Universitat Autònoma de Barcelona

Neural Information Extraction from Semi-structured Documents

Manuel Carbonell

Universitat Autònoma de Barcelona

Neural Information Extraction from Semi-structured Documents

A dissertation submitted by **Manuel Carbonell** at Universitat Autònoma de Barcelona to fulfil the degree of **Doctor of Philosophy**.

Bellaterra, November 19, 2020

Directors:

Alicia Fornés
Autonomous University of Barcelona
Computer Science Dept. & Computer Vision Center

Mauricio Villegas
omni:us - Qidenus Group GmbH
Berlin, Germany

Josep Lladós
Autonomous University of Barcelona
Computer Science Dept. & Computer Vision Center

Thesis Committee

Dr. Joan Andreu Sánchez
Polytechnic University of Valencia
Valencia, Spain
Dr. Dimosthenis Karatzas
Autonomous University of Barcelona & Computer Vision Center
Barcelona, Spain
Dr. Anjan Dutta
University of Exeter
Exeter, England

This document was typeset by the author using LaTeX 2$_{\varepsilon}$.

The research described in this book was carried out at the Computer Vision Center, Universitat Autònoma de Barcelona with the support of omni:us, Qidenus Group GmbH.

ISBN ISBN “Pending”

Printed by Ediciones Gráficas Rey, S.L.

Dedicated to my parents

Acknowledgements

This work would have not been possible if it weren't for my advisors Mauricio Villegas, Alicia Fornés and Josep LLadós. Thank you for your constant feedback, as well as transfer of knowledge and passion for research, which made it worth it to choose this path. Special thanks to Joan Mas and Pau Riba for their insights and for helping me to continue advancing with my work whenever it got more difficult to do so. I am also very grateful for the great atmosphere found at the CVC. Lei, David, Ali, Andres, Sounak, Sanket, Ruben, Mohamed and so many others, coming every day to this kind of monastery we found ourselves into for exchanging great philosophical as well as random nonsense ideas was a very fun experience thanks to all of you. Thanks to my friends Adri, Ana, Marta, James, Pau and David for being there helping me to leave the cave and disconnect whenever I needed to. Last but not least, I want to thank my family. Reyes, Quim and Caterina, thank you for your unconditional support which allowed me to complete this exciting but at the same time hard journey, and for having made my life everyday a little bit better.

Abstract

Sectors as fintech, legaltech or insurance process an inflow of million of forms, invoices, id documents, claims or similar every day. The success in the automation of these transactions depends on the ability to correctly digitize the textual content as well as to incorporate semantic understanding. This procedure, known as information extraction (IE) comprises the steps of localizing and recognizing text, identifying named entities contained in it and optionally finding relationships among its elements. In this work we explore multi-task neural models at image and graph level to solve all steps in a unified way. While doing so we find benefits and limitations of these end-to-end approaches in comparison with sequential separate methods.

Resum

Sectors com la informació i tecnologia d'assegurances, finances i legal, processen un continu de factures, justificants, reclamacions o similar diàriament. L'èxit en l'automatització d'aquestes transaccions es basa en l'habilitat de digitalitzar correctament el contingut textual així com incorporar la comprensió semàntica. Aquest procés, conegut com Extracció d'Informació (EI) consisteix en diversos passos que són, el reconeixement de el text, la identificació d'entitats nomenades i en ocasions en reconèixer relacions entre aquestes entitats. En el nostre treball vam explorar models neurals multi-tasca a nivell d'imatge i de graf per solucionar els passos d'aquest procés de forma unificada. En el camí, vam estudiar els beneficis i inconvenients d'aquests enfocaments en comparació amb mètodes que resolen les tasques seqüencialment per separat.

Resumen

Sectores como la información y tecnología de seguros, finanzas y legal, procesan un continuo de facturas, justificantes, reclamaciones o similar diariamente. El éxito en la automatización de estas transacciones se basa en la habilidad de digitalizar correctamente el contenido textual asi como incorporar la comprensión semántica. Este proceso, conococido como Extracción de Información (EI) consiste en varios pasos que son, el reconocimiento del texto, la identificación de entidades nombradas y en ocasiones en reconocer relaciones entre estas entidades. En nuestro trabajo exploramos modelos neurales multi-tarea a nivel de imagen y de grafo para solucionar los pasos de este proceso de forma unificada. En el camino, estudiamos los beneficios e inconvenientes de estos enfoques en comparación con métodos que resuelven las tareas secuencialmente por separado.

Contents

List of Tables

List of Figures

LIST OF FIGURES

Chapter 1

Introduction

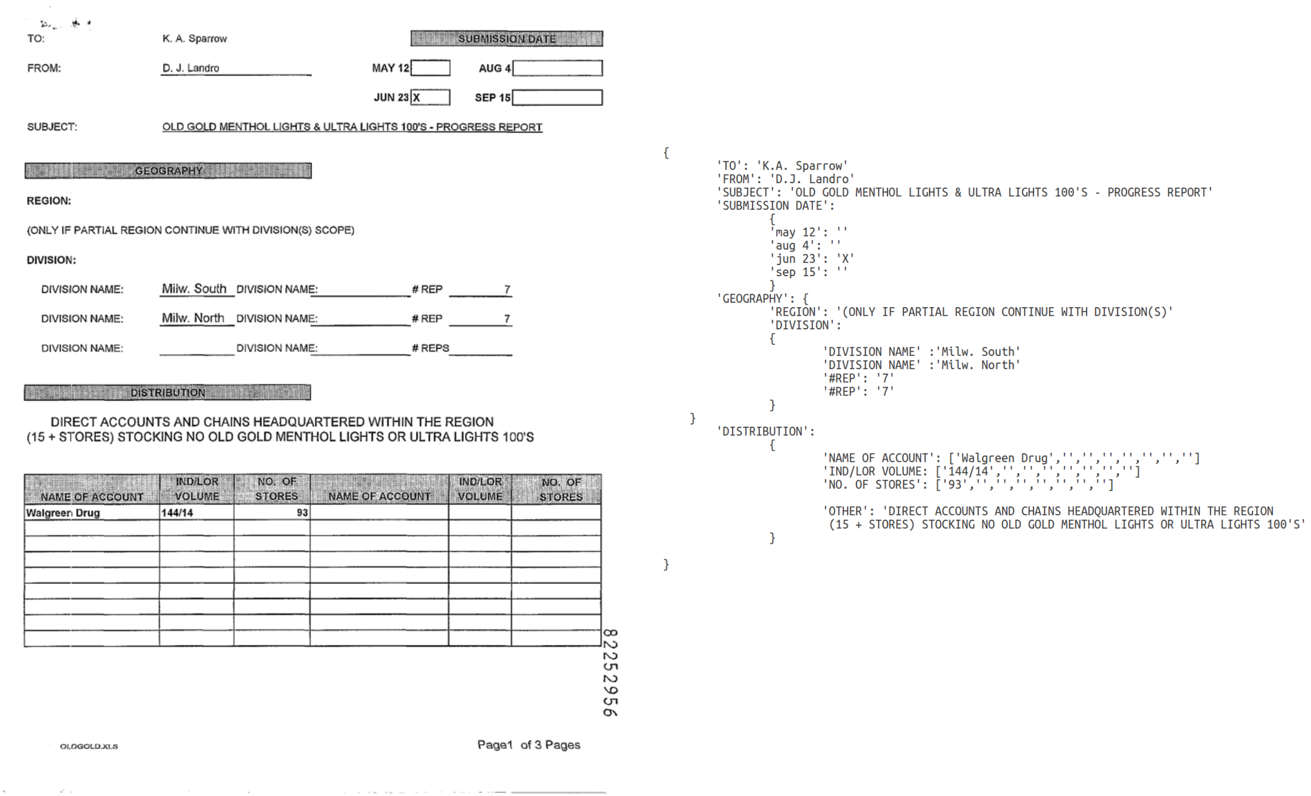
1.1 Information extraction from semi structured documents

As long as humans make use of documents to communicate and leave record of bureaucratic information without a universally defined structure, there will be benefits in using methods that learn to understand these documents and extract their contained information in an automated way. The process of converting a document image or digitized document into a labeled collection of values meaningfully related is known as Information Extraction (IE). This process comprises the steps of localizing and recognizing text, identify named entities contained in it and relationships among them. Due to the increasing existence of software solutions for all kinds of mail room applications, every day there is naturally less amount of printed documents generated where the first two steps of this process are required. Also, many cases in which automated IE from documents is currently used, it is due to incomplete database integration between departments of companies or administration. For example, a person asking for a scholarship might need to give address details in a form to the regional government despite that the national government is already aware of this information. In such case it would be a much better solution to simply ask for the person's id and permission to access this information, but in many cases the databases are not connected for no other reason other than lack of software development. This type of situations will tend to happen in lesser cases as time advances, reducing the amount of documents to be processed. But still, there will always be situations in which this data integration is simply not possible. For example, a person might apply for a given type of position in several different companies, and upload the CV to the recruiting tool of each company, highlighting desired professional experience to be available. Each company might want to have different information from the CV stored in their database, such as the candidate's major, the last professional experience and its time range, or personal information such as birth place and address. Being so, either does the company read and understand the CV to manually extract the desired values to introduce in the database or the candidate will manually fill the form with the required information. There is no way in which this information was already accessible from an integrated database, either because of the constantly varying nature of the data or for privacy reasons. In the case of insurance there is constant transfer of information in which one party (e.g. the insured client) has a default format to store that information (e.g. an invoice) and the party that needs to store the certain values (e.g. the insurance) receives them in very different ways from each client. In this type of situations there are definitely going to be vast amount of documents generated making it very useful any kind of improvement in the methods for automated IE.

An example of document to be processed and the corresponding target structured information to be extracted can be found in figure 1.1. As shown in the example a reasonable structure extracted from the given document should take in account text properties such as font size, indentation, colons, tabs, and tabular elements which define a hierarchical relation and grouping of textual elements.

As it is well known in the industry a big problem when trying to massively extract data from documents is the continuously varying format, which prevents to use template alignment methods with usable results. But this is not the only issue, other challenges that make difficult to automate the IE process include the presence of tables, charts, floating images, handwritten text, etc. These barriers are reflected in companies in terms of high operational costs and completion time, as well as low process efficiency and accuracy. The amount of difficulties and the need to automate the process of extracting information varies depending on the type of document. A good classification of the types of printed documents from which automated IE is helpful can be found in [32].

Among these, due to the industry-tied nature of this work we chosen to use forms. Examples of forms in which we apply proposed methods are industry registers as the ones seen in figure 1.2. The document can be stored in many different ways. Most commonly and in the recent years it consists of a digital pdf, but there is also abundance of scanned images as well as mobile phone camera captured in the context of insurance documents.



TO: K. A. Sparrow

SUBMISSION DATE

FROM: D. J. Landro

MAY 12 □ AUG 4 □

JUN 23 ☒ SEP 15 □

SUBJECT: OLD GOLD MENTHOL LIGHTS & ULTRA LIGHTS 100'S - PROGRESS REPORT

GEOGRAPHY

REGION:

(ONLY IF PARTIAL REGION CONTINUE WITH DIVISION(S) SCOPE)

DIVISION:

DIVISION NAME: Milw. South DIVISION NAME: ________ # REP 7

DIVISION NAME: Milw. North DIVISION NAME: ________ # REP 7

DIVISION NAME: ________ DIVISION NAME: ________ # REPS ________

DISTRIBUTION

DIRECT ACCOUNTS AND CHAINS HEADQUARTERED WITHIN THE REGION
(15 + STORES) STOCKING NO OLD GOLD MENTHOL LIGHTS OR ULTRA LIGHTS 100'S

NAME OF ACCOUNT	IND/LOR VOLUME	NO. OF STORES	NAME OF ACCOUNT	IND/LOR VOLUME	NO. OF STORES
Walgreen Drug	144/14	93			

82252956

OLDGOLD.XLS

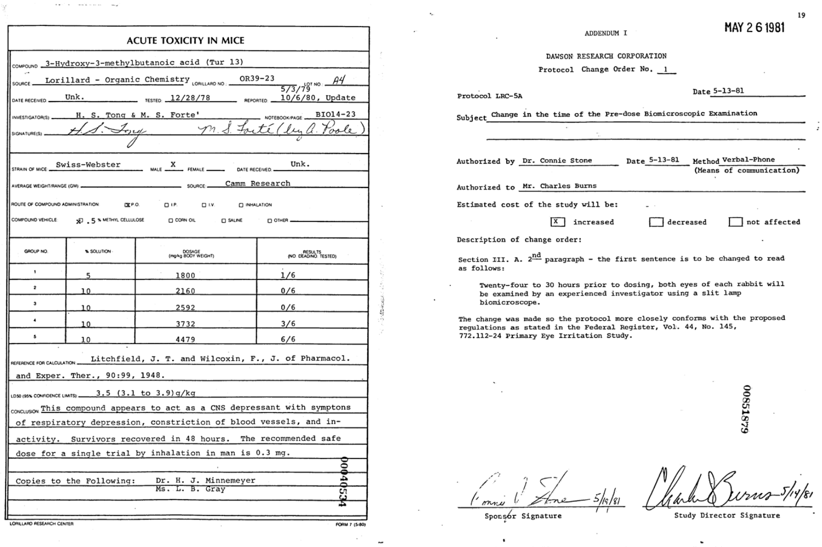
Page1 of 3 Pages

```
{
        'TO': 'K.A. Sparrow'
        'FROM': 'D.J. Landro'
        'SUBJECT': 'OLD GOLD MENTHOL LIGHTS & ULTRA LIGHTS 100'S - PROGRESS REPORT'
        'SUBMISSION DATE':
                {
                'may 12': ''
                'aug 4': ''
                'jun 23': 'X'
                'sep 15': ''
                }
        'GEOGRAPHY': {
                'REGION': '(ONLY IF PARTIAL REGION CONTINUE WITH DIVISION(S)'
                'DIVISION':
                {
                        'DIVISION NAME' :'Milw. South'
                        'DIVISION NAME' :'Milw. North'
                        '#REP': '7'
                        '#REP': '7'
                }
    }
        'DISTRIBUTION':
                {
                        'NAME OF ACCOUNT': ['Walgreen Drug','','','','','','','']
                        'IND/LOR VOLUME: ['144/14','','','','','','','']
                        'NO. OF STORES': ['93','','','','','','','']

                        'OTHER': 'DIRECT ACCOUNTS AND CHAINS HEADQUARTERED WITHIN THE REGION
                         (15 + STORES) STOCKING NO OLD GOLD MENTHOL LIGHTS OR ULTRA LIGHTS 100'S'
                }

}
```

Figure 1.1: An example scanned form from which IE is necessary.



ACUTE TOXICITY IN MICE

COMPOUND 3-Hydroxy-3-methylbutanoic acid (Tur 13)

SOURCE Lorillard - Organic Chemistry LORILLARD NO. OR39-23 LOT NO. A4

DATE RECEIVED Unk. TESTED 12/28/78 REPORTED 5/3/79 10/6/80, Update

INVESTIGATOR(S) H. S. Tong & M. S. Forte' NOTEBOOK/PAGE BIO14-23

SIGNATURE(S) H. S. Tong M. S. Forte (by A. Poole)

STRAIN OF MICE Swiss-Webster MALE X FEMALE DATE RECEIVED Unk.

AVERAGE WEIGHT/RANGE (GM) SOURCE Camm Research

ROUTE OF COMPOUND ADMINISTRATION ☒ P.O. □ I.P. □ I.V. □ INHALATION

COMPOUND VEHICLE ☒ .5 % METHYL CELLULOSE □ CORN OIL □ SALINE □ OTHER

GROUP NO.	% SOLUTION	DOSAGE (mg/kg BODY WEIGHT)	RESULTS (NO. DEAD/NO. TESTED)
1	5	1800	1/6
2	10	2160	0/6
3	10	2592	0/6
4	10	3732	3/6
5	10	4479	6/6

REFERENCE FOR CALCULATION Litchfield, J. T. and Wilcoxin, F., J. of Pharmacol. and Exper. Ther., 90:99, 1948.

LD50 (95% CONFIDENCE LIMITS) 3.5 (3.1 to 3.9)g/kg

CONCLUSION This compound appears to act as a CNS depressant with symptons of respiratory depression, constriction of blood vessels, and inactivity. Survivors recovered in 48 hours. The recommended safe dose for a single trial by inhalation in man is 0.3 mg.

Copies to the Following: Dr. H. J. Minnemeyer
Ms. L. B. Gray

00040534

LORILLARD RESEARCH CENTER FORM 7 (5/80)

19

MAY 26 1981

ADDENDUM I

DAWSON RESEARCH CORPORATION
Protocol Change Order No. 1

Protocol LRC-5A Date 5-13-81

Subject Change in the time of the Pre-dose Biomicroscopic Examination

Authorized by Dr. Connie Stone Date 5-13-81 Method Verbal-Phone
(Means of communication)

Authorized to Mr. Charles Burns

Estimated cost of the study will be:

☒ increased □ decreased □ not affected

Description of change order:

Section III. A. 2$^{\text{nd}}$ paragraph - the first sentence is to be changed to read as follows:

> Twenty-four to 30 hours prior to dosing, both eyes of each rabbit will be examined by an experienced investigator using a slit lamp biomicroscope.

The change was made so the protocol more closely conforms with the proposed regulations as stated in the Federal Register, Vol. 44, No. 145, 772.112-24 Primary Eye Irritation Study.

00851879

Connie Stone 5/19/81 Charles Burns 5/14/81
Sponsor Signature Study Director Signature

Figure 1.2: Forms from the RVL-CDIP database.

Another type of documents that also demand IE automatizing are historical manuscripts including birth, death or marriage records. For this work we used a database of marriage records due to its exhaustive openly available ground truth for the main tasks of IE. The additional difficulty in the case of historical documents is the step of recognizing handwritten text in multiple different styles, in which commercial OCR engines usually struggle to give a good performance. Several different methods have been explored to automate the tasks involved in IE. Until very recently a dominating approach in the industry for IE solutions consisted of template alignment derived methods [71] or other rule-based systems as it is reflected in the survey [13].

1.2 Neural Networks Breakthrough

As it happened in many other application scenarios of Machine Learning (ML), best performing proposed methods for IE tasks shifted from variations of algorithms such as Support Vector Machines (SVMs) [9], Hidden Markov Models (HMMs) or bayesian networks [87, 69, 65] to the currently ubiquitous Deep Neural Networks (DNN) also referred to as Deep Learning (DL) models.

One of the greatest strengths of DNNs is that in contrast with the previously used methods in which features to find patterns on the data had to be hand crafted, neural networks pioneered in showing the capacity to spontaneously learn representations that discriminate and characterize properly elements contained in each example of a given dataset, as it was shown in the case of small images in [46]. This property was also applied with great success in other fields such as automatic speech recognition (ASR), translation [22] or object detection [24].

The term *end-to-end* has been used in very different domains, including cryptography, data corruption prevention or networking design. In general it refers to a process that acts as a single module which includes all sub tasks necessary to complete it in a unified way, receiving the input and directly giving the final result instead of intermediate outputs. In the context of DL it has become increasingly popular and considered a beneficial property of newly presented methods, nevertheless, it has been used with slightly different definitions in recently presented works.

