FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Assignment of master's thesis

Title:	Relative Layout Matching for Document Data Extraction
Student:	Bc. Matyáš Skalický
Supervisor:	Ing. Milan Šulc, Ph.D.
Study program:	Informatics
Branch / specialization:	Knowledge Engineering
Department:	Department of Applied Mathematics
Validity:	until the end of summer semester 2022/2023

Instructions

Consider the task of data extraction from a query document having annotated documents of same relative layout in the database. A relative layout does not have to match exactly due to shifts caused by varying lengths of fields or tables.

Requirements:
1. Familiarize yourself with common structure and layouts of business documents. Review the state of the art in information extraction from structured documents.
2. Prepare a benchmark for the considered information extraction task, which includes various cases of documents with the same relative layout.
3. Prepare baselines for the task, e.g. methods assuming exact placement of values in documents of the same layout. Consider the related work and the existing method(s) within Rossum.
4. Propose methods to efficiently retrieve documents of the same relative layout and to match the corresponding areas, even if the layout match is not exact.
5. Evaluate the proposed methods against baselines on the benchmark(s) from 2.

Electronically approved by Ing. Karel Klouda, Ph.D. on 30 December 2021 in Prague.

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Master's thesis

Relative Layout Matching for Document Data Extraction

Bc. Matyáš Skalický

Department of Applied Mathematics
External Supervisor: Ing. et Ing. Milan Šulc, Ph.D.

May 5, 2022

Acknowledgements

First and foremost, I would like to thank my family for supporting me all the way through my studies. I would like to also thank Veronika for withstanding my terrible sleeping habits and for being supportive and understanding at all times. I know that things are not always easy with me.

Next, I must thank Rossum, the fantastic company and the incredible people that make it. Without Rossum, this thesis would have never existed. I hope that our work will once lead to a world free of manual data entry. Special thanks go to the coauthors of the publication, which was published along with this thesis, namely Štěpán Šimsa, Michal Uřičář and Milan Šulc.

The most important person — the true MVP — that was instrumental to the conception, and the whole odyssey of this thesis is Milan Šulc. He has guided me through the whole journey, always providing great tips and sharing his wisdom. I sincerely appreciate all of his time and energy that went into consulting, brainstorming, and patiently explaining things to me. I have learned a lot. Thank you.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended. In accordance with Article 46 (6) of the Act, I hereby grant a nonexclusive authorization (license) to utilize this thesis, including any and all computer programs incorporated therein or attached thereto and all corresponding documentation (hereinafter collectively referred to as the "Work"), to any and all persons that wish to utilize the Work. Such persons are entitled to use the Work in any way (including for-profit purposes) that does not detract from its value. This authorization is not limited in terms of time, location and quantity.

In Prague on May 5, 2022 . . . . . .

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis

Skalický, Matyáš. *Relative Layout Matching for Document Data Extraction.* Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2022.

Abstract

This thesis explores the field of business document information extraction, emphasizing one-shot learning systems that improve their performance by utilizing a database of previously processed documents. A benchmark to evaluate one-shot information extraction systems was defined and used with a newly created dataset. A novel representation-learning approach to one-shot document information extraction was proposed. For a newly received document, the proposed approach uses learned document representation to first retrieve field representations from similar documents. Retrieved representations are then used to localize information on the newly received document. The proposed method was evaluated and compared against several proposed baselines showing an improvement on fields with high positional variance. The baseline method still achieves better results on fields that remain fixed within the layout.

Keywords key information extraction, one-shot information extraction, neural networks, relative layout matching, business document, contrastive learning

Abstrakt

Tato práce se zabývá oblastí extrakce informací z obchodních dokumentů, přičemž klade důraz na systémy, které využívají již dříve zpracované dokumenty pro rychlou a flexibilní extrakci dat. V této práci byl navržen inovativní přístup založený na učení reprezentací jednotlivých políček v dokumentech za pomocí neuronových sítí. Tento přístup byl vyhodnocen a porovnán se základními přístupy na nově vytvořeném datasetu. Nově navržený přístup funguje lépe na políčkách, která nezůstávají stabilně na stejné pozici v rámci šablony. Základní přístup je nicméně stále lepší na ostatních typech políček.

Klíčová slova extrakce klíčových informací z dokumentů, učení reprezentací, neuronové sítě, obchodní dokumenty, kontrastivní učení

Contents

List of Figures

Introduction

We live in a world full of documents. Digital, printed, and even handwritten documents are among the main mediums of communication. Communication between individuals, but also between businesses as well as government institutions. Millions of invoices, tax forms, letters, legal contracts, orders, resumes, and financial reports are sent every day. These documents are optimized for human readability and understanding. However, in the modern, digital-first setting, we strive to automate as many repetitive processes and tasks as possible. And the ability to reliably extract structured information from incoming documents is one of the cruxes of achieving fully automated document communication.

The day-to-day work of large companies includes the processing of thousands of documents. Manual data entry is an expensive process that requires substantial human labor. Elimination of this burden has the potential to save precious resources and to allow people to focus on different, perhaps more creative aspects of their jobs.

The task of extracting information from documents falls under the broad category of document understanding. Even nowadays, document understanding is a complex and challenging task that has not yet been sufficiently solved. This is not only due to the nature of the input data but also due to the heterogeneous nature of the documents that come in a variety of languages, templates, formats, and mediums. This huge variability makes document understanding very challenging. Well beyond the "just OCR the document" as perceived by the wide public.

