: a probabilistic model for information storage and organization in the brain.” *Psychological review*, vol. 65 6, pp. 386–408, 1958.
- [6] M. Sazli, “A brief review of feed-forward neural networks,” *Communications, Faculty Of Science, University of Ankara*, vol. 50, pp. 11–17, 01 2006. doi: 10.1501/0003168
- [7] G. Dematos, M. Boyd, B. Kermanshahi, N. Kohzadi, and I. Kaastra, “Feedforward versus recurrent neural networks for forecasting monthly japanese yen exchange rates,” *Financial Engineering and the Japanese Markets*, vol. 3, pp. 59–75, 1996.
- [8] L. Torrey and J. Shavlik, “Transfer learning,” *Handbook of Research on Machine Learning Applications*, 01 2009. doi: 10.4018/978-1-60566-766-9.ch011
- [9] A. Wiki, “A beginner’s guide to neural networks and deep learning.” [Online]. Available: <https://wiki.pathmind.com/neural-network>

- [10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016. ISBN 0262035618
- [11] J. Eisenstein, *Introduction to Natural Language Processing*, ser. Adaptive Computation and Machine Learning series. MIT Press, 2019. ISBN 9780262042840. [Online]. Available: <https://books.google.com/books?id=72yuDwAAQBAJ>
- [12] I. M. Konkol, “Named entity recognition,” Ph.D. dissertation, 2015.
- [13] I. Sutskever, O. Vinyals, and Q. Le, “Sequence to sequence learning with neural networks,” *Advances in Neural Information Processing Systems*, vol. 4, 09 2014.
- [14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>
- [15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *NAACL-HLT*, 2019.
- [16] J. Brownlee, “How do convolutional layers work in deep learning neural networks?” [Online]. Available: <https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/>
- [17] N. Hirschkind, S. Mollick, J. Pari, and J. Khlm, “Convolutional neural network.” [Online]. Available: <https://brilliant.org/wiki/convolutional-neural-network/>
- [18] Standford, “Cs231n convolutional neural networks for visual recognition.” [Online]. Available: <https://cs231n.github.io/convolutional-networks/>
- [19] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, “Object detection with deep learning: A review,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, pp. 1–21, 01 2019. doi: 10.1109/TNNLS.2018.2876865

- [20] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 11 2013. doi: 10.1109/CVPR.2014.81
- [21] J. Uijlings, K. Sande, T. Gevers, and A. Smeulders, “Selective search for object recognition,” *International Journal of Computer Vision*, vol. 104, pp. 154–171, 09 2013. doi: 10.1007/s11263-013-0620-5
- [22] R. Girshick, “Fast r-cnn,” 4 2015. doi: 10.1109/ICCV.2015.169
- [23] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, 06 2015. doi: 10.1109/TPAMI.2016.2577031
- [24] TensorFlow, “An end-to-end open source machine learning platform.” [Online]. Available: <https://www.tensorflow.org>
- [25] Pytorch, “From research to production.” [Online]. Available: <https://pytorch.org/>
- [26] B. P. Rasmus, W. Ole, and L. Florian, “Cloudscan - a configuration-free invoice analysis system using recurrent neural networks,” Nov. 2017. doi: 10.1109/ICDAR.2017.74. [Online]. Available: <https://arxiv.org/pdf/1708.07403.pdf>
- [27] L. Marko, “Using natural language processing to extract information from receipt text,” Aug. 2020. [Online]. Available: <http://kth.diva-portal.org/smash/get/diva2:1458900/FULLTEXT01.pdf>
- [28] A. Katti, C. Reisswig, C. Guder, S. Brarda, S. Bickel, J. Höhne, and J. Faddoul, “Chargrid: Towards understanding 2d documents,” 09 2018.
- [29] B. Majumder, N. Potti, S. Tata, J. B. Wendt, Q. Zhao, and M. Najork, “Representation learning for information extraction from form-like documents,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, 2020, pp. 6495–6504.
- [30] F. Krieger, P. Drews, B. Funk, and T. Wobbe, “Information extraction from invoices: A graph neural network approach for datasets with high layout variety,” 03 2021.

- [31] X. Liu, F. Gao, Q. Zhang, and H. Zhao, “Graph convolution for multimodal information extraction from visually rich documents,” 01 2019. doi: 10.18653/v1/N19-2005 pp. 32–39.
- [32] H. Matthias, P. André, E. Kemal, and M. Tobias, “Data-driven recognition and extraction of pdf document elements,” Nov. 2017. doi: 10.3390/technologies7030065. [Online]. Available: <https://www.mdpi.com/2227-7080/7/3/65/htm>
- [33] jsvine, “pdfplumber.” [Online]. Available: <https://github.com/jsvine/pdfplumber>
- [34] tzutalin, “Labelimg.” [Online]. Available: <https://github.com/tzutalin/labelImg>
- [35] T. Hub, “Faster r-cnn with resnet-101 v1 object detection model.” [Online]. Available: https://tfhub.dev/tensorflow/faster_rcnn/resnet101_v1_1024x1024/1
- [36] COCO, “What is coco?” [Online]. Available: <https://cocodataset.org/#home>
- [37] Roboflow, “Everything you need to start building computer vision into your applications.” [Online]. Available: <https://roboflow.com/>
- [38] Huggingface, “bert-base-uncased.” [Online]. Available: <https://huggingface.co/bert-base-uncased>
- [39] Google, “What is colab?” [Online]. Available: <https://colab.research.google.com/notebooks/intro.ipynb>
- [40] Wikipedia, “F-score.” [Online]. Available: <https://en.wikipedia.org/wiki/F-score>
- [41] M. Åsa, B. Clara, G. Finnveden, and S. Tyskeng, “Effects of a total change from paper invoicing to electronic invoicing in sweden,” Oct. 2010. [Online]. Available: <https://kth.diva-portal.org/smash/get/diva2:355958/FULLTEXT01.pdf>

TRITA -EECS-EX-2021:632