In this thesis we understand that a neural model is end-to-end if the input data is fed a single time to the model and this one directly gives the final result, while using the error of all tasks to simultaneously calculate some parameters shared for all sub-tasks, instead of sequentially modifying the input data and feeding it to smaller separate models for each step. This trend of creating increasingly larger models to solve more complex tasks, despite being intuitively beneficial for the advance in the path to emulate human intelligence needs a sound analysis in the case of deep neural networks to ensure that it is reasonable to go for its use, and that it is the cause of reported improvements when proposing new architectures. In this work we focus on exploring benefits and limitations of such approaches in the context of information extraction.

1.3 Motivation and research questions

The motivation behind this thesis is to get closer to a robust and generic solution for automated IE from semi-structured documents, from Mail-Room applications until historical ones. Based on the current existing work, a reasonable approach for this purpose is to study how good does the unification of tasks in end-to-end architectures perform to solve the whole process of IE, in real use case data. Also, due to the success of Graph Neural Networks in many domains where arbitrary relationships and patterns among the data can be found, it makes sense to explore their effectiveness in the context of named entity recognition and relation extraction.

Our research questions are the following:

- Is it beneficial to share features of neural end-to-end models for subsequent tasks in the context of IE from semi structured documents?
- How does the context affect in the categorization and relation extraction of semi structured document textual elements?
- To which extent can GNNs be a suitable approach when recognizing entities

and finding relationships among them in semi structured documents?

Motivated by these questions the research goals are the following

- Implement end-to-end models for task pairs in each of the steps of IE.
- Explore the benefits and limitations of these models in front of previous existing sequential approaches.
- Explore the use promising architectures that succeeded in other fields of pattern recognition such as Graph Neural Networks for the most challenging tasks of IE.

1.4 Contributions

The contributions of this thesis are the following:

- A method for end-to-end recognition of text and named entities, while showing that the neural model is capable to learn both visual as well as semantic features when reading documents.
- A method for localizing and recognizing text in full handwritten pages that can handle intermediate step error by performing recognition in pooled features instead of the input image.
- A method to combine the tasks of text localization transcription and named entity recognition in a unified neural model, together with an exhaustive study of benefits and limitations in different scenarios.
- A method for recognizing entities and extracting relations in scanned forms with Graph Neural Networks, with state of the art performance in a form understanding benchmark.

1.5 Outline

This thesis contains six chapters which are listed below.

In the next chapter we mention some relevant existing work that constitutes the context of this thesis, going from former popular ML methods for document processing, to different branches of Neural Networks used in different steps of IE from documents.

In the third chapter we explain the first proposed method to combine the processes of recognizing text and named entities in weakly structured documents in which the data is organized in a sequential manner. We do this by applying a multi-modal use of the CTC function.

In chapter four we propose a model that combines localization and recognition of handwritten text in full pages while digging down into sequential task error. The qualitative results show benefits in joining these two tasks in an end-to-end model in contrast to separate approaches.

In chapter 5 we close the IE task cycle by proposing a simultaneously trained triple task model for localization, transcription and named entity recognition. Being so, we explore benefits and limitations of the proposed approach and give results in different use cases.

In the sixth chapter we investigate the use of GNN for named entity recognition and relation extraction, while achieving successful results and state of the art performance in a scanned form understanding benchmark for the relation extraction task.

Finally, in the last chapter we extract the conclusions of this thesis and propose possible continuation lines.

Chapter 2

Related work

In this chapter we outline the some methodologies that have preceded and are related to different parts of our work. These include relevant contributions of deep neural network variations and other machine learning methods as their description, and some main applications in the field of information extraction from semi-structured documents, considered in an image, layout and semantic scenario.

2.1 Deep Learning

As introduced before, the dominating approach for almost all of the pattern recognition tasks consists of the DL family of algorithms, which constitutes the ground of this thesis. In this section we briefly introduce the principles of this type of algorithms and some of its variations, such as perceptrons, convolutional neural networks and recurrent neural networks.

2.1.1 Perceptrons and convolutional neural networks

In this section we give brief introduction to Deep Learning models [48] as they constitute the starting ground of this work. Let $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$ a dataset containing n input examples $\mathcal{X} = \{x_1, ..., x_n\}$, $x_i \in \mathbb{R}^s$ corresponding to n target values $\mathcal{Y} = \{y_1, ..., y_n\} \in \mathbb{R}^t$

A Deep Neural Network is a function $f_\theta : \mathbb{R}^s \longrightarrow \mathbb{R}^t$ which can learn some parameters θ to approximate a function f^* that maps values from $\mathcal{X}$ to their corresponding targets $\mathcal{Y}$. The approximating function f_θ consists of a composition of functions, $f_\theta = f_1 \circ f_2 \circ \cdots \circ f_{L-1} \circ f_L$, which are usually called *layers*, the first one input layer and the L-th one output layer. The fact that f_θ is composed by a sequence of function compositions is the reason why they are called *deep* neural networks. The term *neural* comes from the idea that the output values z of functions f_i in the composition sequence emulate animal neurons which transmit a sequence synapses. The calculation of an output $f_\theta(x) = z_L$ is known as $feedforward$ step. A typical example of layer is linear transformation, i.e. a learnable weight matrix product with the layer input

$$f_l(z_{l-1}) = Wz_{l-1}$$

where z_{l-1} is the output of previous layer f_{l-1}. Another commonly used layer which gives the name to a whole family of DNNs are convolutional layers

$$f_l(z_{l-1}) = \sum_{i,j\in\{0,...,Z_l\}} \sum_{u,v\in\{0,...,H_l\}} w_{uv} {z_{l-1}}_{i+u,j+v}$$

where ${z_l}_{i,j}$ are the real values of layer l output, Z_l is the vector size of layer l output, H_l is the convolutional weight matrix size for layer l and w_{jk} are the values of the weight matrix. These layers are applied depending on the type of pattern that needs to be identified. When it is assumed that there can be dependencies between any of the input elements then fully connected layers (equivalently linear transformations) are used. When there is the assumption of prevalent local relationships as it is for convolutions, which work very effectively on image data. Neural networks that include this latter type of layer are called Convolutional Neural Networks (CNNs).

To approximate the model parameters θ the commonly used method is Stochastic Gradient Descent (SGD). This method consists of iteratively calculating the gradient of the loss function C and subtract it from each weight matrix of f_θ. That is, if $C(f_\theta, y)$ is a cost function that determines how much error is committed by the network f_θ approximating f^*, then at each SGD step $t \in \{1, ..., T\}$ the gradient $\nabla C(f_\theta, y)_t$ is calculated using *backpropagation* algorithm from the last layer of the network. Then the weights at time step $t+1$ are updated with the previously calculated gradient

$$\theta_{t+1} = \theta_t - \alpha \nabla C(f_{\theta_t}, y)_t$$

An exhaustive description with details of how this is done can be found in [25].

2.1.2 Recurrent Neural Networks

Together with CNN architectures, a type of neural networks which also shown to achieve great performance in a wide variety of problems are recurrent neural networks (RNNs). The intuition behind building such model relies on the idea that the mind has a certain 'state' that continuously gets updated based on a sequence of inputs, which can be sounds, images, characters etc. In this way, the above described feedforward operation is repeated a certain number of time steps updating a *state* h_t value which emulates the 'memory' of that network and giving a different output result at each time step $t \in [1, ..., T]$

$$h^{(t)} = f_\theta(h^{(t-1)}, x^{(t)})$$

To calculate the gradient for RNNs, the network is unfolded in the time dimension and error is calculated and backpropagated for all time steps in a similar way as in feedforward networks.

An experimental limitation found with first proposed RNNs was that for longer time sequences the dependencies seemed hard to be taken in account by the model. To fix this the variant of RNN named *Long Short-Term Memory* networks were proposed and increased substantially the performance on many sequential tasks. The idea behind LSTMs is to enhance RNNs with additional 'cells' (learnable weight vectors) that emulate the process of adding new information to the memory, updating it or forgetting it. This is modeled with the following equations. The forget gate controls the self loop update

$$f^{(t)} = \sigma\left(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)}\right)$$

where $x^{(t)}$ is the current step input vector, $h^{(t)}$ is the current hidden layer output and b^f,U^f and W^f are biases, input weights and recurrent weights for the forget gates. The internal LSTM state is also calculated with a combination of linear transformations with sigmoid activation as follows

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + g_i^(t)\sigma\left(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} h_j^{(t-1)}\right)$$

being b,U and W biases and weights of the LSTM cell. In a similar way the input and output gates are calculated using previous input and hidden states

$$g_i^{(t)} = \sigma\left(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)}\right)$$

$$q_i^{(t)} = \sigma\left(b_i^o + \sum_j U_{i,j}^o x_j^{(t)} + \sum_j W_{i,j}^o h_j^{(t-1)}\right)$$

and finally the LSTM cell output is calculated as

$$h_i^{(t)} = \tanh(s_i^{(t)})q_i^{(t)}$$

. LSTMs shown to give better performance in tasks that involve finding long range dependencies, such as predicting the next word in a sentence, in which it can be necessary to be aware of the subject's genre given a few words earlier.

2.2 End-to-end vs separate methods

As introduced earlier a rising trend in DL literature is to join models to form end-to-end architectures with stacked layers that give the output of a complex procedure instead of sequentially use separate step modules. A remarkable example of this idea is YOLO [66]. Where earlier approaches such as [16] used separate blocks to first tile a window over the image and scan the existence of certain features in each location sequentially identifying the content to be detected, YOLO uses a single CNN to directly infer all the contents in the image, implicitly solving the intermediate task of scanning each location, with impressive performance. If we look at ASR, state of the art evolved in a similar way. Until the beginning of 21st century handcrafted feature-based methods such as Mel Frequency Cepstral Coefficients (MFCC) worked as a feature extraction from the audio spectrogram, which combined with a HMM allowed to predict phonemes. This would later be replaced by end-to-end neural methods that achieved state of the art performance on larger pieces of raw audio without any preprocessing [29]. However, recent attempts of solving much more complex tasks such as autonomous driving in an end-to-end fashion highlighted several limitations of the approach and concluded that further research is required so that such solutions can be used in a real-world scenario [14]. This situation contrasts with market-ready solutions which make use of modules that are separately being improved due to the difficulty of each of the tasks involved in the main process, e.g. [12] despite having a modularized solution working in the real world. These observations leave the path open to explore whether the unification of IE tasks in end-to-end models will bring more benefits or drawbacks when looking for a market-ready solution.

2.3 Transfer learning

The main prestige behind deep neural network relies on their capability of learning necessary features to discriminate the key properties of training examples for classification or regression, in contrast with earlier used methods which needed of handcrafted features. One of the most commonly reported weaknesses of this family of methods is the need of large amounts of annotated data to achieve usable results in almost any problem, which sometimes can be costly to produce. This is usually solved with the use of transfer learning, by training the network for a similar problem for which we do have large amount of annotated data, and then fine-tune the model using learned features for the new task [91]. Regardless of their limitations, deep neural networks constitute the main research path of artificial intelligence and its related fields, including information extraction.

2.4 Machine learning and information extraction

Before the beginning of the past decade methods in the academia for IE were variate within a wide range of ML techniques other than DL. In this section we mention some of them and several works that exploit them for IE or document processing related tasks. A method that stood out for its simplicity and effectiveness was Bag of Words (BoW) [77]. This NLP derived model consists in gathering a representation of each image with handcrafted features such as Scale-Invariant Feature Transform (SIFT) vectors [56] to later form 'visual words' as combinations of these features that allow classification or clustering of unseen examples. Classification can be obtained with algorithms such as k-means [58], k-Nearest Neighbors (k-NN) [15] or SVMs. An example of usage of BoW combined with 2-nearest neighbors for categorizing documents by logo spotting can be found in [72]. When looking at cases in which local relationships within an example bring meaningful patterns to identify and classify them, CRFs constituted a solid approach to treat such type of problem. A conditional random field is an undirected graphical model whose nodes can be divided into exactly two disjoint sets which are the observed and output variables X and Y respectively. Then the conditional distribution $p(Y|X)$ is modeled. When the graph is defined as sequence of hidden states fulfilling the Markov property it is known as a Hidden Markov Model. As a usage example for IE, in [34] a CRF model is used to extract structures in printed newspaper documents. In the case of HMMs their arrival in the field of Handwritten Text Recognition (HTR) was the dominating approach for a long period of time despite their initial moderate success [47]. In some cases successful results were achieved with the use of user interaction to correct machine predictions [82].

The main drawback of these methods is their limited performance, which in most cases prevented that they were brought to production systems that automate tasks solving them with better or comparable results as human performance. Later on the majority of these methods used by the whole document analysis community where to be outperformed by DL techniques, as in many other ML application domains.

2.5 Neural network based information extraction

Coming up next we mention some methods that make use of DL in different steps of the IE process.

The most common first step when receiving a document is to classify it in one of the predefined categories such as the ones mentioned above. This allows later application of the domain knowledge to extract the information in a structured way. We specify that it is the most common first step since in some cases methods can integrate this part with more generic models. To do this the widely chosen approach is a simple CNN classifier from the document image. A most refined version of this with great performance is recently presented in [20]. Still, other works shows that combination of Image data with processed layout and text data helped achieve a better final performance [90].

The task of separating the document into different sections and types of contents such as paragraphs, tables, lists, equations, figures etc. is known as layout analysis. The capacity of CNN architectures to locally identify patterns in images that gives top performance in object detection tasks [50] also happened to be successful in this step of IE, as shown in [49], leaving it as the currently used method for generic document layout analysis.

A necessary yet not trivial step to correctly extract information from documents in case they are stored in images is to localize and recognize the text contained in it and convert it to a sequence of machine readable characters. This is specially notable in the scenario of handwritten documents, or hybrid documents in which we find a combination of printed and handwritten text. Great progress has been achieved recently thanks to the use of Deep Neural Architectures. For the localization part most methods rely on state of the art object detection architectures, i.e. variations of RetinaNet [51]. For the recognition part, the Connectionist Temporal Classification loss which gave great performance in Automated Speech Recognition (ASR) also gave great success for recognizing handwritten when combining it with Recurrent Neural Networks (RNNs) [27].

Wether we start from digital pdf documents or the previously mentioned tasks have been solved, a crucial part of IE, is to identify and classify relevant entities in the document to ease access when later browsing a given database. This step is known as Named Entity Recognition (NER). The great majority of work in this field is explored for large corpuses of natural language. A very popular benchmark is the CoNLL2003 task [79], in which relevant entities consisting words or groups of few words, are labeled with classes such as location, organization or person. The state of the art for this task as in almost every NLP related task is a variation of the Transformer [85], in this case the method consists two 'tower' blocks of self attention mechanisms that share input textual embeddings [2].

When it comes to find relationships among elements in a document the situation is similar as in NER, most of the existing work is done on plain corpus or sentences instead of business or historical documents in which there is some kind of underlying structure. Notable examples exploiting the patterns in the language to find relationships are [52, 41]. This work leaves open a question of how could a neural model work finding relationships among elements in a semi structured document in which layout plays a relevant role.

If we focus on the case of IE from invoice documents, one of the a possible approach to use is Cloudscan [62]. In their work, they extract the text with a commercial OCR system and use an LSTMs model fed with text n-grams as well as localization values as features to classify each element in the document and parse it with rule based system. An overview can be seen in figure 2.1. Due to

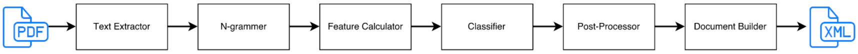


Figure 2.1: An overview of the approach proposed in Cloudscan, extracted from Palm et al [62].

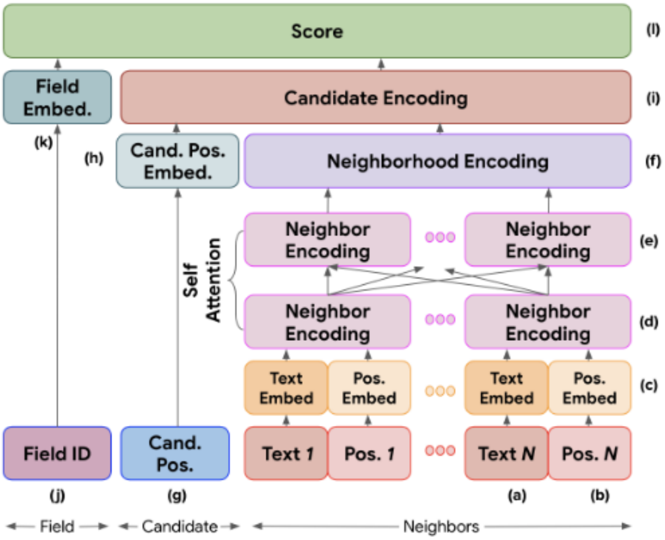


Figure 2.2: The Representation Learning for IE model, extracted from [61].

the limited performance of the LSTM cell as a memory unit they later improved this idea in [61] applying the Neural Turing Machine concept [30], to extend the model with external memory and selectively attending to the document inputs to compare it with the contents in the memory and correctly parse them. Later on, the encoder-decoder architecture which gave great results on Neural Machine Translation (NMT) [3] was applied on the extracted 2d grid of characters from invoice documents [43]. Analogously to networks in which the input is a matrix whose elements represent color pixel intensities, in this case the matrix elements represent existence of a character in a given document position. Then after the input layer a series of stacked convolutions encode the document information in a hidden state vector which is fed to two decoder networks, one extracts the semantic segmentation and the other the bounding box regression.

The reported results claim that this grid-like approach outperforms purely sequential models, specially in fields where 2D text entities are important. This idea is further extended with the application of the BERT model [18], instead of the seq2seq, serializing the document grid and in a similar way, extracting the

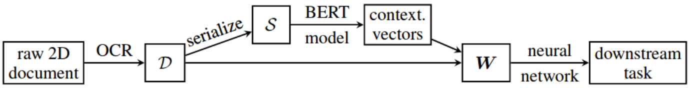


Figure 2.3: The BERTGrid invoice processing pipeline extracted from [17].

required information to define the so called BERTGrid model [17]. In this work the performance overcomes previous approaches which is attributed among other reasons to the success in dealing with out of vocabulary words. An overview of this approach can be seen in figure 2.3.

In [80] a LSTM model is combined with a CNN to sequentially annotate images of words exploiting the contextual information, achieving state of the art on the solved task and leaving the path open for improve such approach to use it in a segmentation-free scenario.

In the context of form-like documents, in a similar way as in [61] very recently another neural approach generates candidates to fill fields in a previously known target schema scoring each candidate by means of a model inspired in the Transformer [85]. An overview of the approach can be seen in figure 2.2.

As it can be observed the main trend in the previously mentioned works consists of applying architectures that give state of the art performance in more generic NLP tasks that do not necessarily involve understanding a structure in a given document and adapt them to different IE scenarios. This leaves open the path to explore how each of the tasks interacts with each other in the whole IE pipeline as well as the other previously stated questions.

Chapter 3

Joint transcription and named entity recognition

When extracting information from handwritten documents, text transcription and named entity recognition are usually faced as separate subsequent tasks. This has the disadvantage that errors in the first module affect heavily the performance of the second module. In this chapter we propose to do both tasks jointly, using a single neural network with a common architecture used for plain text recognition. Experimentally, the work has been tested on a collection of historical marriage records. Results of experiments are presented to show the effect on the performance for different configurations: different ways of encoding the information, doing or not transfer learning and processing at text line or multi-line region level. The results are comparable to state of the art reported in the ICDAR 2017 Information Extraction competition, even though the proposed technique does not use any dictionaries, language modeling or post processing.