Companies like Rossum[1] strive to create a world free of manual data entry. This thesis aims to help with the research of this still-prevalent problem, helping to make the world a better place where manual data entry is a thing of the past.

[1] `https://www.rossum.ai/`

Thesis Goals and Structure

State-of-the-art in the document information-extraction tasks — described in Section 2 — are supervised deep learning models. However, their training requires substantial amounts of annotated data, expensive dedicated hardware, and most importantly, it takes time. Even though the information extraction systems are trained with a generalization ability in mind, they often fail when a document from an unseen template is presented to the model.

Over time, a machine learning system for document information extraction processes a large number of different layouts. After some time, some of the processed documents will be similar, with some of the templates repeating. Naturally, this fact can be exploited when a new document is received. A system that that utilizes a database of already-processed documents can be used to the aid document information extraction.

An important aspect to consider is the system's ability to reuse the knowledge from documents where the layout does not match exactly (relative layout) and for fields that do not stay at the fixed location within the document. This secondary system can be used in tandem with powerful (but less-flexible) deep learning models. Possibly bridging the gap presented by the data distribution shift that inherently happens between retrainings.

The thesis is structured as follows: Chapter 1 introduces reader to the world of business documents. Chapter 2 then presents an overview of the shattered landscape of document understanding research. Special attention was put on the approach of learning-by-case. The overview is accompanied by an exhaustive research of publicly available datasets for information extraction from business documents. Together with the thesis, the review of related work, task definitions, datasets and benchmarks have been addressed in a position paper [12] (in review, submitted on CLEF[2] conference, see Appendix C). Chapter 3 formally describes the presented problem and proposed evaluation benchmark, which can be further reused for research. Since no suitable dataset was publicly available, a large, albeit private dataset was manually created as a part of this thesis. Chapter 4 describes the proposed baselines, which were also evaluated on the newly created dataset. A more sophisticated approach inspired by contrastive learning is also described, implemented, and compared with the baseline, which turned out surprisingly strong. Chapter 5 then presents the reader with the results of the proposed experiments. Contributions and Future work are discudded in Chapter 6.1.

[2]Conference and Labs of the Evaluation

Chapter 1

Documents, their Structure and Origins

Saund [13] defines a document as "information presented in a format for human reading". We can immediately see the issue regarding the automated data extraction – documents simply were not designed to be readable by computers but rather to serve as a medium of communication between humans. Furthermore, business documents typically do not have any fixed layout, language, currency, fonts, images or even number of pages [14], which complicates the processing even further.

Even though we might think that the exchange of documents is nowadays mostly digital, a study by Ardent Partners [15] reveals that 49.8% of invoices is still sent manually on paper. Invoice processing comes with a significant burden on the recipients – the cost to process a single invoice was estimated [15] to be over $10 in 2020.

1.1 Printed and Digital-Born Documents

In order to digitally process a printed document, it first must be converted into a digital representation. This is commonly done by scanning or taking a picture of the document. OCR (*Optical Character Recognition*) engines are then employed to add the missing text representations. Extracted images, texts and other data are then stored in a transport format such as PDF.

PDF (*Portable Document Format*) has been widely used for both scanned and digital-born documents since its initial release in 1993. As described by Han and Wan [16], "PDF/A file can be a structured, self-contained and self-described container allowing a simpler one-to-one relationship between an original physical document and its digital surrogate". The PDF is, however, primarily designed for visual representation and ease of display rather than for ease of data extraction.

When faced with the struggle of automated document understanding, many people wonder why electronic formats that contain structured, machine-readable information (for example, as a part of an invoice's metadata) are not used. One could argue that since the costs associated with the document processing lie on the party that received the document, the issuer lacks the incentives to make the document processing easier. While there have been attempts to use structured formats such as *Electronic Data Interchange* (EDI) or *EXtensible Markup Langage* (XML), they have never became widely used for business communication [17].

1.2 Structured and Semi-Structured Documents

Business documents within a document type typically follow a similar structure and logical layout. This helps the people working with the documents to quickly localize and extract the necessary information. Our focus lies on the information extraction from (semi) structured business documents. Generally speaking, we classify documents into categories based on their appearance:

Structured documents have a fixed format that does not change. Example of structured documents might be unified tax forms. Since the data is generally simple to locate, a template-based zonal OCR can be utilized to reliably extract the information [13].

Semi-structured documents follow similar general structure, but the locations and visual features of the data within the document might change. Example of a semi-structured document would be an invoice.

Unstructured documents do not follow any predefined structure making the automatic data extraction a challenging task.

It is also important to distinguish *machine readable* — the ability to extract textual information from *machine-understandable* — the ability of a machine to extract relevant information with the target task in mind [14].

Business documents come with several characteristics that distinguish them from other document categories. In particular, certain document types, such as invoices, always contain predefined types of data. They are often also structured in an unified manner. Since the assignment mentions "information extraction from structured documents", it is worth to clarify, that with respect to the business document types mentioned above, this includes both structured and semi-structured documents. It is also worth noting that the definition of a structured document varies across the literature [17]. Holecek [14] describes structured documents as documents whose structure is clear and understandable to a human working in a given field.

1.3 Invoices

Invoice is a commercial document that records a transaction between a buyer and a seller. The vendor usually issues invoice after delivering a product or after providing a service to the client. Invoices serve as a way to track the date of the transaction, outstanding balance and the involved goods/services [18]. Invoices also have a legal value. Companies are often legally obliged to process and archive invoices for prolonged time periods [17].