3.1 Introduction

As introduced in the previous chapter, the best existing methods for each of the steps of IE consist of DNN-based architectures which, in the case of scanned document images, include the processes of recognizing segmented text and annotate named entities. Also we have seen that neural architectures tend to grow into end-to-end models that solve several tasks in a unified way that usually would be faced with separated modules, tackling this way the intermediate error accumulation problem. Motivated by these observations, in this chapter we propose a method to study the interplay between the text recognition and semantic segmentation tasks by combining the visual-textual and semantic information in the

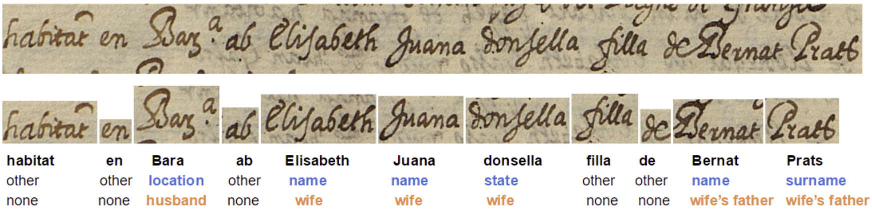


Figure 3.1: An example of a document line annotation from [73].

Table 3.1: Semantic and person categories in the IEHHR competition

Semantic	**Person**
Name	Wife
Surname	Husband
Occupation	Wife's father
Location	Wife's Mother
Civil State	Husband's father
Other	Husband's mother
	Other person
	None

ground truth. Results show that the model is able to internally learn both the visual features required to differentiate the characters in the text as well as the semantic ones to produce the named entity annotations.

The rest of the chapter is organized as follows: Next section explains the task being considered. In 3.3 we explain our model architecture, ground truth setup and training details. In Section 3.4 we analyze the results for the different configurations and last in 3.5 we give the conclusions.

3.2 Information extraction in marriage records

The approach presented in this chapter is evaluated on the task of information extraction in a system for the analysis of population records, in particular handwritten marriage records. It consists of transcribing the text and assigning a semantic and person category to each word, i.e. to know which kind of word has been transcribed (name, surname, location, etc.) and to what person it refers to. The dataset and evaluation protocol are exactly the same as the one proposed in the ICDAR 2017 Information Extraction from Historical Handwritten Records (IEHHR) competition [73]. The semantic and person categories to identify in the IEHHR competition are listed in table 3.1.

Two tracks were proposed in the competition. In the basic track the goal is

Table 3.2: Marriage Records dataset distribution

	Train	Validation	Test
Pages	90	10	25
Records	872	96	253
Lines	2759	311	757
Words	28346	3155	8026
Out of vocabulary words: 5.57 %			

to assign the semantic class to each word, whereas in the complete track it is also necessary to identify the person. An example of both tracks is shown in Figure 3.1.

The dataset for this competition contains 125 pages with 1221 marriage records (paragraphs), where each record contains several text lines giving information of the wife, husband and their parents' names, occupations, locations and civil states. The text images are provided at word and line level, naturally having the increased difficulty of word segmentation when choosing to work with line images. More details of the dataset can be found in table 3.2.

3.3 Methodology

The main goal of this work is to explore a single end-to-end trainable DNN model that receives as input text images and gives as output transcripts, already labeled with their corresponding semantic information. One possibility to solve it could be to propose a DNN with two sequence outputs, one for the transcript and the other for the semantic labels. However, keeping an alignment between these two independent outputs complicates a solution. An alternative is to have a single sequence output that combines the transcript and semantic information, which is the approach taken here. There are several ways in which this information can be encoded such that a model learns to predict it. The next subsection describes the different ways of encoding it that were tested in this work. Then there are subsections describing the architecture chosen for the neural network, the image input and characteristics of the learning.

3.3.1 Semantic encoding

The first method variation which we studied is the way in which ground truth transcript and semantic labels are encoded so that the model learns to predict them. To allow the model to recognize words not observed during training (out-of-vocabulary) the symbols that the model learns are the individual characters and a space to identify separation between words. For the semantic labels special tags are added to the list of symbols for the recognizer. The different possibilities are explained below.

Open & close separate tags

In the first approach, the words are enclosed between **opening and closing tags** that encode the semantic information. Both the category and the person have independent tags. Thus, each word is encoded by starting with opening category and person symbols, followed by a symbol for each character and ends by closing person and category symbols. The "other" and "none" semantics are not encoded. For example, the ground truth of the image shown in Figure 3.1 would be encoded as:

```
h a b i t a t {space} e n {space} <location> <husband> B a r a
</husband> </location> {space} a b {space} <name> <wife> E l i s a
b e t h </wife> </name> ...
```

This kind of encoding is not expected to perform well in the IEHHR task, since tags are assigned to only one word at a time, so it is redundant to have two tags for each word. However, in other tasks it could make sense having opening and closing tags and this is why it has been considered in this work.

Single separate tags

Similar to the previous approach, in this case both category and person tags are independent symbols but there is only one for each word added before the word. Thus, the ground truth of the previous example would be encoded as:

```
h a b i t a t {space} e n {space} <location/> <husband/> B a r a
{space} a b {space} <name/> <wife/> E l i s a b e t h {space} J u a
n a {space} <state/> <wife/> {space} d o n s e l l a ...
```

Change of person tag

In this variation of the semantic encoding the person label is only given if there is a **change of person**, i.e. the person label indicates that all the upcoming words refer to that person until another person label comes, in contrast to previous approaches where we give the person label for each word. This approach is possible due to the structured form of the sentences in the dataset. As we can see in Figure

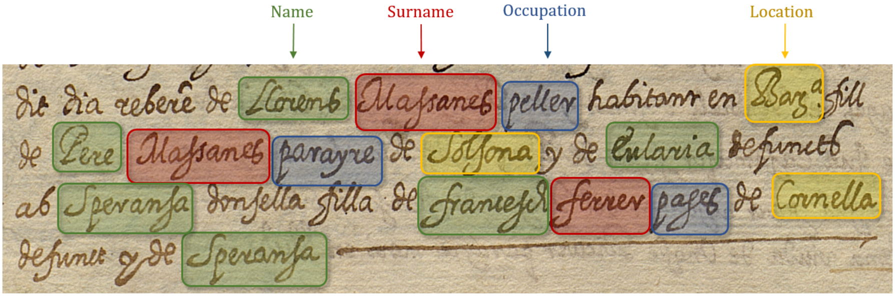

Name
Surname
Occupation
Location

Figure 3.2: Reading the whole record makes it easier to transcribe as well as to identify the semantic categories based on context information.

3.2 the marriage records give the information of all the family members without mixing them.

```
<wife/> <name/> E l i s a b e t h {space} <name/> J u a n a {space}
                   <state/> d o n s e l l a ...
```

Single combined tags

The final possibility tested for encoding the named entity information is to **combine category and person** labels into a single tag. So the example would be as:

```
   h a b i t a t {space} e n {space} <location_husband/> B a r a
    {space} a b {space} <name_wife/> E l i s a b e t h {space}
  <name_wife/> J u a n a {space} <state_wife/> d o n s e l l a ...
```

3.3.2 Level of input images: lines or records

The IEHHR competition dataset includes manually segmented images at word level. But to lower ground truthing cost or avoid needing a word segmentator, we will assume that only images at line level are available. Having text line images then the obvious approach is to give the system individual line images for recognition. However, there are semantic labels that would be very difficult to predict if only a single line image is observed due to lack of context. For example, it might be hard to know if the name of a person corresponds to the husband or the father of the wife if the full record is not given. Because of this, in the experiments we have explored having as input both text line images and full marriage record images, concatenating all the lines of a record one after the other.

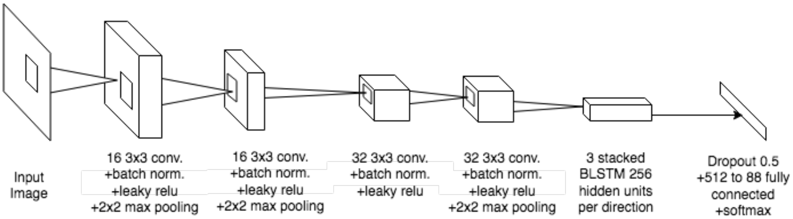

Input Image
16 3x3 conv. +batch norm. +leaky relu +2x2 max pooling
16 3x3 conv. +batch norm. +leaky relu +2x2 max pooling
32 3x3 conv. +batch norm. +leaky relu
32 3x3 conv. +batch norm. +leaky relu +2x2 max pooling
3 stacked BLSTM 256 hidden units per direction
Dropout 0.5 +512 to 88 fully connected +softmax

Figure 3.3: Used model architecture

3.3.3 Transfer learning

The next variable we examined was the effect of the use of transfer learning from a previously trained model for HTR. Transfer Learning consists of training for the same or a similar task (HTR) using other datasets, and then fine tune it for the current purpose, in this case HTR+NER. To perform transfer learning from a generic HTR model, the softmax layer is removed and replaced with a softmax that allows as an output the activations for the number of possible classes in the fine tuning step. In this case, they will be all the characters in the alphabet plus the semantic labels. In the experiments for transfer learning we have tested only one HTR model that was trained with the following datasets: IAM [84], Bentham [74], Bozen [75], and some datasets used by us internally: IntoThePast, Wiensanktulrich, Wienvotivkirche and ITS.

3.3.4 Curriculum learning

The last variation that we propose is curriculum learning i.e. start with easier demands to the model and then increase the difficulty. In this case this method can be interpreted as starting by learning to transcribe single text lines, and when the training is finished, continue with learning to transcribe images of a whole marriage record.

3.3.5 Model architecture and training

In this work we use a CNN+BLSTM+CTC model, which is one of the most common models for performing HTR exclusively, although other HTR models could be used as well. In particular, the architecture consists of 4 convolutional layers with max pooling followed by 3 stacked BLSTM layers. The detailed model architecture is shown in Figure 3.3.

To train the model we use the Laia HTR toolkit [64] which uses Baidu's parallel CTC [28] implementation, which consists of minimizing the loss or "objective" function

$$O^{ML}(S,\mathcal{N}_w) = -\sum_{(x,z)\in S} ln(p(z|x)) \tag{3.1}$$

where S is the training set, x is the input sequence (visual features), z is the sequence labeling (transcription) for x and

$$\mathcal{N}_w : (\mathbb{R}^m)^T \mapsto (\mathbb{R}^n)^T \tag{3.2}$$

is a recurrent neural network with m inputs, n outputs and weight vector w. The probabilities of a labeling of an input sequence are calculated with a dynamic programming algorithm called "forward-backward".

Some special features of the model are that the activation function for the convolutional layers is leaky ReLu $f(x) = x$ if $x > 0.01,\ 0.01x$ otherwise.

We also use batch normalization due to its proven performance increase in CNN architectures.

3.4 Results

We compare the performance of the proposed methods[1] with the results of the participants of the IEHHR competition in [73] thereby using the same metric, see Table 3.3. The evaluation metric counts the words that were correctly transcribed and annotated with their category and person label with respect to the total amount of words in the ground truth. For those words that were not correctly transcribed but the category and person labels match one or more words in the ground truth, we add to the score 1 - CER (character error rate) on the best matching word. This means that the named entity recognition part is vital for a good score, since a perfect transcription will count as 0 in the score if its named entity is incorrectly detected.

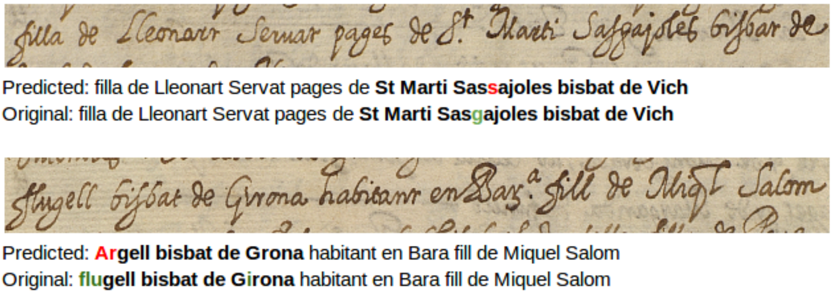
We can observe in the results that the best performance is reached when receiving the whole marriage record, which is probably due to the help of contextual information. For example, it can benefit the detection of named entities composed of several words when they are written in separate consecutive lines. Also we observe that the best performing encoding of the semantic labels is the combined tags setup. This can be due to the lower amount of symbols to predict, which might require to store less long term dependencies in the network.

The most significant improvement was achieved when picking the best performing configuration and running it with an alternative line extraction. In the competition, the text lines were extracted by including all the bounding boxes of the words within every line. As a result, when there are large ascenders and descenders, the bounding box of the line is too wide, including sections of other

[1] Scripts used for the experiments available at `http://doi.org/10.5281/zenodo.1174113`

Method	**Proc. Level**	**Track Basic**	**Track Complete**
IEHHR competition results			
Baseline GMM+HMM	Register	80.24	63.08
Hitsz-ICRC-1 CNN HTR+NER	Word	87.56	85.72
Hitsz-ICRC-2 ResNet HTR+NER	Word	94.16	91.97
CITlab-ARGUS-1 LSTM+CTC+regex	Line	89.53	89.16
CITlab-ARGUS-2 LSTM+CTC+OOV+regex	Line	**91.93**	**91.56**
Results of the experiments			
Separate-open tags	Line	74.04	64.77
Separate-open-close tags	Line	82.04	73.73
Separate-open tags + transfer learn.	Line	88.47	82.99
Separate-open-close tags + transfer learn.	Line	65.91	55.77
Change person tags + transfer learn.	Reg.	84.41	80.51
Combined-open tags + transfer learn. + curriculum learn.	Reg.	**90.32**	**89.08**

Table 3.3: Average scores of the experiments compared with the IEHHR competition participants' methods.



Predicted: filla de Lleonart Servat pages de **St Marti Sassajoles bisbat de Vich**
Original: filla de Lleonart Servat pages de **St Marti Sasgajoles bisbat de Vich**

Predicted: **Argell bisbat de Grona** habitant en Bara fill de Miquel Salom
Original: **flugell bisbat de Girona** habitant en Bara fill de Miquel Salom

Figure 3.4: Some of the errors committed in the predictions

text lines. In order to cope with this limitation, we used the XML containing the exact location of the segmented words within a page, and for the y-coordinates, we used a weighted (by the words widths) average of upper and lower limits of the word bounding boxes. As expected, the performance highly improves because the segmentation of the text lines is more accurate. However, this result is not directly comparable to the other participants's methods because the segmentation is different.

In Figure 3.4 we show some examples of committed errors. We can see that they consist of small typos that are understandable when looking at the text images. It is definitely difficult to transcribe certain names that have never been seen before. The proposed approach could be combined with a category-based language model [68] which could potentially improve the results.

Our best performing model took 4 hours 38 to run 133 training epochs with a NVIDIA GTX 1080 GPU. As training configuration we used an adversarial regularizer [26] with weight 0.5, an initial learning rate of $5 \cdot 10^{-4}$ with decay factor of 0.99 per epoch and batch size 6.

3.5 Conclusion

In this chapter we have proposed the first contribution of the thesis. We have proposed a method to solve a complex task (i.e. text recognition and named entity recognition) with a single end-to-end neural model. The first conclusion is that, in information extraction problems, a generic model for solving two subsequent tasks can perform at least similarly as two separated models. This is true even if there is less prepared data (record level images instead of a sequence of word images) and we do not make use of task specific tools like dictionaries or language model.

By investigating different ways of encoding the image transcripts and semantic labels we have shown that the recognition performance is highly affected, even though it is indeed representing the same information. Also, curriculum learning (first text lines and then records) can make the model reach a higher final prediction accuracy.

Continuation of this work includes to explore the effect of text localization and its interaction with the recognition and tagging named entities, as we study in the upcoming chapters. Also, a possible extension is to evaluate the method in other datasets.

In [88] continuation of this work has been done, by applying this idea to more cases in which CTC can help predict different types of labels sequentially, such as Chinese scripts or music sheet recognition.

Chapter 4

Joint text localization and transcription

When transcribing handwritten document images, inaccuracies in the text segmentation step often cause errors in the subsequent transcription step. For this reason, some recent methods propose to perform the recognition at paragraph level. But still, errors in the segmentation of paragraphs can affect the transcription performance. In this work, we propose an end-to-end framework to transcribe full pages. The joint text detection and transcription allows to remove the layout analysis requirement at test time. The experimental results show that our approach can achieve comparable results to models that assume segmented paragraphs, and suggest that joining the two tasks brings an improvement over doing the two tasks separately.

4.1 Introduction

As we introduced before, the performance of handwritten text recognition (HTR) methods has significantly improved with the arrival of deep convolutional network architectures and attention models [45], [6]. Nevertheless, when transcribing document images, layout analysis (i.e. word, line or paragraph segmentation) is a required previous task that usually supposes a source of error [31],[1]. Traditionally, methods for transcription of handwritten documents rely on the output of some post-processing steps to obtain the different segmented objects, i.e., lines or words depending on the level of recognition that the method works [81, 42]. It is for sure known that the performance of such methods is conditioned by the correctness of the output from the segmentation step. In the other way around, to provide a good segmentation it would be beneficial to have the transcription of the word. This dilemma is defined by the well-known Sayre's paradox: a good segmentation is necessary for a good recognition and vice-versa.

Many HTR methods perform a joint segmentation and recognition at line level to cope with the above mentioned paradox. In this way, they can avoid the segmentation at character and word level. However, this is only partially solving the segmentation problem, because lines that are not properly segmented obviously affect the recognition at line level. For this reason, some recent approaches propose to recognize text at paragraph level [6], [63]. But still, an inaccurate segmentation into paragraphs will affect the HTR performance.

If we put the view into another domain such as the detection of text in the wild, where we encounter text in cluttered images, the task is usually divided into first localizing the text, and then recongizing the detected region [38]. Text localization, which can be faced as an object detection problem, has been divided into two main type of paradigms, one-stage and two-stage. In [83] a two stage method is proposed by first generating a sparse set of candidate proposals followed by a second stage that classifies the proposals into different classes and background. Regions with CNN features (R-CNN) [24] replace the second stage with a CNN, improving the previous methods. The next big improvement in terms of performance and speed came with Faster-RCNN [24], where the concept of *anchors* was introduced. When prioritizing speed in front of accuracy, we find one stage detection as the best option. Concretely, SSD [53] and YOLO [66] have put one-stage methods close to two-stage in precision but having much greater speed performance. The decrease in precision of one-stage against two-stage methods is due to the class imbalance in [51], so focal loss is introduced to cope with this problem and achieve state of the art performance both in accuracy and speed.

Recent work in scene text detection [55] [10] [57] claimed that a single end-to-end model to jointly localize and transcribe words can reduce intermediate step error thereby leading to better detection results, and consequently, better transcriptions. The intuition behind this phenomena would be that giving transcription annotations gives additional valuable information to the detection model.

One may think that this principle applies in handwritten documents as well. In the few last years some works have appeared in the domain of the document transcription following an idea similar to the end-to-end models. However, in [89] this statement is put under doubt, since the best transcription performance is achieved by detecting the start of text line, segmenting it with a line follower and then transcribing it with three separately trained networks. Nevertheless no results are shown regarding the end-to-end trained model approach to come to a definitive conclusion.

In [60] a method to join the two tasks is proposed by predicting the text line beginning and letting the recognition network predict the characters till the end of the line without having an explicit end of line segmentation. The drawback of this method is that it does not attempt to backpropagate the recognition errors

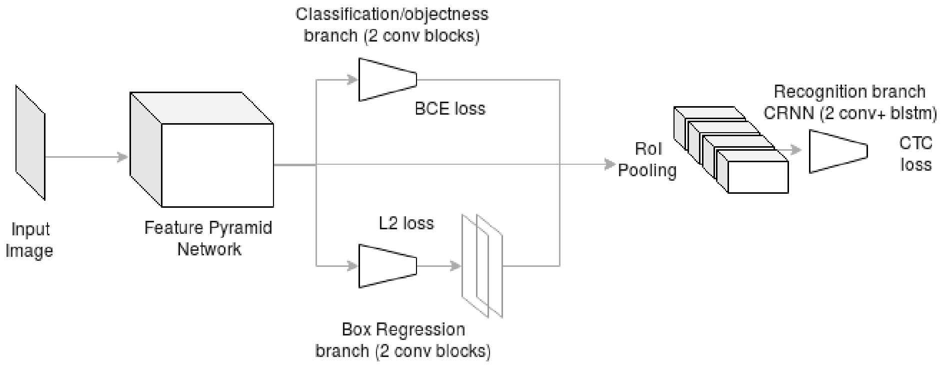


Figure 4.1: Overview of the proposed method. We extract convolutional features using FPN. The classification and regression branches calculate the positive boxes and the recognition branch predicts the transcription of the content of each box. Binary cross entropy, squared-sum and CTC losses are backpropagated through the whole model.

to the segmentation which might make it difficult to achieve a high performance on difficult benchmark datasets. In [5] the results of transcribing full paragraphs by implicitly segmenting lines with attention are comparable with the traditional automatic line segmentation methods, but the lack of a comparison using the state of the art neural segmentation separate system followed by the CRNN architecture prevents from concluding whether performing both tasks jointly supposes an advantage or not. In [78] a three-stage model is proposed by joining a two stage detection network with large feature extractor such as ResNet-50 with a recognition CNN.

In this chapter we propose a end-to-end model for text detection and recognition at page level. Our method jointly performs text localization and transcription, and thus, the transcription network can exploit the shared features with the detection branch. In addition, we evaluate our method with an ablation study that suggests that there are benefits in the end-to-end approach over training two models separately.

The rest of the chapter is organized as follows, first, in section 5.2 we describe the proposed model architecture, including the feature extractor, detection and recognition modules. In Section 4.3, we test the proposed method and perform an ablation study to compare our approach with a traditional two-step method. In the last section we draw the conclusions of the work and outline possible continuation lines.

4.2 Methodology

As explained in the previous section, most of the existing approaches for automatic handwritten text recognition consist of separated models for localizing and transcribing the text in the page image. Contrary, in this work we propose a method to exploit the benefits of deep neural architectures for multitasking, and to evaluate its performance compared to traditional approaches. For this purpose, we built a neural model to recognize the text from either a page or paragraph image, by detecting each word and transcribing its content. The architecture consists of four connected deep neural networks, one for extracting page or paragraph features (ResNet18 + feature pyramid network), another for detecting the existence of text in each part of the image (classification/objectness branch), another to regress the bounding box of each one of the words in a image (regression branch), and a recognition branch (conv+blstm). An overview of the whole model can be seen in Figure 5.1.

4.2.1 Feature extractor

The first module of our model is a deep feature extractor, whose weights are shared for the recognition and detection tasks. Taking into account that the localization of text in a scanned document (where we are previously aware of its existence) might be easier than detecting an object in the wild, we have chosen the ResNet18 [33], a light state-of-the-art architecture for object detection and classification. This architecture consists of 5 convolutional residual blocks, i.e. 2 convolutions with rectifier linear unit activation and a residual connection.The detailed list of blocks and their configuration is shown in Table 4.1.

Table 4.1: ResNet18 architecture used for feature extraction.

Layer	output shape	kernel size	# kernels
res-conv-block 1	H/2·W/2	3 x 3	64
res-conv-block 2	H/4·W/4	3 x 3	64
res-conv-block 3	H/8·W/8	3 x 3	128
res-conv-block 4	H/16·W/16	3 x 3	256
res-conv-block 5	H/32·W/32	3 x 3	512

We have tried other even lighter configurations, but when reducing the amount of layers, we observed slower convergence and worse final results. This was most probably caused by noisy detections and false positives, confusing text with non-relevant text. For this reason we have chosen an intermediate depth architecture that allowed regressing the characteristics of the text, and skipping the step of separating the regions of interest (i.e. including the layout analysis step in the whole process).

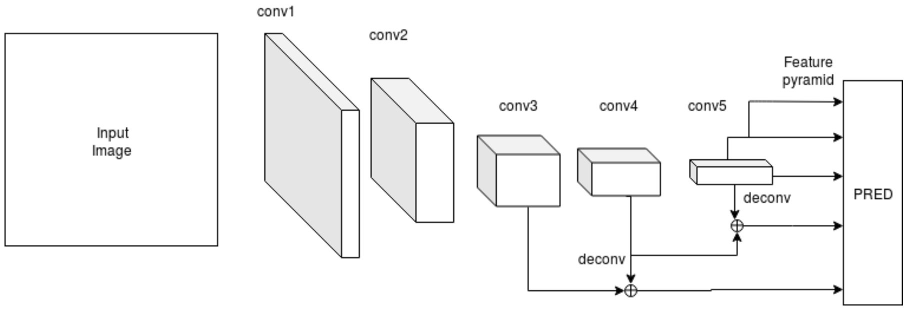


Figure 4.2: Feature pyramid network. It consists of 5 ResNet18 convolutional blocks followed by residual connections plus deconvolutions [50]. The input is an image of shape $H \cdot W \cdot 3$, the output is 5 tensors of 256 channels with down sampling factors of 4,8,16,32 and 64 respectively.

Based on recent work on object detection, we build a feature pyramid network (FPN) [50] which combines the extracted deep features of different levels of abstraction by means of *deconvolutions*. These type of layers consist of bilinear interpolation to apply differentiable upscaling of the high level features, followed by convolutions to reduce the number of channels, allowing to add them to lower level features. A diagram of this module is shown in Figure 4.2. This approach gave a boost in performance to detect objects at different scales, which is definitely also a beneficial feature for localizing text in documents.

4.2.2 Classification and regression branches

For each one of the levels of the pyramid, the extracted features are fed to the classification network, which after four convolutions will predict the probability of the presence of a text object for each point of the image grid.

After predicting the probability of an existing text object p_{cl}, the binary cross entropy (CE) loss is calculated and backpropagated through the classification branch and shared feature network. Formally, CE is computed as follows:

$$CE(p_{cl}) = -(y_{cl} \cdot log(p_{cl}) + (1 - y_{cl}) \cdot log(1 - p_{cl})) \tag{4.1}$$

In this case p_{cl} predicts the probability of the object of being text or not, i.e. it is used as an "objectness" value but it could be replaced by a probability vector to predict which kind of text it is, e.g. handwritten or printed or any other text categorization. In a similar way, the regression network receives the whole page image features and after four convolutions, it regresses the box coordinate offsets from the predefined anchors. Formally:

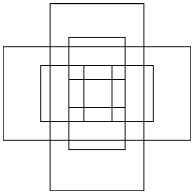


Figure 4.3: We used 9 predefined anchors, result of combining three ratios $\frac{1}{2}$,1,2 and scales 1, $2^{\frac{1}{3}}$, $2^{\frac{2}{3}}$.

$$\begin{aligned} x &= X + dx \cdot W \\ y &= Y + dy \cdot H \\ w &= e^{dw} \cdot W \\ h &= e^{dh} \cdot H \end{aligned} \tag{4.2}$$

where (x, y, w, h) are the predicted box coordinates, dx, dy, dw, dh are the predicted offsets and X, Y, W, H are the predefined anchor box values. The anchors are generated as the combination of the ratios $\frac{1}{2}$,1,2 and the scales 1, $2^{\frac{1}{3}}$, $2^{\frac{2}{3}}$ with a base size of 32 (9 anchors) as shown in figure 4.3 . The offsets are regressed by minimizing the mean square error shown in equation 4.3.

$$MSE(\delta, \delta') = \frac{1}{n} \sum_{i}^{n} (\delta_i - \delta'_i)^2 \tag{4.3}$$

where δ_i is the vector of target offsets from the anchors for the i-th ground truth box.

Once class probabilities and box offsets are predicted, the box sampler computes the bounding box coordinates for those anchors in the image grid whose class probability surpasses a given threshold. Once the box coordinates are calculated, we apply non-maximal suppression based on the maximum class score of each box. This will make that only a variable number of boxes with high confidence are going to be fed to the recognition branch.

4.2.3 Recognition branch

This module takes as inputs the features extracted with the FPN network and the corresponding regressed boxes and returns a probability tensor of variable width per alphabet length corresponding to the possible transcriptions of the words. Since the features are computed at page level we first need to apply some strategy that crops these features to be the input of the different successive layers in the module. This crop could be done in different ways, such as Region of Interest (ROI) pooling [24] strategies or affine transformation followed by Bilinear Sampling [39]. Since words in handwritten documents are mostly horizontally oriented and have no significant skew as it would happen in text in the wild data, for each regressed box, we apply RoI pooling to the page image features. Equivalently, for more complex distribution of documents with significant skew, or text orientation variations, affine transformations would allow to predict text skew or vertical orientation. The pooled features are rescaled to a fixed predefined height H' and variable width W', followed by padding to allow later addition of fully connected layers. For the recognition part we use a standard CRNN architecture containing 2 convolutional layers, 2 bidirectional long short-term memory layers[35], and a fully connected layer. As done in the previous chapter, to optimize the model for recognition we calculate the CTC loss.

To get the final full page transcription we concatenate the word predictions in a rule based reading order calculation from the word box coordinates. It consists of a projection of the word boxes to get text line continuation groups. To do so we used the `pagexml` library, the source code is publicly available here [1].

4.3 Experiments

This section is devoted to the experimental evaluation. We describe the dataset and the task to be realized, the different metrics and discuss the results.

4.3.1 Database and task description

We tested our method on the IAM [59] database, a multi-writer handwritten document collection with ground-truth at page level. It consists of 1539 pages of scanned text, written by 657 different writers, including a total of 13.353 annotated text lines and 115.320 words. Pages were scanned at a resolution of 300dpi (2479 × 3542) saved as PNG images with 256 gray levels. For our work, due to the limited GPU space we rescale the images to 150dpi (1240 × 1753). In our partition we use 1198 page images, 747 for training and 451 for validation/test.

4.3.2 Metrics

Two different metrics have been used in the experimental evaluation of the proposed methodology. One to evaluate the performance of the text detection and another for transcription.

[1] `https://github.com/omni-us/pagexml`

To evaluate the performance in text localization, we used the mean Average Precision (mAP), the standard metric in object detection. Let

$$p = \frac{TP}{TP + FP}$$

be the precision metric, i.e. the number of true positives out of the total positive detections;

$$r = \frac{TP}{TP + FN}$$

be the recall metric, i.e. the number of true positives out of the total ground truth positives, i.e. the true positives plus false negatives. We consider the recall-precision map, $p : [0,1] \mapsto [0,1]$ which maps the recall value r to the precision p that we would get if we had the detection threshold to get such a recall. Then, the Average Precision is the value $\int_0^1 p(r)$, i.e. the area under the precision-recall graph.

As a transcription score we choose the character error rate (CER), i.e. the number of insertions, deletions and substitutions to convert the output string into the ground-truthed one, divided by the length of the string. Formally:

$$CER = \frac{i + s + d}{\text{label length}}$$

4.3.3 End-to-end vs separate training

The performance of our method in terms of the CER and mAP is shown Table 4.2. As stated in the introduction, we also include an ablation study to assess the model visual capabilities when combining the text location and transcription annotations. Concretely, we are interested in the interaction of the learning processes of text localization and transcription. To evaluate the benefits of the intermediate error reduction, we compare the end-to-end versus the two-step training, i.e. separating localization and transcription. For the end-to-end training, approach we feed the full page image as an input, pass the predicted bounding boxes to the RoI pooling layer and back-propagate the 3 summed losses: CTC, classification and regression. Since our end goal is to get the best possible transcription, we use the validation character error rate as the early stop criterion. A main advantage in using end-to-end with RoI pooling instead from page features instead of training separately and cropping the predicted boxes directly from input image is that in the first approach the features contain contextual information that allows some error in the segmentation while still getting the correct transcription, as seen in figure 4.5

For the two-step training we first train the model by only back-propagating the regression and classification losses, and use as a early stop criterion the mean average precision of the validation detections (mAP). This means that we train the whole model, ignoring the recognition branch, to get best possible detections

Table 4.2: Comparison of methods for full page / paragraph recognition without language model. **These results are not directly comparable to our work due to segmentation level at test time and alphabet.*

	Val CER (%)	Test CER (%)	Det mAP	Segmentation at test time	Resolution	E2E feed forward
E2E	13.8	**15.6**	0.89	Full page	150dpi	YES
Two-step	10.5	19.3	0.9	Full page	150dpi	NO
With box GT	-	15.5	-	Word	150dpi	NO
Bluche et al. [6]	-	7.9 *	-	Paragraph	150dpi	YES
Puigcerver [63]	-	5.8*	-	Paragraph	-	YES

in the hypothetical case that we could not do the two tasks jointly. With this approach the model finished the training stage with a mAP of 0.9. When the detection training is finished, we train the whole model, but this time we ignore the classification and regression branches. Here we only backpropagate the CTC loss, using the ground truth word segmentation in the RoI pooling step, i.e. to get the separate best possible transcription using the same architecture as in the end-to-end approach.

Figure 4.6 presents a comparison of the behaviour (in CER) of both approaches: the separate recognition training and the end-to-end one, where we use the predicted segmentations and backpropagate the classification and regression losses. As expected the curve belonging to the separate is much smoother and decreases fast since there is no noise in the segmentation of the words. Nonetheless, we can see that in general, the end-to-end method is not far away from the separate one, and in the end, the difference in CER values is not significant. Looking at test time, the performance of the CER in both methodologies is the reverse, the end-to-end method is able to generalize better than the separate one, and gives a lower CER as shown in Table 4.2.

In addition, we evaluate the separate trained approach using the ground truth test word segmentation, to know which is the best transcription we could get at test time by using our architecture. Surprisingly, the CER in this case is not significantly lower than the end-to-end approach, which does not use segmentation at all.

Finally, Table 4.2 compares our method to some existing methods in the literature. We have chosen two methods that work at paragraph level, as the closest segmentation level to our work. Note that both approaches [6, 63] use the segmented paragraph as input. Moreover, the method described in [63] also uses data augmentation, which also brings a boost in performance. Also, we evaluate the character error rate at full page instead of line level, ignoring special characters and considering all characters lower case. Consequently, these results are not directly comparable to our work. In any case, we can observe that our method is competitive, especially taking into account that it does not require any kind of layout analysis.

Some qualitative results are shown in Figures 4.4 and 4.5. As it can be observed, most errors come from the miss-recognition of some characters. Looking in deep at those errors, we realize that humans could make the same mistakes if they only rely on the visual appearance of text. Of course, dictionaries and language models could help to reduce this kind of errors. For example, the last predicted word shown in Fig.4.5 would be correctly transcribed as "colleagues" instead of "caleagues".

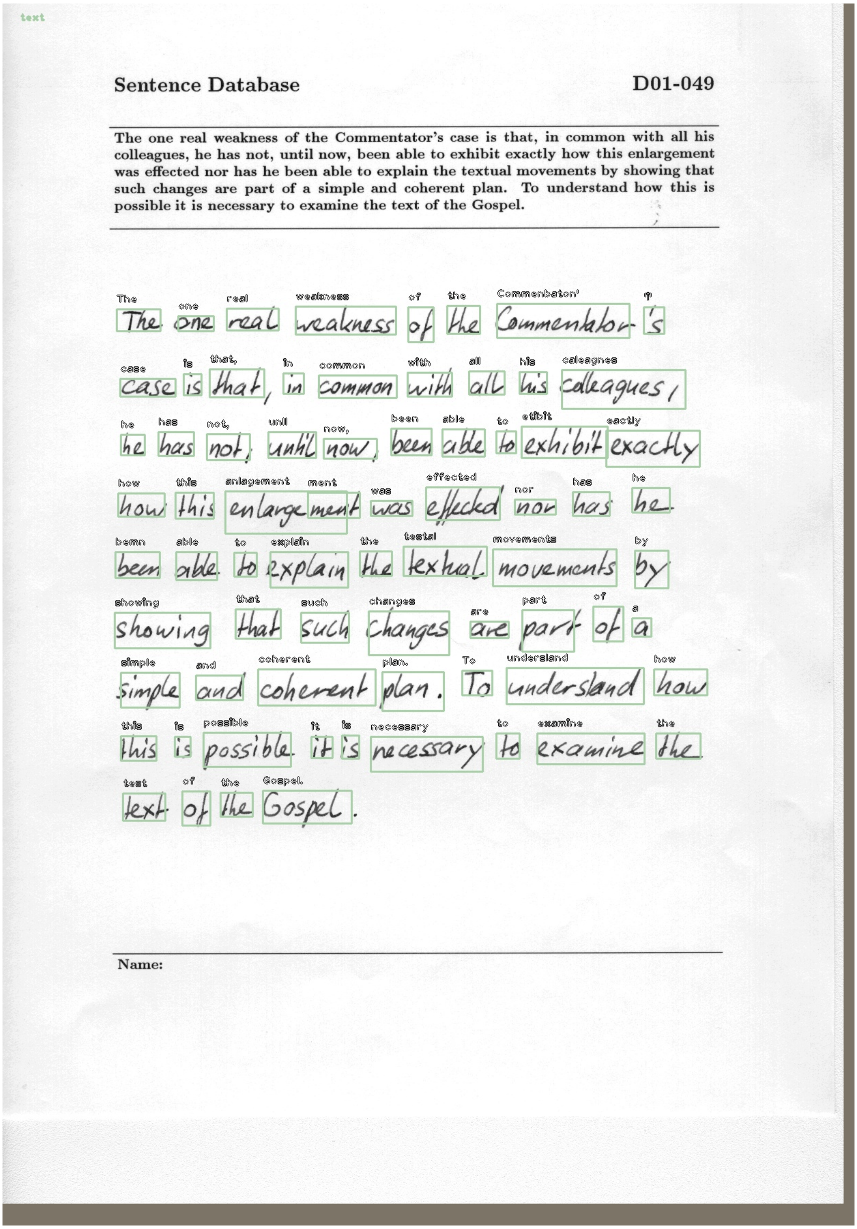
In summary, from the results we can conclude that our end-to-end approach is a promising technique in segmentation-free text recognition scenarios, and it can serve as a baseline for future works.

4.4 Conclusions

We have proposed an end-to-end method for text detection and recognition. It addresses the potential improvements suggested in previous HTR works by recognizing the text in a full page in a single feed forward end-to-end model. The model successfully allows end-to-end training backpropagating output transcription error to segmentation layers. This brings a couple of benefits and limitations. First, there is an intermediate step reduction with respect to the separate approach. The improvement could be caused by multiple reasons. One is the intended effect of backpropagating of the transcription (CTC) loss that leads to more adequate segmentation (but not necessarily with a higher detection score) at test time. Another possible cause is the regularization effect of training the recognition with more noisy segmentation, that prevents the model from overfit to the data. A unquestionable improvement is that this approach brings an inference time reduction, so for transcribing a large dataset the total difference might be significant. It would be interesting to perform some experiments in this direction and study in depth the causes of the improvement when the training is end-to-end.