It is no question that invoices play a vital role in the daily business communication. Automation of invoice data extraction is not only an interesting research task, but also a real problem that businesses face. Reliable automation of invoice processing has a great potential to cut associated costs and save valuable human work.

We will further describe a structure of an invoice as it is related to the dataset used in this thesis. Example of an annotated invoice (invoice that has already been processed) is shown in Figure 1.1.

Invoices, representing (semi)structured business documents, typically follow a similar structure: the vendor and billing information is located on the upper part of the page, followed by *date due* and *invoice number*. We call all fields that do not belong to any tabular structure *header fields*. Invoices often include a tabular breakdown of billed products or services which includes descriptions, quantities, and billed rates/prices. This table is commonly called *line items*. It is worth noting that line items can span multiple pages. An invoice summary (also called *tax details*) typically lies below the line items table. It sums the prices of all billed services with the taxes of different rates. Total amount to be paid then sums the base price with the tax amounts.

European Union Value Added Tax (VAT) directive [19] article 226 formally defines the data that is required to be present on a VAT invoice. Most importantly date of issue, invoice ID, VAT ID, customer's VAT ID, customer name and address, quantity and nature of provided goods/services, date of supply, taxable amount, applied VAT rate, and VAT amount to be paid.

1.4 Glossary

The field of document understanding comes with its specific vocabulary. The following overview of the document understanding terminology aims to explain some of the commonly used keywords and phrases:

field Localized piece of information on the document. It is specified by type, and a bounding box. Field is often used as a short form of *header field*, a field that does not belong to any tabular structure.

field type Category of a field. Defines a semantic name of the information within the field. For example *date due*, *amount total* or *invoice id*.

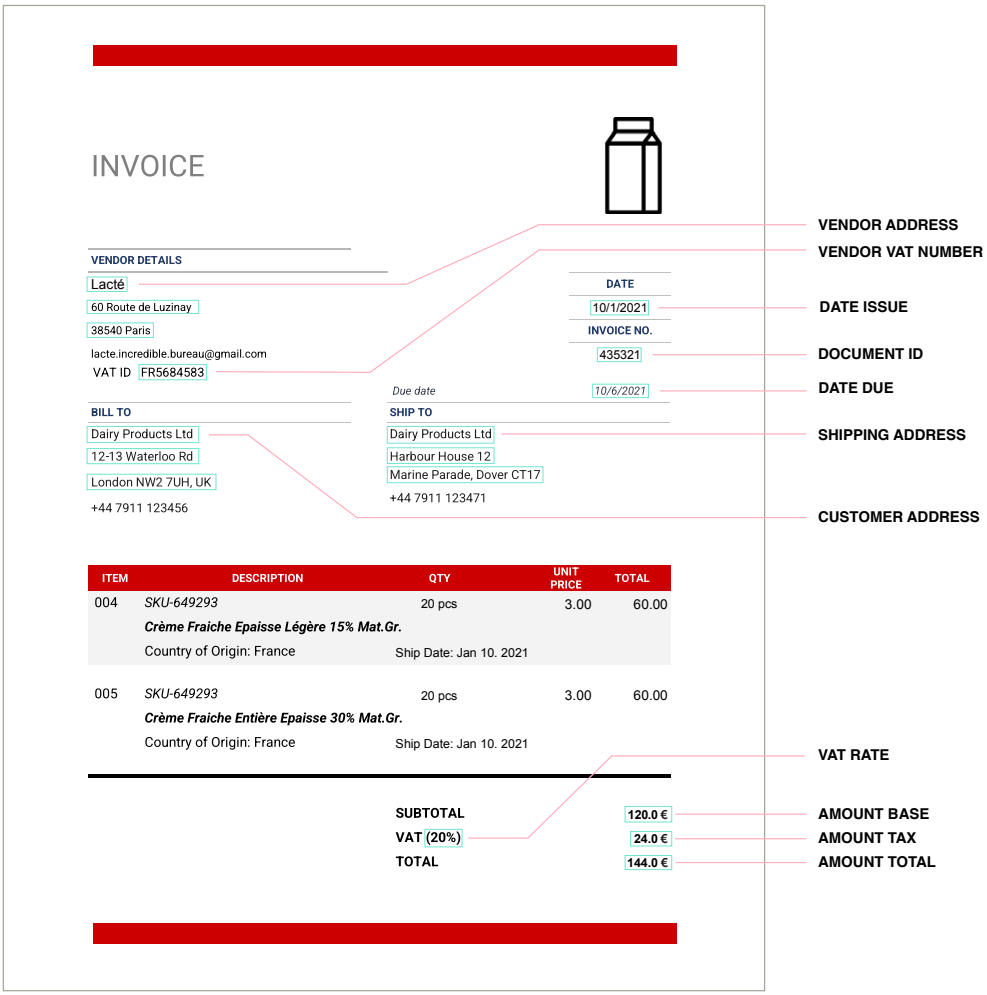


Figure 1.1: Example invoice with annotated fields and their types. The tabular structure (often called *line items*) is not annotated in this case. *Vat rate*, *amount base*, *amount tax* and *amount total* belong to a section that is commonly called *tax details*.

line items Items in a tabular structure which contains descriptions, quantities and prices of the goods or services provided. A line item can be viewed as a composition of multiple fields.

document layout Two documents that belong to the same layout were produced from the same template. Terms *template* and *layout* are used interchangeably within this work.

field annotation Pair composed of bounding box and field type. Describes a piece of information located within the document.