It is also noticeable the memory usage reduction at inference time by sharing the model parameters for localization and transcription, and only seeing the page image once to predict the transcriptions, instead of having to work with two separate models. However, at training time it is necessary a high capacity GPU (at least 10GB) to load the recognition branch which increases the memory requirements significantly. To decode the output of our model we use a rule based reading order extractor from the word boxes. A possible future improvement to handle other arbitrary reading orders would be to add a sorting decoder RNN, inspired by the attention layer in [6].

A possible continuation of this would be to concatenate the detected words in lines before the RoI pooling. This would allow to train with documents where the ground truth only includes part of the segmentation information. Also, in the case of documents with text in different orientations, affine transformation with bilinear sampling as in [39] could be used instead of the RoI pooling. Having these



Sentence Database D01-049

The one real weakness of the Commentator's case is that, in common with all his colleagues, he has not, until now, been able to exhibit exactly how this enlargement was effected nor has he been able to explain the textual movements by showing that such changes are part of a simple and coherent plan. To understand how this is possible it is necessary to examine the text of the Gospel.

The one real weakness of the Commentator's
case is that, in common with all his colleagues,
he has not, until now, been able to exhibit exactly
how this enlargement was effecked nor has he
been able to explain the texhual movements by
showing that such changes are part of a
simple and coherent plan. To understand how
this is possible it is necessary to examine the
text of the Gospel.

Name:

text
The one real weakness of the Commenbaton' 'ṙ
case is that, in common with all his caleagnes
he has not, unll now, been able to etibit eactly
how this anlagement ment was effected nor has he
bemn able to explain the testal movements by
showing that such changes are part of a
simple and coherent plan. To undersland how
this is possible it is necessary to examine the
test of the Gospel.

Figure 4.4: Example of Localized words in a page from the IAM dataset.

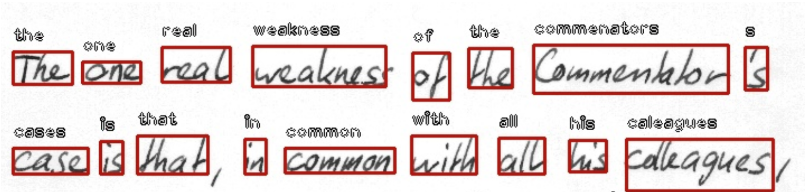


Figure 4.5: Example predictions on unseen page. Note that by predicting the text sequence from pooled regions instead of the input image in an end-to-end fashion, the model is able to handle segmentation errors.

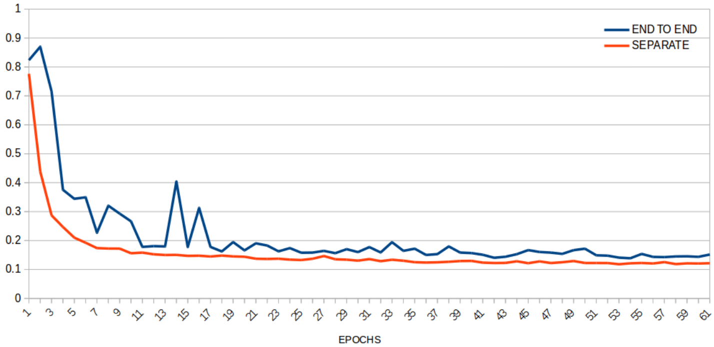


Figure 4.6: Character Error Rates for the Validation set. We compare the separate recognition training vs the end-to-end training.

observations, the presented method opens the door to combine more tasks coming after localization and transcription in a unified model, such as the recognition of named entities.

Chapter 5

Joint text localization, transcription and named entity recognition

In the last years, the consolidation of deep neural network architectures for information extraction in document images has brought big improvements in the performance of each of the tasks involved in this process, consisting of text localization, transcription, and named entity recognition. However, this process is traditionally performed with separate methods for each task. In this chapter we propose an end-to-end model that combines a one stage object detection network with branches for the recognition of text and named entities respectively in a way that shared features can be learned simultaneously from the training error of each of the tasks. By doing so the model jointly performs handwritten text detection, transcription, and named entity recognition at page level with a single feed forward step. We exhaustively evaluate our approach on different datasets, discussing its advantages and limitations compared to sequential approaches. The results show that the model is capable of benefiting from shared features for simultaneously solving interdependent tasks.

5.1 Introduction

Having explored end-to-end models that combine recognition of text with NER and with localization, in this chapter we close the IE task interdependency loop and explore the combination of localization and NER as well as the three tasks (localization, HTR and NER) simultaneously unifying the whole process in a single end-to-end architecture. We test our method on different scenarios, including data sets in which there is bi-dimensional contextual relevant information for the named

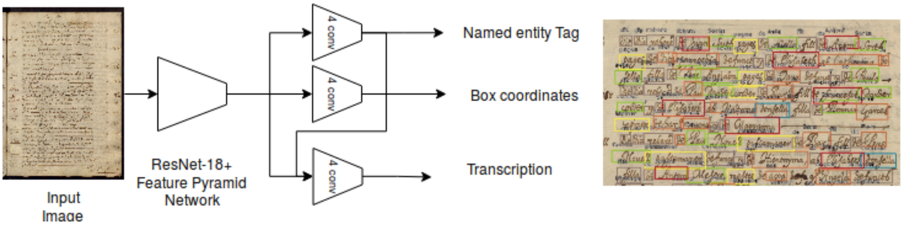


Figure 5.1: Overview of the proposed method. Convolutional features are extracted with ResNet18 and FPN. The classification and regression branches calculate the positive boxes and the recognition branch predicts the transcription of the content of each box. Cross entropy, squared-sum and CTC losses are backpropagated through the whole model for training.

entity tag, or there is an inherent syntactic structure in the document.

We experimentally validate the different alternatives considering different kind of documents, while studying properties such as geometric context relevance, layout regularity or density of named entities in the text.

The resto of the chapter is organized as follows. In section 5.2 we describe our joint model. In sections 5.4 and 5.5, we present the datasets, the experimental results and discuss the advantages and limitations of our joint model. Finally, in section 5.6 we draw the conclusions.

5.2 Methodology

This method follows the procedure explained in previous chapter, while extending the model for the named entity recognition in different setups. As introduced before, our model extracts information in a unified way. First, convolutional features are extracted from the page image, and then, different branches analyze these features for the tasks of classification, localization, and named entity recognition, respectively. An overview of the architecture is shown in Figure 5.1.

5.3 Shared features

Since the extracted features must be used for very different tasks, i.e. localization, transcription and named entity recognition, we need a deeper architecture than the one used for each isolated task. In a similar way as in previous chapter we choose as a backpone architecture the ResNet18.

We chose an intermediate depth model which allows to tackle the tackled complex tasks at once. The output of the Feature Pyramid Network is a set of 5 down sampled feature maps with scales 8,16,32,64,128. Each of these are forwarded to the upcoming branches and their output is stacked in a single tensor, from which we later select the most confident predictions.

5.3.1 Classification branch

For similar reasons as in previous chapter we have chosen the state of the art one-stage object detection as our classification branch. The architecture of this branch is shown in table 5.1.

Table 5.1: Classification and regression branch architectures, where downsampling levels are $ds_{l_i} \in \{8, 16, 32, 64, 128\}$.

Layer	**output shape**	**kernel size**	**| kers|**
conv-block 1	$H/ds_{l_i} \cdot W/ds_{l_i}$	3 x 3	256
conv-block 2	$H/ds_{l_i} \cdot W/ds_{l_i}$	3 x 3	256
conv-block 3	$H/ds_{l_i} \cdot W/ds_{l_i}$	3 x 3	256
conv-block 4	$H/ds_{l_i} \cdot W/ds_{l_i}$	3 x 3	256
conv-block 5	$n_{\text{anchors}} \cdot \{n_{\text{classes}}, 4\}$	3 x 3	1

We also explored to use this branch as a named entity classifier. The motivation behind is to take context into account through the prediction of the presence of certain features in a neighbourhood of a point of the convolutional grid. The difficult part comes when attempting to capture dependencies between distant parts of the image, as it happens when a sequential approach is used. The classification branch, or objectness loss in case of a pure text localizer classifier, is trained with the cross-entropy loss as in the previous text localization loss equation 6.5.

5.3.2 Regression branch

To predict the coordinates of the box positives, the regression branch receives the shared features and, in a similar way as the model presented in the previous chapter the offset values are predicted from the predefined anchors.

Also in this case the offset of the predefined anchors is regressed by minimizing the mean square error. Again the anchors are generated as the combination of the ratios $\frac{1}{2}$,1,2 and the scales 1, $2^{\frac{1}{3}}$, $2^{\frac{2}{3}}$ with a base size of 32 (9 anchors).

5.3.3 Feature pooling

Once we have predicted the class probabilities and the coordinate offsets for each anchor in each point of the $Im_H/8 \times Im_W/8$ convolutional grid, we select the boxes whose confidence score surpasses a given threshold, and remove the overlapping ones applying a non-maximal suppression algorithm. With the given box coordinates, we apply RoI pooling [24] to the convolutional features of the full page, but saving the input to allow error backpropagation to further branches. We use the 5 levels of the feature pyramid to calculate the box anchor offsets and the objectness values. For computational reasons, we only keep the least downsampled features for the text recognition and named entity recognition branches, as we need the highest possible resolution for those tasks.

5.3.4 Text recognition branch

As in the previous chapter, we build a recognition branch that will predict the text contained in each box. The architecture of this branch, shown in Table 5.2, consists of two convolutional blocks followed by a fully connected layer. The output of this layer is the probability of a character for each column of each one of the pooled features. From these predictions, we calculate the CTC loss as done with the previously proposed models.

This loss is added to the classification and regression losses to backpropagate them together for each gradient update.

Table 5.2: Recognition branch architecture.

Layer	**Output shape**	**ker size**	**| kers|**
conv-block 1	pool H·pool W	3 x 3	256
conv-block 2	pool H·pool W	3 x 3	256
Fully connected	pool W· $\|alphabet\|$	-	-

5.3.5 Semantic annotation branch

One possibility to assign a semantic tag to each word is to predict its class from the classification branch for each anchor. However, this would not capture the context as the activations only rely on the convolutional feature maps of a neighborhood of each point. For this reason, we add this network branch to predict the semantic tags as a sequence from the ordered pooled features of each box. For simple layouts, such as single paragraph pages, the pooled features, which correspond to text boxes in the page, are sorted in conventional reading order (i.e. left to right and top to bottom) by projecting a continuation of the right side of the text box. Once we have the ordered pooled features, we pad them and apply two convolutions followed by a fully connected network as a standard named entity recognition architecture. Then, we minimize the cross entropy loss shown in equation 6.5 for each of the sequence values.

5.3.6 Receptive field calculation

Our approach assumes that each activation of a neuron in the deepest layers of a CNN depends on the values of a wide region of the input image, i.e. its receptive field. Also it is important to notice that the closer a pixel is to the center of the field, the more it contributes to the calculation of the output activation. This can be a useful property for documents where the neighboring words determine the tag of a given word, but it can also be a limitation when distant entities are related in a document. To calculate how much context is taken in account for each unit of the features that are fed to the RoI pooling layer, we must look at the convolutional kernel sizes k and strides s of each layer. In this way, as in [19], we can calculate the relation between the receptive field size of a feature map depending on the previous layer's feature map:

$$r_{out} = r_{in} + (k - 1) \cdot j_{in} \tag{5.1}$$

where j_in is the *jump* in the output feature map, which increases in every layer by a factor of the *stride*

$$j_{out} = j_{in} * s \tag{5.2}$$

By using these expressions with our architecture (ResNet 18 + FPN), we obtain a receptive field size of 1559 in the shared convolutional feature map. That means that, since the input images are 1250×1760, the values predicted for each unit mostly depend on the content of the whole page, giving more importance to the corresponding location of the receptive field center.

5.4 Datasets

One of the limitations when exploring learning approaches for information extraction is the few publicly available annotated datasets, probably due to the confidential nature of this kind of data. Nonetheless, we test our approach on three data sets. The details of amount of pages, words, out of vocabulary (OOV) words and partitions can be found in Table 5.3.

5.4.1 IEHHR

For this chapter we use the full page version of the IEHHR which contains 125 handwritten pages with 1221 marriage records (paragraphs). In a similar way as before, for each record we find the information of the husband, wife and their parents' names, occupations, locations and civil states. On the sides of each paragraph we find the husband's family name and the fees paid for the marriage. An example page is shown in Figure 5.6.

Table 5.3: Characteristics of the datasets used in our experiments. Entities refer to the amount of relevant words (i.e. they do not belong to the class "other").

	Part	**IEHHR**	**WR**	**sGMB**
Pages	train	79	994	490
	valid	21	231	53
	test	25	323	50
Words	train	2100	2837	7010
	valid	878	731	1740
	test	1020	1033	4085
‖OOV‖	all	387	853	1372
% OOV	all	37	82	34
% entities	all	52.5	100	17
‖entity tags‖	-	5	3	5

5.4.2 War Refugees

The War Refugees (WR) archives contain registration forms from refugee camps, concentration camps, hospitals and other institutions, from the first half of 20th century. We have manually annotated the bounding boxes, transcriptions and entity tags of names, locations and dates. Due to data privacy we cannot share the images, but instead we show in Figure 5.2 a plot of all annotated text normalized bounding boxes, where the colors correspond to different tags. As we can observe, there is a strong pattern relating the text location and its tag, although it is not fixed enough for applying a template alignment method. The main difficulty of this dataset is to distinguish relevant from non-relevant text, which in most cases only differs by its location or by a nearby printed text key description. Another challenge is the high amount (82%) of out of vocabulary words, together with the high variability of the writing style and the mixture of printed and handwritten text.

5.4.3 Synthetic GMB

We have generated a synthetic dataset (sGMB) to explore the limitations of our model, concretely in a standard named entity recognition task, in which text is unstructured and the amount of named entities within the text is low. For this purpose, we have generated synthetic handwritten pages with the text of the GMB dataset [8] by using synthetic handwritten fonts, applying random distortions and noise to emulate realistic scanned documents. Although it is easier to recognize synthetic documents than real ones, the difficulty here remains on the sequential named entity recognition task, especially because, contrary to the previous datasets, here the text does not follow any structure. An example is shown in

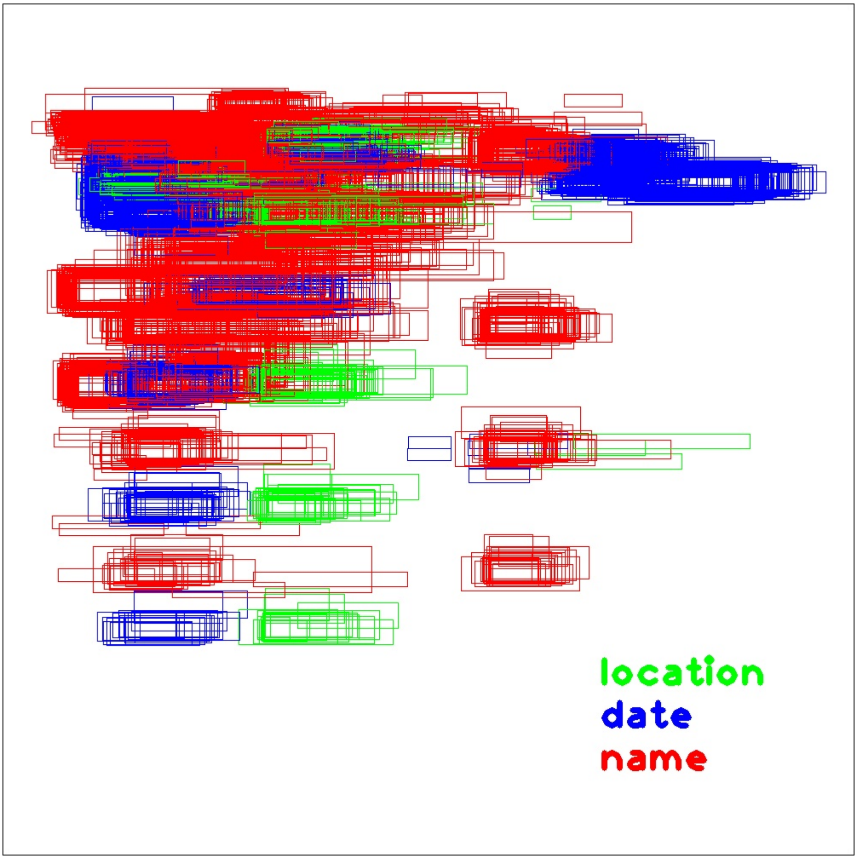


Figure 5.2: Normalized bounding boxes of the tagged text of all training images in the WR dataset.

Figure 5.3. The dataset and ground truth are available here [1].

5.5 Experiments

In this section we describe the experiments performed for each data set. We optimized our model with stochastic gradient descent on the three described losses in section 5.2. When unifying tasks in a single model, the model becomes quite memory expensive. To overcome this limitation, during the training, we had to set our batch size to be 1 page. Based on previous object detection work we have chosen the learning rate to 10^{-4}, the non-maximum suppression threshold to 0.2 and the box sampling score threshold to 0.5. We have chosen a patience of 100

[1] `https://github.com/omni-us/research-dataset-sGMB`

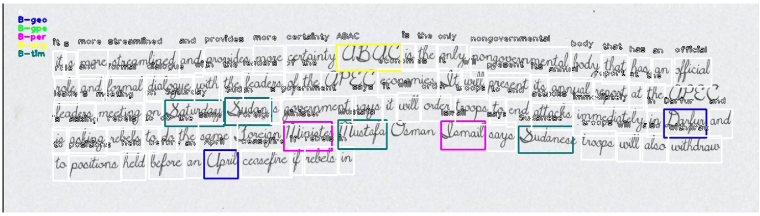


Figure 5.3: A generated page from the SynthGMB dataset. A major difficulty is the sparsity of named entities with respect the other words.

epochs to trigger early stop.

5.5.1 Setup

In this section we explain the possible setups in which our proposed model can be used. Our approach has been evaluated using the following different configurations:

- **A: Triple task model**. The first method variation consists in using our proposed model to perform all tasks in a unified way using the objectness branch for named entity classification as explained in section 5.3.1, with no sequential layers but only convolutional ones.
- **B: Triple task sequential model**. The second variation also performs the three tasks in a unified way, but by concatenating the pooled features in reading order and predicting the labels sequentially with the semantic annotation branch described in section 5.3.5.
- **C: Detection + named entity recognition**. In this case we consider to face the extraction of the relevant named entities as a detection and classification problem. Here, we ignore the recognition part and only backpropagate the classification and regression losses from their respective branch outputs. We also consider the sequential version of this approach using the semantic annotation branch.
- **D: Detection + transcription**. Here we combine in a unified model the tasks of localization and transcription, as our previous work explained in chapter 4, in contrast to an approach in which the two tasks are faced separately, where the recognition model would cope with inaccurate text segmentations. Here we aim to observe how precisely we can obtain text boxes and transcriptions in this context, so that named entity recognition can be applied afterwards.
- **CNN classifier**. Finally, we evaluate the variability of the cropped words among the different categories and the difficulty of annotating words separately. Thus, we train a CNN network, similar to the classification branch

from our proposed method, that classifies words without using any shared features for recognition or localization. So, this network does not benefit from context information.