Chapter 2

Related Work

2.1 Document Understanding

First Optical Character Recognition (OCR) systems date back to 1980s, so the task of Document Recognition can be considered pretty mature. Yet, there are still many unsolved challenges [20]. Document processing still requires manual human labor and blocks business process automation.

The task of document data extraction lies under the wide category of Document Understanding. *Document Understanding (DU)* is a task that aims to extract human-understandable information from documents and to present it in a machine-readable format [21]. There are several research tasks related to document understanding, including:

Key Information Extraction (KIE) is a task which consists of extracting information of number of key fields (such as *amount total*) from semi-structured documents [3, 22]. It can be further distinguished into end-to-end *Key Information Extraction* (KIE) which aims to extract the target information regardless of its position, and *Key Information Localization and Extraction* (KILE) where the task is to also locate the extracted information within the document. Both were defined in [12].

Visual Question Answering (VQA) also aims to extract information, but the queries are presented as natural language questions [23].

Layout Analysis is a task of recognizing individual components that documents are built of. For example titles, paragraphs, tables and other page regions [20]. It can be also formulated as page segmentation and region classification [24].

Document Classification is the process of classifying a document into a predefined set of semantic types characterized by similarity of expressions, style form or contents [25].

Tightly related to Document Understanding is *Optical Character Recognition (OCR)* – process that aims to convert printed text and images into computer readable form [26]. Data extracted in Document Understanding can be used for *Robotic Process Automation (RPA)*. The goal of RPA is to automate the human work related to automation of business processes dealing with unstructured data [27].

2.1.1 Key Information Extraction

The challenge of automated data extraction is as old as the digital document itself. In case of invoice processing, most of accounting software require to extract several key fields such as *amount total*, *date due* and *vendor id*. The information extraction systems can be divided into 2 categories [28]:

Knowledge engineering systems utilize handcrafted rules and manually created templates in order to extract information from documents. These systems are flexible and simple to understand, but they require significant human labor to set up and maintain. This category includes the traditional template-based systems described in Section 2.1.1.1.

Trainable systems try to avoid the manual engineering by automatically learning the extraction rules. They often aim for high generalization and minimal manual work. We can divide the trainable systems further, into models that are defined explicitly, or by a set of training documents [29], such as many of the one-shot learning approaches described in Section 2.1.1.2.

The following section summarizes notable contributions to the document understanding and key information extraction tasks. A special emphasis was put on one-shot information extraction: systems that in one way or another utilize a database of already-processed documents to quickly learn from new annotations.

2.1.1.1 Traditional Data Capture

Before diving deep into the current methods, it is worth understanding the legacy template-based extraction approach. Especially since many related works draw inspiration from the aspects of this technique. Example of a traditional data capture service is Docparser[3].

An extraction template is created for each document class or vendor-specific document. This template describes rules that are used to extract

[3]Free trial is available at `https://docparser.com/`

the key information from vendor's documents. These rules can be regular expressions[4] as well as absolute positions of the elements within the documents of the given template.

When a document is received, it is first processed and converted into a standardized format. Printed documents are therefore scanned, and OCR system is applied to extract and add the missing text layer into unified transport format such as PDF. Overall, prediction consists of three steps:

1. Preprocess the document. Apply OCR if needed.
2. Classify the document to select extraction template.
3. Extract the data using the rules defined in selected extraction template.

The first problem of this approach is the selection of the extraction template. This can be done using visual features and/or information extracted from the document texts. But keep in mind, that the document layouts might change. Also, documents can be scanned or even photographed. Errors in the texts extracted by OCR can also hinder the text-based classification.

The second issue is the application of the extraction rules (given they exists). As previously mentioned, documents can be skewed, shifted, photographed at an angle, contain mistakes, drawings and other visual noise. Layouts can contain variable-sized elements; they might evolve over time or even change for different languages.

But most importantly, the process of creating an extraction template is labor-intensive and requires qualified work. For some cases of highly structured and standardized documents, this technique can be very effective. However, this is not the day-to-day reality for most companies dealing with incoming documents, especially invoices. Each vendor typically uses a tailor made or at least customized template, which makes template-based systems very expensive and impractical in the long run.

smartFIX: A Requirements-Driven System for Document Analysis and Understanding The *smartFIX* [1] – short for smart For Information eXtraction (2002) – was developed to extract billing information from the domain of medical bills. It represents a typical example of a template-based information extraction system.

As visualized in Figure 2.1, smartFIX was designed with the human operator in mind, aiming to reduce the human work. The document is first deskewed and preprocessed to correct for visual defects. Next step is classification, which tries to match the document with an existing template from the database of manually-created extraction templates. This matching is done

[4]Regular expression that matches the "TOTAL" prefix TOTAL\s+([d,\.]+) could be used to extract amount from invoice on Figure 1.1.

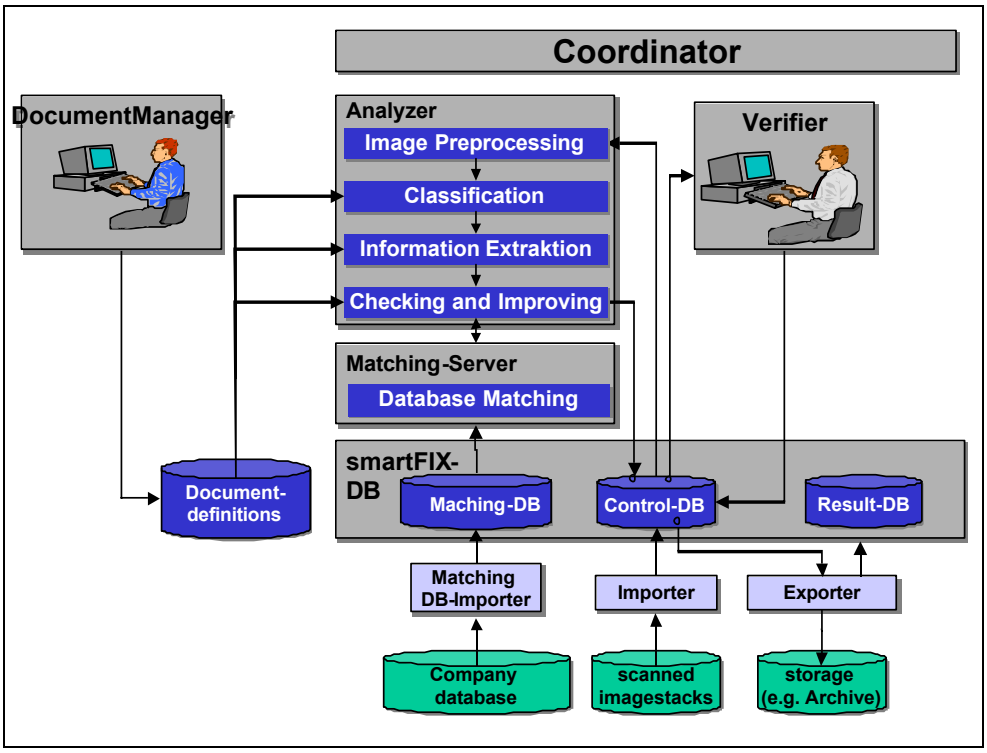


Figure 2.1: Overview of the smartFIX [1] system. Example of a legacy template-based document information extraction approach. Document manager creates the extraction templates (document definitions) which are then used for document classification and information extraction.

using layout similarity and other document features such as user-defined patterns. After the template is detected, it is used for information extraction. Otherwise, the user is prompted to design a new template.

2.1.1.2 One-shot Information Extraction

Document structure often varies significantly across different document types. But documents also vary greatly within their type. There are thousands of different invoice layouts, and vendors often further adjust templates to their own style. This was always the Achilles' heel of traditional template-based systems – manual creation and maintenance of a variety of extraction templates does not scale well.

Systems without the possibility of fast re-training are at risk of degraded performance when faced with a shift in the incoming data distribution [2]. This might be caused by receiving a vendor that was not in the training set, or by a change in the existing template.

One-shot learning systems are designed to quickly adapt to the changing

data distributions either by directly using a database of processed documents, or by iteratively improving the extraction models with each processed sample. Authors call this one-shot template matching [5], case-based reasoning [2], or configuration-free information extraction [28]. This includes systems that lookup similar documents in the database [5, 28], as well as systems that iteratively build and refine a representation of a document class [2, 29, 3, 4].

A lot of emphasis was given to studying existing approaches in this domain, as it relates to the task of this thesis. One-shot learning systems are motivated mainly by two reasons:

- They try to improve the traditional document template-based extraction methods by automatically classifying documents, or by implementing a logic to efficiently construct an extraction model given example documents. These approaches aim to reduce the manual work of crafting templates and to make this process trainable.

- Second motivation is to use these systems to aid larger deep-learning models that are slow to retrain, and inherently not robust against changes in the incoming data distribution.

Analysis and Understanding of Multi-Class Invoices Authors of [30] (2003) note, that when the processed documents can be grouped into a smaller set of classes, a small set of invoices can be used to obtain a reliable knowledge for document understanding. Invoices can usually be clustered into classes according to issuing company or institution.

The system builds the document structure in a bottom-up approach, finally modeling it as a set of horizontal and vertical lines, logos and text boxes. After the texts are extracted from the document, this system tries to interpret its content by combining the intra- and inter-class knowledge. If the document class is known, system tries to enhance its prediction by using its class-related knowledge. Otherwise it is interpreted using generalized class-independent extraction patterns. Extraction is done either by looking at relationships between keywords in the document, or by reusing the absolute positions of fields.

Case-based reasoning for invoice analysis and recognition (2007) [2] also actively reuses the previous data. If a similar document was already processed, it is looked up in the database and the existing document is used to aid with information extraction. If such document does not exist or it is not identified, a generic model is used to extract the data.

This paper is very related to the task of this thesis as it identifies the studied problem: documents within the same class have the same relative positions, but their absolute positions vary from one document to another

depending on the varying sizes of elements in documents. This is nicely visualized in Figure 2.2.