Diagrams of each setup can be seen in Figure 5.4. The full source code for all experiments is publicly available here [2].

5.5.2 Metrics

Different metrics have been used to evaluate the proposed methodology. One to evaluate the performance of the text detection, one for named entity recognition, and another for transcription. For text localization we used the the widely used object detection metric, average precision. For named entity recognition, we used the $F1$ score. Let $p = \frac{TP}{TP+FP}$ be the precision metric, i.e. the number of true positives out of the total positive detections; $r = \frac{TP}{TP+FN}$ be the recall metric, i.e. the number of true positives out of the total ground truth positives, i.e. the true positives plus false negatives. We consider the recall-precision map, $p : [0,1] \mapsto [0,1]$ which maps the recall value r to the precision p that we obtain if we had the detection threshold to get such a recall. Then, the Average Precision is the value $\int_0^1 p(r)$, i.e. the area under the precision-recall graph. The $F1$ score consists of the harmonic mean between the precision and the recall:

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{5.3}$$

For text localization all the word bounding boxes are considered in the metric. In contrast, for named entity recognition only the entity words count, as it is standard in this kind of task. Also the recognized text is not used to match the entities, only the location of the word within the page and the entity tag. This is so that the metric only evaluates the prediction of entities completely decoupled from the transcription performance.

For the transcription score we use the Character Error Rate (CER), i.e. the number of insertions, deletions and substitutions to convert the output string into the ground-truthed one, divided by the length of the string. Formally:

$$CER = \frac{i + s + d}{\text{label length}}$$

5.5.3 Results

From the results shown in Table 6.1, we observe that the localization performance is high in general being this one the less challenging task of all. Differences among the different setups and datasets are not significant.

[2] `https://github.com/omni-us/research-e2e-pagereader`

For text recognition all methods perform similarly except in WR dataset. Setup A shows a slight better performance than B and D. This suggests that the eavesdropping effect mentioned in [70] might be taking place in this case. The outnumbering of non-labeled words vs labeled in WR dataset increases considerably the difficulty of the recognition task for this dataset due to the same type of error observed in figure 5.5. We cannot show images of this phenomena observable during training due to the confidential nature of the documents. A possible solution to this would be to also have annotated words which are not entities as it happens in IEHHR and sGMB.

For the entity recognition part, we do not observe significant differences among the end-to-end approaches in the *IEHHR* dataset. This suggests that the local neighborhood information seems enough to give correct predictions. The high named entity recognition performance using the triple task sequential approach (case B) in WR dataset suggests that it is beneficial to combine the tasks of named entity recognition and localization. To have a better idea of whether the proposed method makes use of context or the sole content of the word is sufficient, we compare its performance to the CNN classifier for segmented words. Since we are facing a named entity recognition task, we only take into account the entities, i.e. words labeled as 'other' are not taken into account after the text localization step. By doing so, in the IEHHR dataset we observe a greater performance for the end-to-end models compared to the CNN. This is an evidence that context is being used since in the CNN approach there is the advantage that an explicit perfect segmentation is given. In the case of WR we observe an even greater boost of performance when using the context, specially the bi-dimensional one (setup A). We attribute this to the inherent layout pattern contained in the dataset as it can be intuited observing Figure 5.2. We also assume that predictions were based on the layout due to the large amount of out of vocabulary words, which would make it difficult to predict the word category based on a known vocabulary. sGMB is clearly the most difficult one since it has the sparsest distribution of entities as it can be seen in Table 5.4. With this dataset we also see a very substantial increase in performance when using the context, concretely, the sequential one (setups B and C). This makes sense as the type of text is natural language, which means that it has a sequential structure but lacks the bi-dimensional layout structure of registration forms or marriage records. Consequently, sequential patterns existing in natural language are more suitable to recognize entities.

5.6 Discussion and Conclusion

In this chapter we have presented a unified neural model to extract information from semi-structured documents. The proposed method shows the strengths of the pairwise interaction of some of the tasks, such as localization and transcription and also for localization and named entity recognition when the spatial information or the neighbourhood (geometric context) of a text entity influences the value

Table 5.4: Performance of the different method variations on each dataset.

Method	IEHHR	WR	sGMB
Text localization (AP)			
A: triple task	0.97	0.976	0.994
B: triple task seq	0.972	0.973	0.994
C: Det+ner seq	0.969	0.975	**0.997**
D: det+htr	**0.974**	**0.981**	0.996
Text recognition CER (%)			
A: Triple task	**6.1**	**23.7**	**2.3**
B: Triple task seq	6.3	28.7	2.6
D: Det+HTR	6.5	27.5	2.5
Named entity recognition (F1)			
A: Triple task	0.797	**0.975**	0.347
B: Triple task seq	**0.806**	0.924	**0.535**
C: Det+NER seq	0.796	0.963	0.510
CNN classifier	0.700	0.821	0.382

to predict. Nevertheless observing the performance of triple task neural model variations, it must be noted that a unified model can be limited in performance in cases where one specific task is much harder and unrelated to the others. In such a case, a separate approach would allow us to use specific techniques for this difficult unrelated task. For example, named entity recognition performance is limited by the fact that it is very difficult to generate semantically meaningful word embedding vectors (e.g. word2vec, glove) when the model input is a page image.

In summary, we conclude that a joint model is suitable for cases in which there is a strong task interdependence, but not for documents where the main difficulty is on one independent single task.

As we have seen the proposed approach can identify local spatial relationships, due to the design of the receptive fields. However, unless we are working on sequentially organized data this type of approach would struggle to capture longer distance spatial relationships in a semi structured document. This leads us to study models that solely focus on this more difficult task in an isolated way.

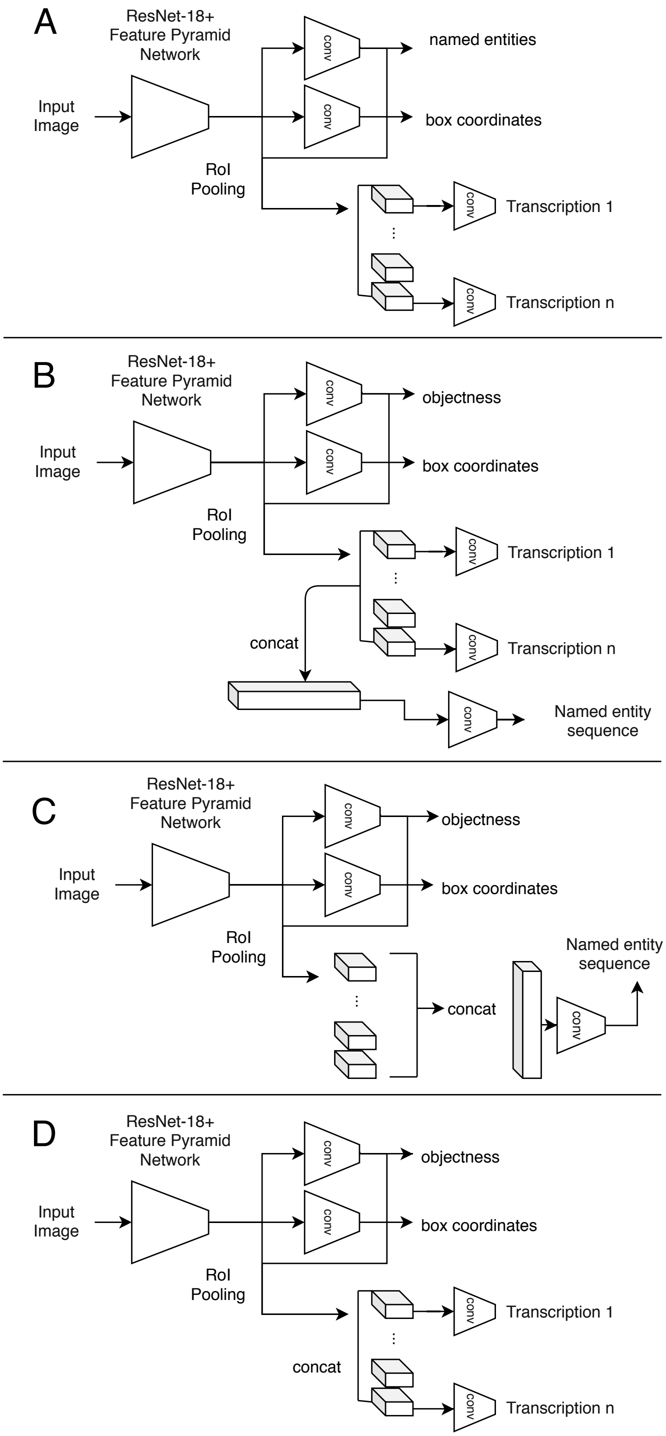


Figure 5.4: Model setup variations, A-D. The differences rely on how the named entities are recognized, and the optional integration of the recognition branch.

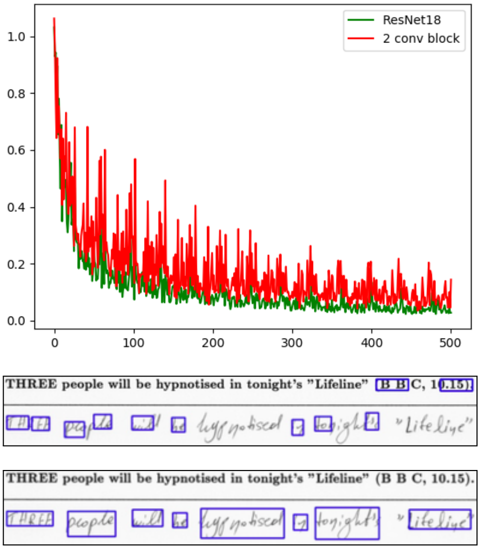


Figure 5.5: (Top) Plot comparing the regression loss during training on the benchmark dataset IAM for ResNet18 (green) against a 2 convolutional block reduced version (red). (Middle) Predictions on IAM with 2 convolutional block model. (Bottom) Predictions on IAM with ResNet18 model. The model does not confuse relevant with irrelevant text.

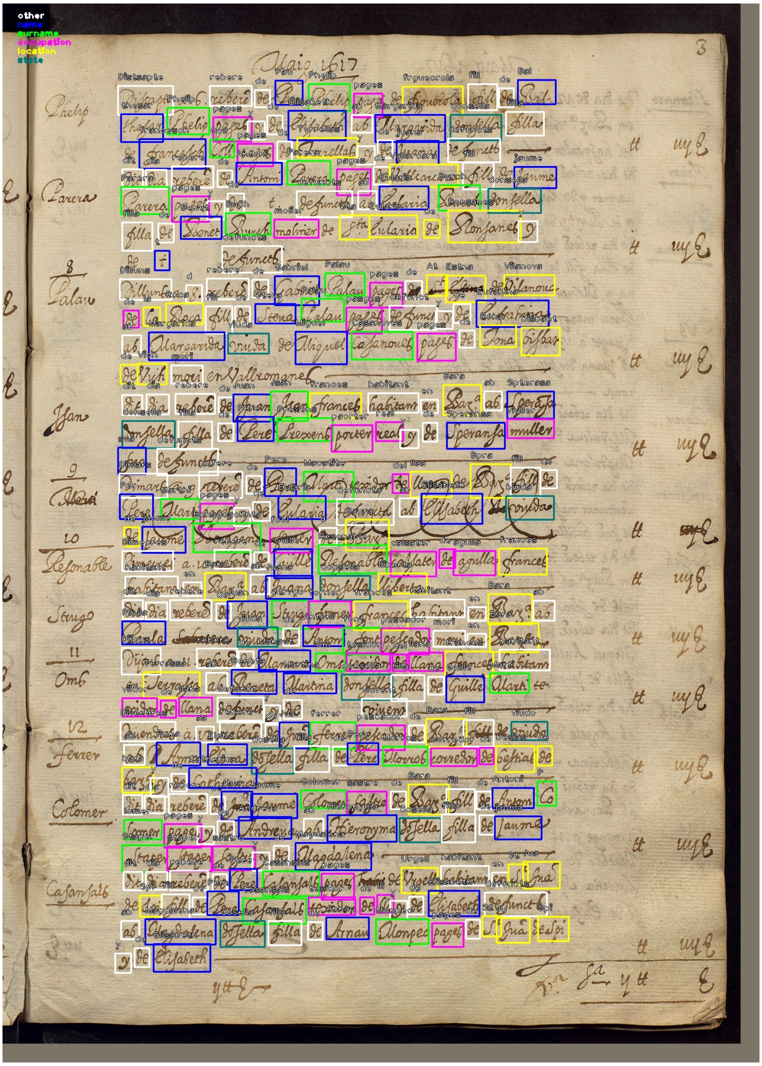

other
name
surname
occupation
location
state

Figure 5.6: Word localizations, transcriptions and semantic annotations on an unseen page of the IEHHR dataset. The model learns to detect and classify words based not only on its appearance but also on its context. The colors illustrate the different type of named entities.

Chapter 6

Named Entity Recognition and Relation Extraction with Graph Neural Networks

Graph Neural Networks provide the proper methodology to learn relations among the data elements in semi-structured bureaucratic documents. In this chapter we study their use to tackle the problem of entity recognition and relation extraction. Our approach achieves state of the art results in the three tasks involved in the process. Additionally, the experimentation with two datasets of different nature demonstrates the good generalization ability of our approach.

6.1 Introduction

When building a model to learn some pattern on the data, there are relational assumptions implicitly made, such as assuming that the pattern is going to be present in bidimensional neighborhoods in the case of convolutions on images, or assuming that it is going to be present in a sequence as in the case of RNNs. This idea is known as *relational inductive bias* [4]. In the context of IE the high variability of the data in the task of finding layouts and relationships within document elements suggests that other arbitrary inductive bias should be chosen. In our previous work [11], and continuations of it [93] the mentioned tasks are faced with architectures originally designed for vision, which have locality inductive bias. These methods achieve acceptable results but also are not allowing the mentioned arbitrary relational inductive bias assumptions among the document, which motivates the use of Graph Neural Networks (GNNs). Several work has been done exploiting the combination of neural architectures with graph structured data with great success, extending their breakthrough on vision and natural language to many other domains such as quantum chemistry, knowledge graphs, or citation networks [76] [44] [23]. In the work of Velickovic *et al.*. [86] the idea of attention is brought together with GNNs leveraging masked self attention layers, having in this way a specially adequate architecture to efficiently solve not only problems such as node classification with prior known graph structure but also structure inferring problems such as link prediction. In this work we tackle the problem of finding relationships between elements in a document, *i.e.* predict links between entities by means of a Graph Neural Network model.

Liu *et al.* [54] proposed a GNN based approach for NER in visually rich documents that successfully classifies named entities suggesting its potential capability of performing other tasks of information extraction. Recently, in the work of Riba *et al.* [67] a Graph Neural Network is trained to detect tables in different types of business documents, predicting relationships between table elements. Other notable contributions in the field are the LayoutLM model [90], and [92]. The first one is based in the idea that BERT [18] derived architectures provide a powerful resource to extract patterns in sequential data. Hence in their work they convert the input data in a sequential format comprising embedded layout as well as textual information to successfully classify entities. The latter one combines this idea with the use of GNNs to jointly predict the contents of documents with a predefined structure as in the case of the ICDAR 2019 Competition on Scanned Receipt OCR and Information Extraction [36]. Conversely, in the case of this work we further extend the previous method by giving to the model the possibility to predict links between the entities whose type and amount might be unknown a priory.

In this chapter, we propose a novel method to extract structured information from semi structured documents by means of GNNs. Inspired by [67] we extend this idea to a more generic context were also key-value pairs which are not strictly table elements are predicted, and also entities are classified in different categories. The whole system demonstrates the ability to solve the three tasks with state of the art performance. Summarizing, the main contents of this chapter are:

- We cast the named entity recognition and relation extraction as a supervised message passing task.
- We surpass state-of-the-art performance of the three tasks involved.
- Our model generalizes to weakly structured documents, as we show in the experimental part validating it images of historical marriage licenses.

The rest of the chapter is organized as follows. Section 6.2 introduces the proposed pipeline for named entity recognition and relation extraction, as well as the specific GNN chosen architecture for our work. Next in section 6.3 we describe the datasets and metrics to test the approach, and we show the obtained results. Finally, section 6.4 draws the conclusions extracted from the experiments.

6.2 Methodology

In this section, we introduce our approach for name entity recognition and relation extraction. We focus on the steps of document understanding coming once the OCR has been already performed. Therefore, we consider that the raw textual content of the document is already available and to better isolate the problem we make use of the ground-truth transcriptions as well as bounding boxes.

6.2.1 Problem formulation

Given an input document the model has to be able to (i) detect the document entities *i.e.* groups of words with a semantic meaning; (ii) classify the detected entities into predefined categories and; (iii) discover the meaningful pairwise relationships between entities. These tasks are named as word grouping, entity labeling and entity linking respectively.

The proposed architecture is divided in several components. Each of them is trained for a single task independently from the others. Thus, in total three different GNN models, $f_1(\cdot)$, $f_2(\cdot)$ and $f_3(\cdot)$, are considered. The document is initially represented as a graph G_1 whose nodes are the words detected in the OCR process. Edges between words are created using k nearest neighbors (k-NN) based on the distances of the top-left corner of the word bounding boxes. The GNN first identifies groups of words corresponding to entities by doing edge classification. Subsequently, the graph is contracted according to the detected groups (graph G_2) in order to perform the tasks of entity labeling as a node classification approach and entity linking as link prediction pipeline. An overview of this approach is introduced in figure 6.1 for the first task and in figure 6.2 for the other ones.

6.2.2 Word grouping

The first task towards a framework able to understand the complex structure of a document is to group the words which belong to the same semantic entity. This task requires to combine both sources of information, on the one hand, the textual content and, on the other hand, the pairwise relationships with other words. Thus, we consider the task of finding groups of words as a link prediction problem in the graph of the document.

With this aim, the graph $G_1 = (V_1, E_1)$ is constructed by considering each detected word as a node. To initialize the node features, we first calculate a fasttext

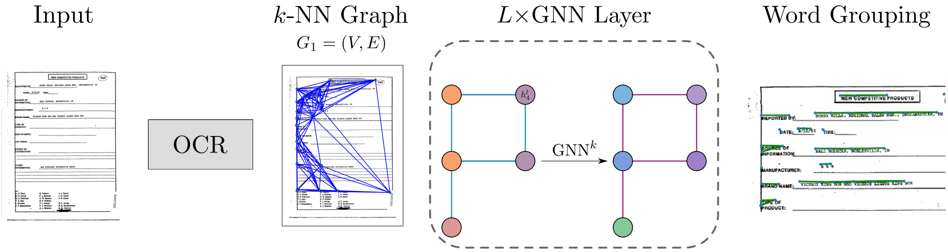


Figure 6.1: Overview of the proposed word grouping approach. The text content and location of the words in the input document is encoded in a word level k-NN graph. This is fed into a GNN with L layers. The word grouping is formulated in terms of a binary edge classification problem, that is, 1's indicates that these words belong to the same entity.