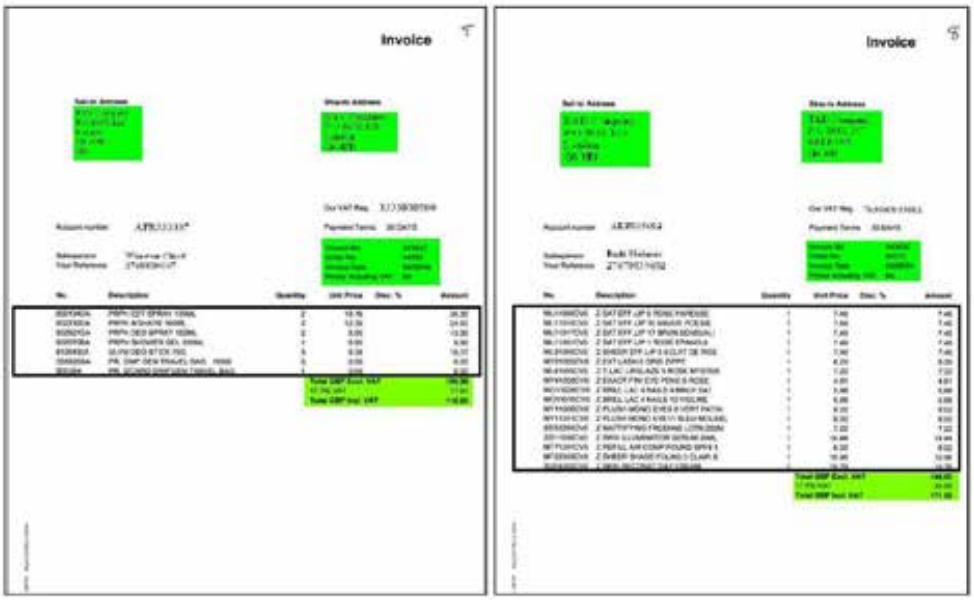


Figure 2.2: Variability of documents from the same layout [2]. Even though the layout is shared, the absolute positions match only for some field types.

The whole extraction process consists of first extracting indices — keywords and their spatial relationship or table rows — from the document. The document is modeled as a graph of such entities. A graph similarity, specifically method called graph probing [31] is used for similar-case retrieval. The information extraction is done by identifying keyword structures that relate to the target values.

A probabilistic approach to printed document understanding (2010) [29] leverages a memory bank of already-annotated documents to enable efficient multi-page document understanding. Document retrieval is based on spatial density of black pixels is the documents.

Document is represented as a set of blocks. Each block consists of position (x, y, page), its size and textual content. Extraction model consists of set of rules, exactly one rule for each target field. A rule might depend on several blocks as some values might be spread across the document. Each rule is composed of its cardinality, matching probability and extraction function that processes the localized values. The model building consists of generating a set of probabilistic rules given a set of documents.

Field Extraction from Administrative Documents by Incremental Structural Templates (2013) [3] deals solely with the task of transferring knowledge from a document of known vendor given relevant samples retrieved from a database.

For each vendor, a structural model is created for all fields that are marked by user for extraction. Each field is modelled as a center of a star graph that connects it to all the other words within the document. This is visualized in Figure 2.3. Once a new document from same vendor arrives, the previously created extraction template is used to detect the target field on the invoice by matching words that appeared on both documents. Once they are identified, the stored relative relationship is applied backwards to reconstruct the position of the target field. To combine the predictions from all matched words, a voting scheme is used to select the target bounding box.

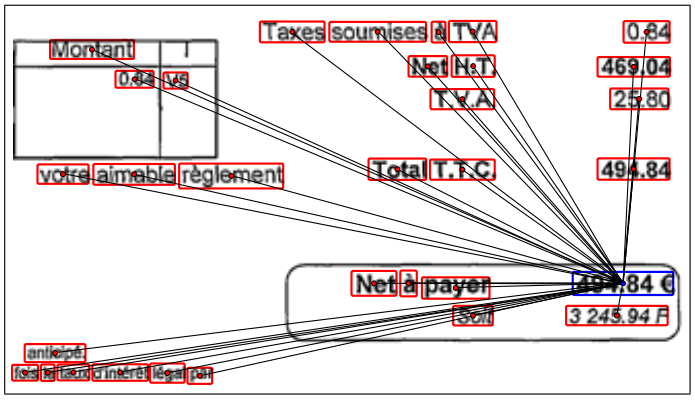


Figure 2.3: Representation of a target field as a center of star graph that captures relationships to all words on the document [3].

The implementation is extended to facilitate learning from multiple documents in order to iteratively improve the structural model as more documents of the same template are registered. A weight is assigned to neighboring words using a weighting scheme similar to tf-idf[5]. This approach is based on the observation that:

- Words that appear multiple times in same document are less informative.
- Words that are always present on the documents from the same provider are more discriminative.
- Words which are close together are more informative as they capture local context.

In 2018 authors extended [4] their prior work by studying the proposed algorithm in depth, and by addressing a common problem among invoices –

[5] `https://en.wikipedia.org/wiki/tf-idf`, Accessed on 03.04.2022

the intra-class layout variability of documents. The errors often come from two sources: human labeling error when the fields are inconsistently labeled, and vertical shifting context – when insertion of text lines invalidates the relationships between most fields. This is shown in Figure 2.4.

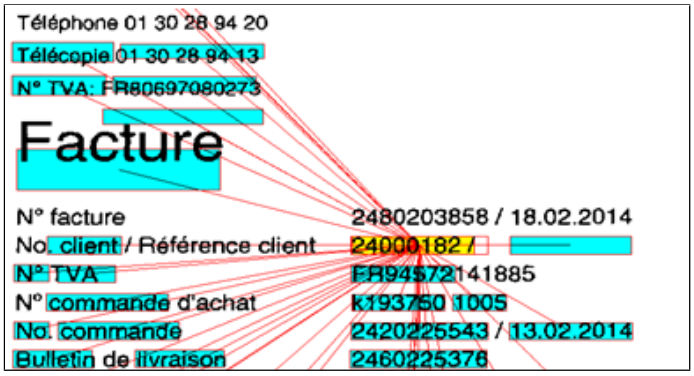


Figure 2.4: Learned spatial relationships are invalidated when a new line of text is inserted [4].

Authors note, that the documents are often organized in a Manhattan structure. And that the fields are often prefixed by the same keyword on the same line. In order to improve the algorithm on document classes with vertical shifting context, the previous weighting scheme is extended by extra term that increases weights of neighboring words that are vertically or horizontally aligned.

Intellix - End-User Trained Information Extraction for Document Archiving (2013) [28] is a commercial solution by DocuWare[6]. Intellix combines text and local features to identify documents stored in a local search engine. Its goal is to extract 10 commonly used fields from documents to enable efficient document archiving. Whole system was designed to instantly adapt to new types of documents by searching for similar documents.

Template detection is based on the authors previous work [32]. The texts are first extracted using an OCR. Each wordbox is appended with a positional information that consists of quantized x and y coordinates. Wordboxes are indexed in Apache Lucene[7] database which allows for efficient and fast k nearest neighbor lookups. This approach works better than a purely visual retrieval based on binarized document images presented in [32].

5–10 most relevant documents are retrieved to aid with information extraction. Intellix generates different kinds of extraction rules to extract the target fields. Fixed-position fields are analyzed by the *Template-based Indexer*. For fields that change their position based on the relative layout, a *Position-based*

[6] `https://www.docuware.com`
[7] `https://lucene.apache.org`

Indexer predicts the field positions based on spatial relationships between pairs of relevant words.

One-Shot Template Matching for Automatic Document Data Capture [5] (2019) presents a simple framework for automated document capture using a database of already-annotated documents. This process consists of three steps: template matching, region proposal and final area selection. To retrieve the most similar document, a combination of visual and text similarity is used. For visual similarity, the document image is decomposed using Singular Value Decomposition (SVD). Cosine similarity of the Σ diagonal matrices combined with fuzzy editing distance between the document texts is used for retrieval.

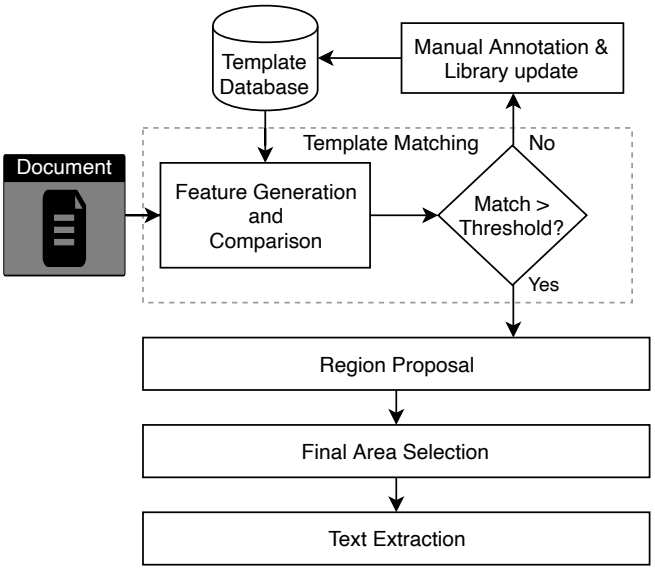


Figure 2.5: Overview of the one-shot information extraction system proposed by Dhakal et al. [5]. Document is retrieved based on text and visual similarity. Each field from the source document is transferred to the incoming document by correlating its visual representation against the incoming document.

Once the most-similar document is retrieved, each annotated field from the source document is visually correlated with the target document to receive approximate region proposals of its location. To further adjust the position within the proposed region, common texts between the source annotation and proposed area are used to precisely resize and position a bounding box on the target document. Once the bounding box is found, OCR engine is used to extract the information.

Learning from similarity and information extraction from structured documents [14] (2021) Holecek experimentally verifies that having an access to a database of previously-annotated documents boost the performance of information extraction as compared to a pretrained neural network on the same dataset. Given access to the source page, target (query) page and annotations of the source page, the task is to classify all of the word-boxes on the target page.

Each page is represented as a graph of word-boxes along with other features. Triplet-loss architecture is compared with a pairwise classification approach and also with transformer-based query-answer architecture. Having access to the similar documents has boosted the performance of the previous approach solution by 8.25% which is a significant improvement.

Major difference between the presented paper and this work is that the utilized approach uses precomputed embedding vectors for a lookup of the nearest neighbors and only deals with the information transfer between documents. This task was formulated as multi-label multi-class word-box classification.

2.1.1.3 Deep Learning Approaches

The recent improvements in graphical processing units (GPU), novel training techniques and large datasets has enabled the deep learning to be used for efficient document information extraction. Early deep learning approaches were based only on single modality, such as text in CoudScan [6]. Later approaches [7, 33] combine multiple modalities. This is natural, since the document structure and the spatio-visual relations are often crucial for document understanding. The information extraction from visually rich documents (VRD's) belongs somewhere between NLP, Computer Vision, and Layout Analysis [34].

To utilize the layout information, the document texts can be organized into structures that preserve their spatial relationships. These approaches include representing the document as a grid (CharGrid [35], BertGrid [36]) or as a graph structure [37, 14, 38] utilizing graph neural networks (GNN). Alternative approach as proposed by Majumder et al. [8] is to use 2D positional embeddings and attention mechanism.

Recently, we have seen a wide adoption of Transformer [39] based architectures for document understanding [34, 40, 41, 42] which typically combine the text and image modality. Transformer-based approaches also rely on unsupervised pre-training using large datasets, similar trend to what we have recently seen in the recent state of the art language models [43].

Following subsection describes CloudScan [6], one of the early deep-learning approaches for document information extraction. The second mentioned approach is Attend, Copy, Parse [7], one of the first multi-modal approaches for end-to-end document information extraction using deep neural networks. The last presented approach [8] uses representation learning for document information extraction using the previously mentioned attention mechanism.