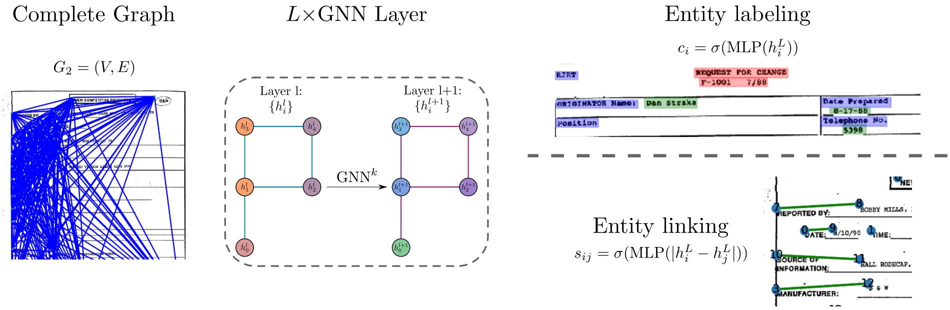


Figure 6.2: Given the discovered entities (see figure 6.1), a complete entity level graph is generated and fed into L GNN layers. Thus, the tasks of entity labeling and entity linking are formulated in terms of node and edge classification respectively. The GNN is trained separately for each task.

word embedding model [7] by linearizing the text of the training documents ordered as given by the OCR process. An important benefit of using fasttext is that at prediction time it is possible to get meaningful embeddings for words not observed in the training set, which is a rather common occurrence in administrative documents.

Given a node $v_i \in V_1$, its initial hidden state vector $h_i^0 = [x_i, y_i, w_i, h_i, w_{\text{embed}}]$ is the concatenation of the word embedding with the corresponding bounding box width, height, and top left corner position normalized with respect to the page size. Having calculated $h^0 = \{h_1^0, \ldots, h_n^0\}$ we generate k-NN graph G_1 with $k = 10$ since the complete graph–all nodes connected with each other–makes the problem computationally unfeasible. The number of neighbors for constructing the graph has been chosen experimentally making sure the minimum number of candidate edge between words is missing while keeping the number of edges low. This hyper-parameter could be further tuned but it is beyond the main scope of this work. The generated graph is going to be further processed by our L layer GNN architecture $s = f_1(G_1)$ where s are the final link predictions.

To get the word groups from the link predictions we keep the edges whose predicted scores are greater than a threshold τ, and, by connected components, we define the entities.

6.2.3 Entity labeling

Assuming that the previous word grouping task has been successfully solved, in this step we want to classify each group of words or equivalently *semantic entity* with its corresponding label. For this case, let us consider a graph $G_2 = (V_2, E_2)$ as the entity graph, where each node represents an entity. For this module we considered the complete graph since the number of nodes is drastically reduced. Then the label for a given entity is calculated in terms of node classification. Thus, following the notation mentioned above, $c = f_2(G_2)$ where c are the predicted entity labels.

6.2.4 Entity linking

Similarly to the previous task, entity linking makes use of the complete graph G_2 as its input. However, this task is cast as an edge classification framework following the same pipeline introduced for the word grouping task. Therefore, our model binary classifies edges to predict the existence or absence of links between nodes. Thus, $s = f_3(G_2)$ where s are the predicted scores per each edge.

6.2.5 Architecture

Here we describe how our three graph models are built to solve the above described problem. With our approach the model extracts structured information combining two types of processes: (i) given a set of node vectors, find the structure of graph, i.e. predict the existing edges between them. This is used for the word grouping part as well as for entity linking; (ii) given a set of nodes, classify each of them in a predefined category. This is used for the entity labeling part.

The proposed tasks, do not only predict classes in the set of nodes, but also relationships among words and entities in a document. This second objective requires to infer the meaningful structure given a set of node data and partially known edge information rather than making use of static ground truth edge connectivity to predict values for nodes. For this type of task GAT layers have shown to be very adequate, therefore, we selected them as the base of our GNN architecture.

In the following lines, we describe the backbone of our architecture independently to the final task. Let $G = (V, E)$ be a graph where $e_{ij} \in E$ denotes the edge between nodes $v_i, v_j \in V$. Let $n = |V|$ be the number of nodes in the input graph then GAT layers receive a set of nodes features $h^l = \{h_i^l\}_{i=0^n} \in \mathbb{R}^{F_l}$ and return an updated set of those nodes $h^{l+1} = \{h_i^{l+1}\}_{i=0}^n \in \mathbb{R}^{F_{l+1}}$ according to the pairwise relationships defined in E. GAT layers follow the idea of attention in CNN's to decide which are the important connections. Therefore, for each pair of nodes (v_i, v_j) the *attention coefficients* α_{ij} are calculated:

$$\alpha_{ij} = \frac{\exp(\text{LeakyRelu}(V[Wh_i||Wh_j]))}{\sum_{k \in \mathcal{N}(v_i)} \exp(\text{LeakyRelu}(V[Wh_i||Wh_k))} \tag{6.1}$$

where $\mathcal{N}(v_i)$ is the set of neighboring nodes of v_i, W and V are weight matrices with learnable parameters and $||$ is the concatenation operator. Following the Transformer architecture practices [85] we use K attention heads. Hence, K attention coefficients are computed and aggregated in order to obtain the updated node hidden state h^{l+1}. Thus, a GAT layer is defined as:

$$h_i^{l+1} = g(h_i) = \Big\|_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k W^k h_j^l\right). \tag{6.2}$$

In our experiments, we consider the backbone model of our functions $f_1(\cdot)$, $f_2(\cdot)$ and $f_3(\cdot)$ as L GAT layers.

The tasks that we are facing for document understanding can be summed up in node classification and link prediction. The first one simply consists to assign a label $c_i \in C$ to each node v_i in the input graph G. The second one consists of predicting the existance or absence of an edge between each pair of nodes. For the first case we simply feed the hidden state node representation to a Multi Layer Perceptron (MLP) with a sigmoid activation function, predicting this way each class probability for node v_i:

$$c_i = \sigma(Wh_i^L), \tag{6.3}$$

where $W \in \mathbb{R}^{F_L \times C}$ is a learnable weight matrix, C is the number of classes and h_i^L is the node hidden state at the last layer.

In the case of link prediction we also use a MLP but in this case receiving a list of all the candidate node pairs and returning their link likelihood score:

$$s_{ij} = \sigma(W(|h_i^L - h_j^L|)). \tag{6.4}$$

Note that with this approach we are not predicting directed links as we take the absolute value of the difference between hidden state vectors.

In all cases the GNN is trained with Stochastic Gradient Descent (SGD) on the Cross Entropy (CE) loss for both problems, node or edge classification. CE loss is defined as:

$$CE(y') = -(y \cdot log(y') + (1-y) \cdot log(1-y')) \tag{6.5}$$

where y are the ground-truth labels and y' are the predicted scores.

6.3 Experiments

In this section we present the experiments for our method on the benchmark datasets FUNSD [40] and IEHHR [21] for administrative and historical documents respectively.

6.3.1 Datasets

FUNSD

As we introduced earlier, despite the abundance of research on extracting structured information from semi structured documents and the interest in the industry for obtaining a robust solution for the problem there is no universally accepted main benchmark for the task. An obstacle for the advance and refinement of a solution in the field is the confidential nature of the data in which companies need to run such algorithms. Jaume *et al.* [40] intend to unify efforts with a benchmark on this popular problem, reducing it to the tasks of grouping, labeling and linking. The dataset comprises 199 real, fully annotated, scanned forms extracted from the Truth Tobacco Industry Document6 (TTID), and archive comprising scientific research, marketing, and advertising documents of some of the largest US tobacco firms.

IEHHR

Besides testing our approach on modern bureaucratic document dataset we also want to investigate its versatility in even weaker structured documents, such the ones containing in the IEHHR competition dataset [21]. In this chapter we use a modified version of the earlier used database. In this case the word groups are also forming named entities, but restricted to information of members of the family in which marriages are taking place -wife, husband, wife's father, mother etc.- as well as their related locations, occupations or civil states. All entities corresponding to a family member are linked to the name of the corresponding members. Also wife and husband names are linked for each record. An example page with labeled entities can be seen in figure 6.3.

6.3.2 Metrics

The performance of the tasks faced in this work are measured with two different metrics. For the grouping part, since it consists of clustering elements we calculate the Adjusted Rand Index (ARI) [37]

For the tasks of entity labeling and link prediction we calculate the $F1$ score in the traditional way, being the harmonic mean between precision P and recall R.

$$F1 = 2 \cdot \frac{P \cdot R}{P + R}$$

6.3.3 Results

Table 6.1 presents the quantitative evaluation on the three tasks. Note that our model is not using any external data to train our architecture.

Concerning the grouping task in *FUNSD*, we see that the model is able to correctly predict most links between words, despite the vast amount of edges in the k-NN graph. Although it would be ideal, with this approach it is not intended that every single edge is going to be correctly predicted, remind that we intend to cluster the nodes based on densely connected regions with a semantic meaning. In many cases the groups will be correctly predicted despite some of the links between nodes in the are missing, *i.e.* a false negative link is likely to be harmless to the performance on this step as the aggregation is still correct. On the other hand, false positive links create a bigger problem. They may join two groups that should be separated for a proper detection. Using the validation scores during training, we set the threshold τ to the value above which an existing edge is considered a link on the grouping step. Hence, τ has been set to 0.65, and 0.9 for FUNSD and IEHHR respectively, avoiding as much false positives as possible. Predictions on a k-NN graph from a page can be observed in figure 6.4.

Regarding the entity labeling task, we outperform the BERT + MLP approach proposed in the FUNSD baseline [40]. The same task is performed at word level by the pretrained LayoutLM [90]. Their reported results are convincing, however, they are not directly comparable neither to the FUNSD approach [40] nor our current work. Our results follow the original paper, therefore the F1 is calculated

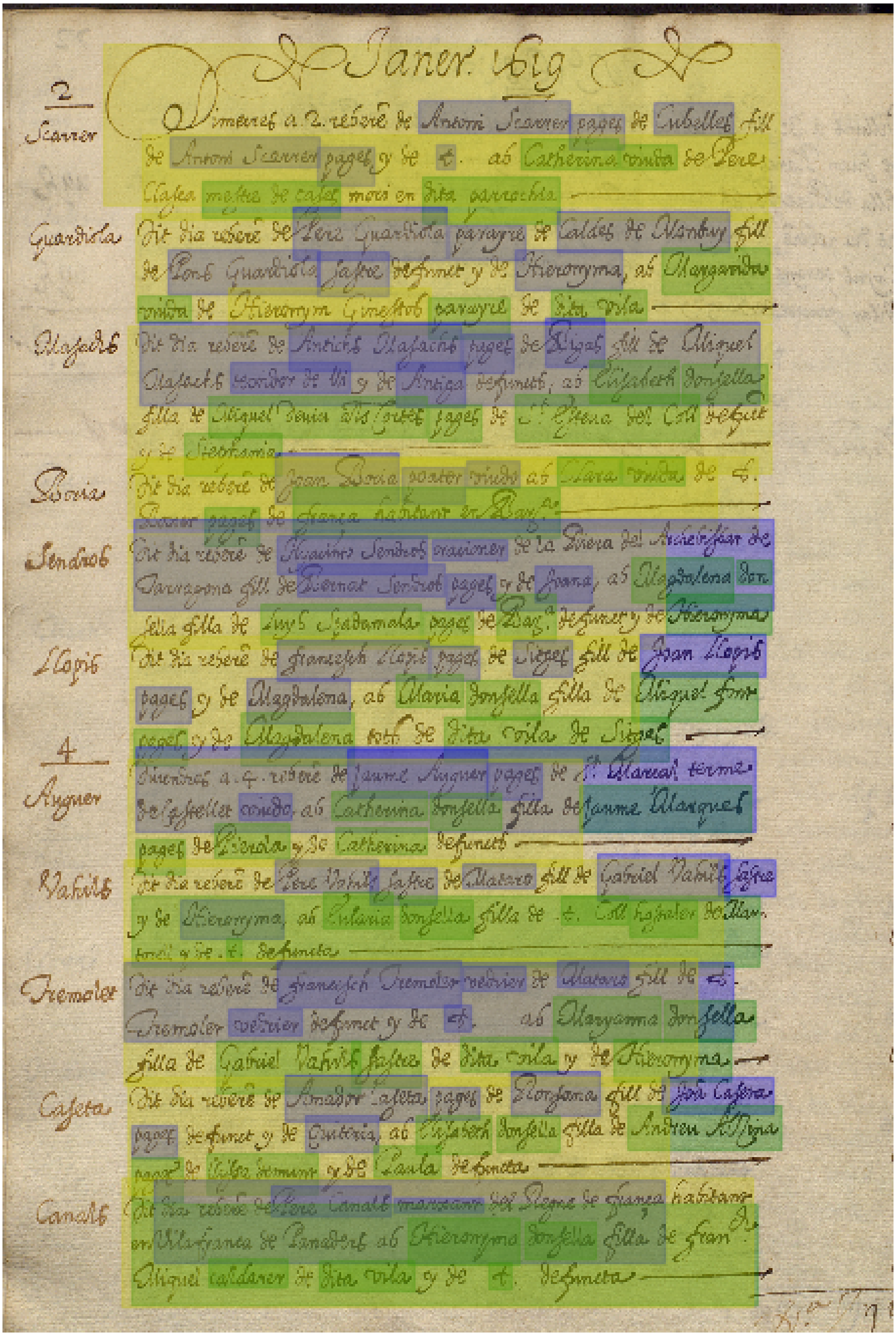
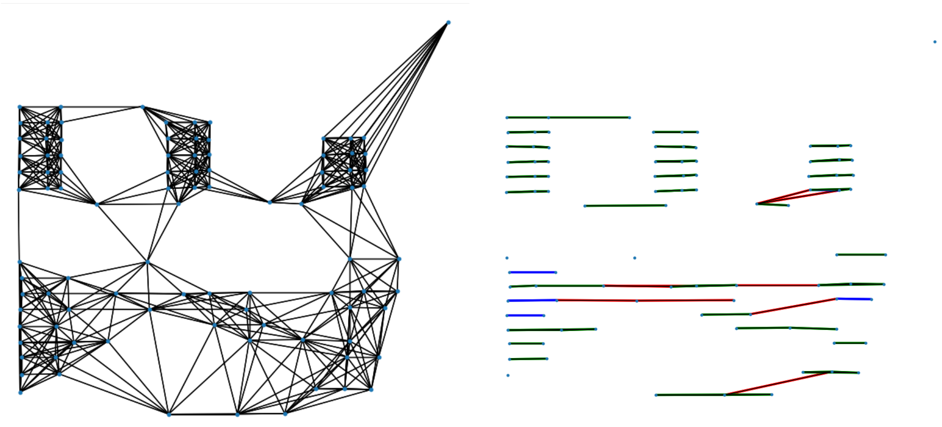


Figure 6.3: Entity label ground truth on a IEHHR page. The amount of words in the groups vary greatly depending on the type of entity.



(a) **(b)**

Figure 6.4: (a) Input k-NN graph fed to the GNN for word grouping on a FUNSD page. (b) Word group predictions on the same document. Green edges are true positives, red are false positives and blue false negatives. We do not plot true negatives and the background to ease interpretation. Node positions are normalized with respect to the page image size.

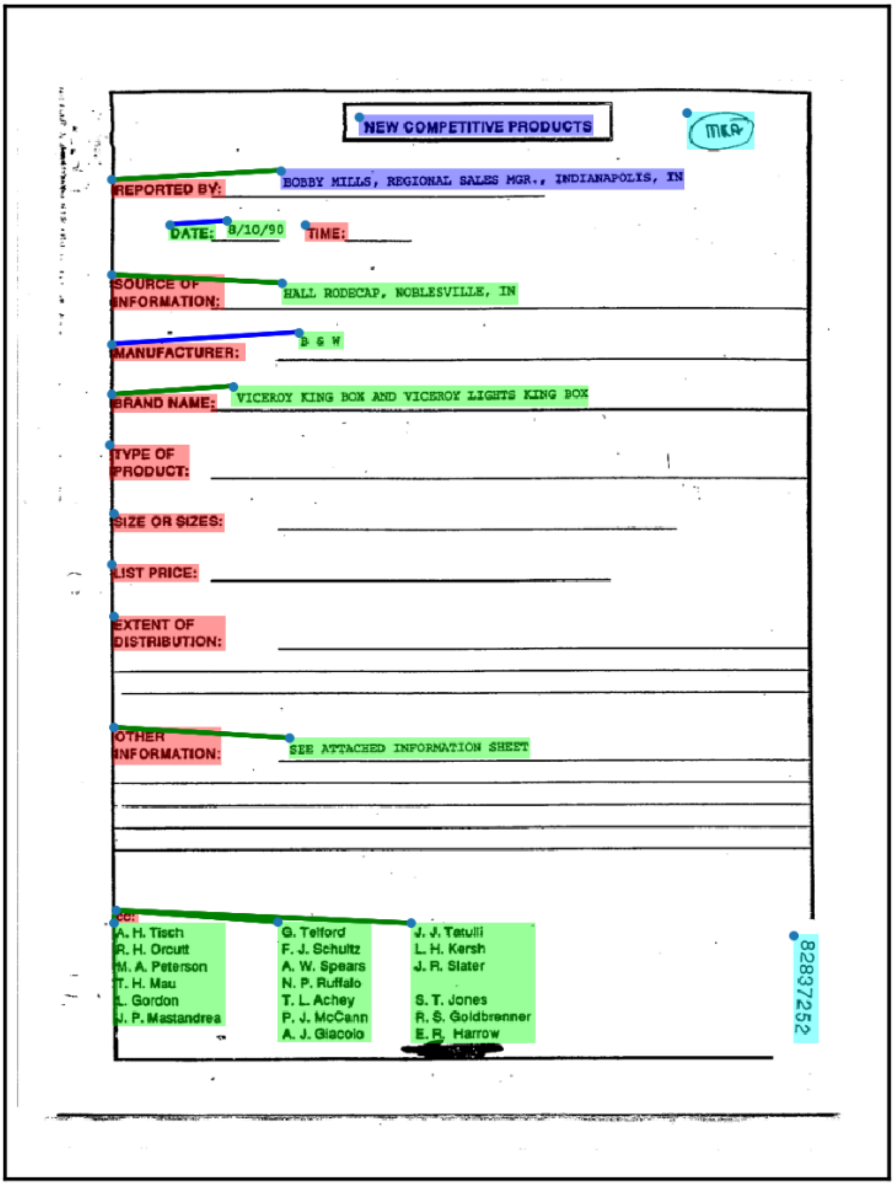
at entity level.

Concerning entity linking, the model performs significantly better than the previously proposed method [40] but with a moderated performance when considering it in a generic context. We are convinced that this could be strongly improved using a dataset with a significant higher amount of training samples.

When observing qualitative results on an unseen page (see figure 6.5) we notice that the model does some wrong link predictions in which a rule restriction based on the content of the entities could give better results. However the scope of this work is to investigate how good a pure learned graph neural model could perform in such a task of finding relationships within the document, without having to classify a layout into a known one but learning to identify pairs of keys and values and other relevant related entities instead.

Regarding *IEHHR*, the grouping model gives an acceptable performance, specially taking into account the strongly regular nature of the paragraphs in each page. Despite this regularity, the difficulty in the labeling part becomes clear, since we have to classify each entity in one of the predefined 20 categories with only 80 pages for training, to which we attribute the low performance in this step. Despite leaving room for improvement our model again gets to successfully solve the linking of entities proving that the approach can also be suitable for this type of task.

[1] Not directly comparable, evaluation at word level



NEW COMPETITIVE PRODUCTS

MKA

REPORTED BY: BOBBY MILLS, REGIONAL SALES MGR., INDIANAPOLIS, IN

DATE: 8/10/90 TIME:

SOURCE OF INFORMATION: HALL RODECAP, NOBLESVILLE, IN

MANUFACTURER: B & W

BRAND NAME: VICEROY KING BOX AND VICEROY LIGHTS KING BOX

TYPE OF PRODUCT:

SIZE OR SIZES:

LIST PRICE:

EXTENT OF DISTRIBUTION:

OTHER INFORMATION: SEE ATTACHED INFORMATION SHEET

cc:

A. H. Tisch
R. H. Orcutt
M. A. Peterson
T. H. Mau
L. Gordon
J. P. Mastandrea

G. Telford
F. J. Schultz
A. W. Spears
N. P. Ruffalo
T. L. Achey
P. J. McCann
A. J. Giacoio

J. J. Tatulli
L. H. Kersh
J. R. Slater

S. T. Jones
R. S. Goldbrenner
E. R. Harrow

82837252

Figure 6.5: Entity linking and labeling predictions on FUNSD. Green and blue lines show true positive and false negative links between entities. Keys, values, headers and other are labeled with red, green, blue and turquoise boxes respectively.

Table 6.1: Results for the three document understanding tasks on FUNSD and IEHHR datasets.

	Word Grouping (ARI)	**Entity Labeling (F1)**	**Entity Linking (F1)**	**External data**	**# Params**
			FUNSD [40]		
[40]	0.41	0.57	0.04	✓	340M
[90]	-	0.79[1]	-	✓	160M
Ours	0.65	0.64	0.39	-	201M
			IEHHR [21]		
Ours	0.65	0.53	0.67	-	201M

6.4 Conclusion

In this chapter we have presented a method to perform named entity recognition and relation prediction in semi structured documents with Graph Neural Networks, bringing promising results in the process of structured information extraction. Our method has been initially designed for administrative document understandig, but we have shown that it can be adapted to other domains, as for example historical manuscripts. The experimental results show that there is still room for improvements, probably due to the reduced size of the open available data sets. For this reason, further research tuning the method and testing on larger data sets could confirm the feasibility of the approach as a generic solution for extracting structured information from semi-structured documents.

Chapter 7

Conclusions and future work

7.1 Conclusions

In this thesis we have studied several neural models pursuing to constitute an optimal method for information extraction from semi-structured documents. We investigated the behavior of these neural models when combining tasks in each of the steps of information extraction. First the processes of recognizing text and named entities, second localizing text and recognizing it, and third, doing all tasks at once with a single neural model. We observed that the benefits and limitations of joining tasks in end-to-end models depend on several factors. Analogously to human learning, when two tasks are closely related or interdependent, excelling at one might ease the learning on the other. Conversely, when two tasks are independent, it might be optimal to focus only on one to achieve highest performance at it. In the first case in which we studied the combination of named entity and text recognition the performance was comparable to other separate approaches but in a balanced situation, putting the two tasks together might make it difficult to isolate the performance bottleneck leaving this way room for improvement. In the case of localization and recognition, there were some qualitative result observations that lead to think that it could be beneficial to join the tasks. This is due to the strong task interdependence, as predicted in Sayre's Paradox, and to the fact that pooling features instead of the input image can be thought as a situation of postponing the final decision until the whole process is finalized, having a broader view of the picture. When joining the three tasks we saw that from start we were missing some powerful task specific resources, such as word embeddings. Still since the visual features are shared with the semantic ones the model can learn the equivalent to a character embedding, but qualitative results were not as good as expected taking in account the difficulty of the task. This lead us to think that for the understanding part, all the effort should be put in that task without mixing the features with the visual part, or at least some kind of architecture improvement should be brought to achieve better performance. Regarding the last part of our work, we explored GNN architectures for the last steps of the process achieving successful results. Our proposed edge score MLP shown to be a plausible method for predicting relationships among document elements allowing to extract the information in a controlled way in contrast with previously explored black box methods in which it is difficult to identify in which situations the algorithm succeeds and where it fails.

7.2 Future Work

Having studied several end-to-end methods for each of the steps of IE from semi-structured documents, a possible continuation line would be to further explore variations of architectures based on multi-head attention as we have done in the last chapter to achieve better performance. A reasonable approach would be to combine to study the combination of multi-modal data, instead of starting from a pure image or pure graph as we have done in our earlier works. Another major barrier in the field is the lack of a solid benchmark dataset to compare different proposed approaches for IE from semi-structured or bureaucratic documents. Due to the legal and privacy issues there have been many similar works tested on different datasets which were not made available to the research community preventing the extraction of a common conclusion and consensus of optimal vs inefficient practices in the field. A large and variate dataset with detailed annotations at different levels of abstraction (word, entity, text region, semantic relationship etc.) with example documents taken from an industrial scenario would quickly allow the document IE field to at the level of other AI fields such as object detection and recognition which are already extremely efficient, high-performing and thereby useful for many real world applications.

List of Publications

Journals

- J. Ignacio Toledo, Manuel Carbonell, Alicia Fornés, Josep Lladós. (2019) Information Extraction from Historical Handwritten Document Images with a Context-aware Neural Model, Pattern Recognition.
- Manuel Carbonell, Alicia Fornés, Mauricio Villegas, Josep Lladós. (2020) A Neural Model for Text Localization, Transcription, and Named Entity Recognition in Full Pages. Pattern Recognition Letters.

International Conferences and Workshops

- Manuel Carbonell, Mauricio Villegas, Alicia Fornés, Josep Lladós (2018). Joint Recognition of Handwritten Text and Named Entities With a Neural End-to-end Model. International Workshop on Document Analysis Systems (DAS).
- Manuel Carbonell, J. Mas Romeu, Mauricio Villegas, Alicia Fornés, Josep Lladós (2019). End-to-End Handwritten Text Detection and Transcription in Full Pages. ICDAR Workshop on Machine Learning.
- Manuel Carbonell, Pau Riba, Mauricio Villegas, Alicia Fornés, Josep Lladós (2020). Named Entity Recognition and Relation Extraction from Semi-structured Documents. International Conference on Pattern Recognition.

Code Available

- **Transcription and Named Entity Recognition (Chapter 3):**
  `http://doi.org/10.5281/zenodo.1174113`
- **Localization, transcription and Named Entity Recognition (Chapters 4 and 5):**
  `https://github.com/omni-us/research-e2e-pagereader`
- **Named Entity Recognition and Relation Extraction with graphs (Chapter 6):**
  `https://github.com/manucarbonell/gcn-form-understanding`

Bibliography

[1] D. Aldavert and M. Rusiñol. Manuscript text line detection and segmentation using second-order derivatives. In *IAPR International Workshop on Document Analysis Systems (DAS)*, pages 293–298, April 2018.

[2] Alexei Baevski, Sergey Edunov, Yinhan Liu, Luke Zettlemoyer, and M. Auli. Cloze-driven pretraining of self-attention networks. In *EMNLP/IJCNLP*, 2019.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*, 2015.

[4] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinícius Flores Zambaldi, Mateusz Malinowski, Andrea Tacchetti, et al. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261, 2018.

[5] Théodore Bluche. Joint line segmentation and transcription for end-to-end handwritten paragraph recognition. *CoRR*, abs/1604.08352, 2016.

[6] Théodore Bluche, Jérôme Louradour, and Ronaldo O. Messina. Scan, attend and read: End-to-end handwritten paragraph recognition with mdlstm attention. *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 01:1050–1055, 2017.

[7] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

[8] Johan Bos, Valerio Basile, Kilian Evang, Noortje Venhuizen, and Johannes Bjerva. The groningen meaning bank. In *Handbook of Linguistic Annotation*. Springer, 2017.

[9] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual*

Workshop on Computational Learning Theory, COLT '92, page 144–152, New York, NY, USA, 1992. Association for Computing Machinery.

[10] Michal Busta, Lukas Neumann, and Jiri Matas. Deep textspotter: An end-to-end trainable scene text localization and recognition framework. *IEEE International Conference on Computer Vision (ICCV)*, pages 2223–2231, 2017.

[11] Manuel Carbonell, Alicia Fornés, Mauricio Villegas, and Josep Lladós. A neural model for text localization, transcription and named entity recognition in full pages. *Pattern Recognition Letters*, 136:219–227, 2020.

[12] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction, 2019.

[13] Laura Chiticariu, Yunyao Li, and Frederick R. Reiss. Rule-based information extraction is dead! long live rule-based information extraction systems! In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 827–832, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[14] Felipe Codevilla, Eder Santana, Antonio M. López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. *CoRR*, abs/1904.08980, 2019.

[15] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[16] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005.

[17] Timo I. Denk and Christian Reisswig. Bertgrid: Contextualized embedding for 2d document representation and understanding. *ArXiv*, abs/1909.04948, 2019.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 4171–4186, 2019.

[19] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *ArXiv*, abs/1603.07285, 2016.

[20] Javier Ferrando, Juan Luis Domínguez, Jordi Torres, Raúl García, David García, Daniel Garrido, Jordi Cortada, and Mateo Valero. Improving accuracy and speeding up document image classification through parallel systems. In Valeria V. Krzhizhanovskaya, Gábor Závodszky, Michael H. Lees, Jack J.

Dongarra, Peter M. A. Sloot, Sérgio Brissos, and João Teixeira, editors, *Computational Science – ICCS 2020*, pages 387–400, Cham, 2020. Springer International Publishing.

[21] A. Fornes, V. Romero, A. Baro, J. Toledo, J. Sanchez, E. Vidal, and J. Llados. Icdar2017 competition on information extraction in historical handwritten records. In *International Conference on Document Analysis and Recognition*, pages 1389–1394, 2017.

[22] Jonas Gehring, M. Auli, David Grangier, Denis Yarats, and Yann Dauphin. Convolutional sequence to sequence learning. In *ICML*, 2017.

[23] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, volume 70, pages 1263–1272, 2017.

[24] Ross B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015.

[25] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016. `http://www.deeplearningbook.org`.

[26] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2014.

[27] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):855–868, 2009.

[28] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 369–376, New York, NY, USA, 2006. ACM.

[29] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. volume 32 of *Proceedings of Machine Learning Research*, pages 1764–1772, Bejing, China, 22–24 Jun 2014. PMLR.

[30] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *CoRR*, abs/1410.5401, 2014.

[31] Tobias Grüning, Gundram Leifert, Tobias Strauß, and Roger Labahn. A two-stage method for text line detection in historical documents. *CoRR*, abs/1802.03345, 2018.

[32] Adam W Harley, Alex Ufkes, and Konstantinos G Derpanis. Evaluation of deep convolutional nets for document image classification and retrieval. In *International Conference on Document Analysis and Recognition (ICDAR)*.

[33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[34] D. Hebert, T. Paquet, and S. Nicolas. Continuous crf with multi-scale quantization feature functions application to structure extraction in old newspaper. In *2011 International Conference on Document Analysis and Recognition*, pages 493–497, 2011.

[35] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997.

[36] Z. Huang, K. Chen, J. He, X. Bai, D. Karatzas, S. Lu, and C. V. Jawahar. Icdar2019 competition on scanned receipt ocr and information extraction. In *International Conference on Document Analysis and Recognition*, pages 1516–1520, 2019.

[37] Laurence Hubert and Phipps Arabie. Comparing partitions. In *Journal of Classification*, volume 2, pages 193–218, 1985.

[38] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *Int. J. Comput. Vision*, 116(1):1–20, January 2016.

[39] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *NIPS*, 2015.

[40] Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. Funsd: A dataset for form understanding in noisy scanned documents. In *International Conference on Document Analysis and Recognition Workshops*, volume 2, pages 1–6, 2019.

[41] Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. Distant supervision for relation extraction with sentence-level attention and entity descriptions. In *AAAI*, 2017.

[42] Lei Kang, J. Ignacio Toledo, Pau Riba, Mauricio Villegas, Alicia Fornés, and Marçal Rusiñol. Convolve, attend and spell: An attention-based sequence-to-sequence model for handwritten word recognition. In *German Conference on Pattern Recognition*, pages 459–472. Springer, 2018.

[43] A. R. Katti, C. Reisswig, Cordula Guder, Sebastian Brarda, S. Bickel, J. Höhne, and Jean Baptiste Faddoul. Chargrid: Towards understanding 2d documents. In *EMNLP*, 2018.

[44] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*, 2017.

[45] Praveen Krishnan, Kartik Dutta, and C. V. Jawahar. Word spotting and recognition using deep embedding. *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 1–6, 2018.

[46] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60:84–90, 2012.

[47] A. Kundu, Y. He, and P. Bahl. Recognition of handwritten word: first and second order hidden markov model based approach. In *Proceedings CVPR '88: The Computer Society Conference on Computer Vision and Pattern Recognition*, pages 457–462, 1988.

[48] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature Cell Biology*, 521(7553):436–444, May 2015.

[49] Kai Li, Curtis Wigington, Chris Tensmeyer, Handong Zhao, Nikolaos Barmpalios, Vlad I. Morariu, Varun Manjunatha, Tong Sun, and Yun Fu. Cross-domain document object detection: Benchmark suite and method. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[50] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyra mid networks for object detection. *CoRR*, abs/1612.03144, 2016.

[51] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and P. Dollár. Focal loss for dense object detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017.

[52] Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2124–2133, Berlin, Germany, August 2016. Association for Computational Linguistics.

[53] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. *CoRR*, abs/1512.02325, 2015.

[54] Xiaojing Liu, Feiyu Gao, Qiong Zhang, and Huasha Zhao. Graph convolution for multimodal information extraction from visually rich documents. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.

[55] Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. FOTS: fast oriented text spotting with a unified network. *CoRR*, abs/1801.01671, 2018.

[56] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2, 1999.

[57] Pengyuan Lyu, Minghui Liao, Cong Yao, Wenhao Wu, and Xiang Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In *ECCV*, 2018.

[58] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, Calif., 1967. University of California Press.

[59] Urs-Viktor Marti and Horst Bunke. The iam-database: an english sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5:39–46, 2002.

[60] Bastien Moysset, Christopher Kermorvant, and Christian Wolf. Full-page text recognition: Learning where to start and when to stop. *CoRR*, abs/1704.08628, 2017.

[61] R. B. Palm, F. Laws, and O. Winther. Attend, copy, parse end-to-end information extraction from documents. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 329–336, 2019.

[62] R. B. Palm, O. Winther, and F. Laws. Cloudscan - a configuration-free invoice analysis system using recurrent neural networks. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 406–413, 2017.

[63] Joan Puigcerver. Are multidimensional recurrent layers really necessary for handwritten text recognition? In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 67–72. IEEE, 2017.

[64] Joan Puigcerver, Daniel Martin-Albo, and Mauricio Villegas. Laia: A deep learning toolkit for htr. GitHub, 2016. GitHub repository.

[65] Mukta Puri, Sargur N. Srihari, and Yi Tang. Bayesian network structure learning and inference methods for handwriting. In *12th International Conference on Document Analysis and Recognition, ICDAR 2013, Washington, DC, USA, August 25-28, 2013*, pages 1320–1324, 2013.

[66] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.

[67] P. Riba, A. Dutta, L. Goldmann, A. Fornés, O. Ramos, and J. Lladós. Table detection in invoice documents by graph neural networks. In *International Conference on Document Analysis and Recognition*, pages 122–127, 2019.

[68] Veronica Romero, Alicia Fornes, Enrique Vidal, and Joan Andreu Sanchez. Using the mggi methodology for category-based language modeling in handwritten marriage licenses books. In *15th international conference on Frontiers in Handwriting Recognition*, 2016.

[69] L. Rothacker, M. Rusiñol, and G. A. Fink. Bag-of-features hmms for segmentation-free word spotting in handwritten documents. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1305–1309, 2013.

[70] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *CoRR*, abs/1706.05098, 2017.

[71] M. Rusiñol, T. Benkhelfallah, and V. P. dAndecy. Field extraction from administrative documents by incremental structural templates. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1100–1104, 2013.

[72] M. Rusiñol and J. Lladós. Logo spotting by a bag-of-words approach for document categorization. In *2009 10th International Conference on Document Analysis and Recognition*, pages 111–115, 2009.

[73] J. A. Sánchez, V. Romero, A. H. Toselli, and E. Vidal. Icfhr2014 competition on handwritten text recognition on transcriptorium datasets (htrts). In *2014 14th International Conference on Frontiers in Handwriting Recognition*, pages 785–790, Sept 2014.

[74] J. A. Sánchez, V. Romero, A. H. Toselli, and E. Vidal. Icfhr2014 competition on handwritten text recognition on transcriptorium datasets (htrts). In *2014 14th International Conference on Frontiers in Handwriting Recognition*, pages 785–790, Sept 2014.

[75] J.A. Sánchez, V. Romero, A.H. Toselli, and E. Vidal. ICFHR2016 competition on handwritten text recognition on the READ dataset. In *ICFHR*, pages 630–635. IEEE, 2016.

[76] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20:61–80, 2009.

[77] J. Sivic and A. Zisserman. Efficient visual search of videos cast as text retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):591–606, 2009.

[78] W. Sui, Q. Zhang, J. Yang, and W. Chu. A novel integrated framework for learning both text detection and recognition. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 2233–2238, 2018.

[79] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003.

[80] J. I. Toledo, Manuel Carbonell, A. Fornés, and J. Lladós. Information extraction from historical handwritten document images with a context-aware neural model. *Pattern Recognit.*, 86:27–36, 2019.

[81] J. Ignacio Toledo, Sounak Dey, Alicia Fornés, and Josep Lladós. Handwriting recognition by attribute embedding and recurrent neural networks. *International Conference on Document Analysis and Recognition (ICDAR)*, 01:1038–1043, 2017.

[82] A. Toselli, V. Romero, M. Pastor, and E. Vidal. Multimodal interactive transcription of text images. *Pattern Recognit.*, 43:1814–1825, 2010.

[83] J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, and A.W.M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 2013.

[84] U. v. Marti and H. Bunke. A full english sentence database for off-line handwriting recognition. In *In Proc. Int. Conf. on Document Analysis and Recognition*, pages 705–708, 1999.

[85] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. 2017.

[86] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[87] H. Wei, M. Baechler, F. Slimane, and R. Ingold. Evaluation of svm, mlp and gmm classifiers for layout analysis of historical documents. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1220–1224, 2013.

[88] Curtis Wigington, Brian L. Price, and Scott Cohen. Multi-label connectionist temporal classification. *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 979–986, 2019.

[89] Curtis Wigington, Chris Tensmeyer, Brian Davis, William Barrett, Brian Price, and Scott Cohen. Start, follow, read: End-to-end full-page handwriting recognition. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[90] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm: Pre-training of text and layout for document image understanding. In *The ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2020.

[91] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3320–3328. Curran Associates, Inc., 2014.

[92] Wenwen Yu, Ning Lu, Xianbiao Qi, Ping Gong, and Rong Xiao. Pick: Processing key information extraction from documents using improved graph learning-convolutional networks. *ArXiv*, abs/2004.07464, 2020.

[93] Peng Zhang, Yunlu Xu, Zhanzhan Cheng, Shiliang Pu, Jing Lu, Liang Qiao, Yi Niu, and Fei Wu. Trie: End-to-end text reading and information extraction for document understanding, 2020.