CloudScan - A configuration-free invoice analysis system using recurrent neural networks (2017) [6] is a commercial solution by Tradeshift developed for extraction of structured information from unstructured invoices. CloudScan does not rely on the templates of invoice layouts, instead, a single global model is trained to generalize on unseen invoices. While the system can detect 32 types of fields, performance is reported only on a fixed subset of 8 fields. Optical character recognition (OCR) is ran first. Extracted texts are further processed and used to generate n-gram features. N-grams are then classified by a recurrent neural network. Final predictions are composed by combining the classes of the n-grams that make the words. Overview of the whole process can be seen in Figure 2.6.

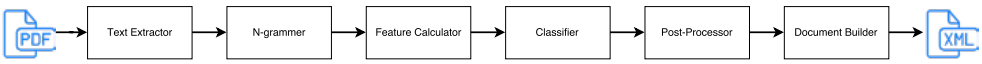


Figure 2.6: Overview of steps involved in the document processing of CloudScan [6] Engine.

It is worth noting that this early approach to document understanding with deep learning does not utilize any visual features from the input document. Also, the reliance on the OCR engine makes it susceptible to OCR errors which can hinder its performance.

Attend, Copy, Parse - End-to-end information extraction from documents [7] (2019) presents a solution to the end-to-end information extraction task. The system is trained directly on the end-to-end data with missing world-level annotations. Attend, Copy, Parse uses a multimodal CNN architecture that combines a subsampled (128 × 128px) image with the text modality extracted by an OCR engine.

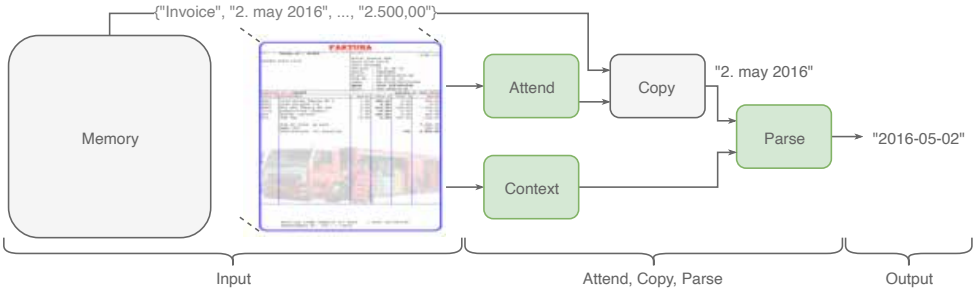


Figure 2.7: Overview of the Attend, Copy, Parse [7]. System utilizes multimodal approach that combines the image information with memory bank of texts extracted by OCR.

Overview of the extraction module is shown in Figure 2.7. The main idea is to store the extracted texts in a matrix of a same shape as the (downsampled) input image. The attend module is implemented mainly by dilated

convolutions. Copy module is responsible of concatenating the textual features with the image features. The parse module finally extracts the data from the document. A separate model was trained to extract each of the 7 target fields.

Representation learning for information extraction from form-like documents [8] (2020) is the first representation learning approach for document information extraction. For each extracted field type, the system first generates multiple candidates from the document texts obtained by OCR. The creation of candidates for each target field type is done using cloud-based entity extraction service.

Each of the candidates is then combined with a representation of its surroundings on the document and projected into an embedding vector. This embedding is then compared with a set of trained prototypical embeddings that represent different field types. Final classification of the candidate is done using cosine similarity over the predicted representations. The neural scoring model is shown in Figure 2.8.

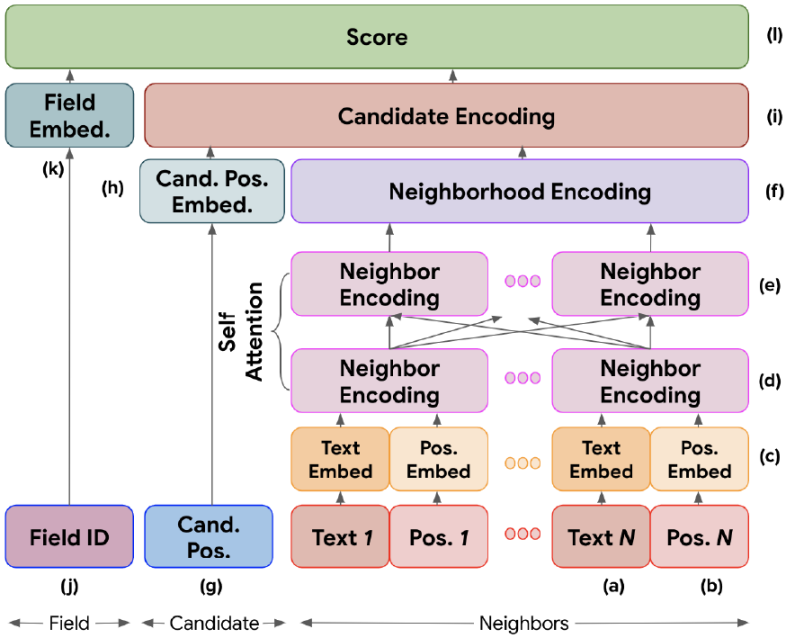


Figure 2.8: Neural scoring model as proposed by [8]. Each candidate is embedded into a representation that includes its neighborhood encoding. This representation is then used to predict a similarity measure against a pretrained representation of a field type.

2.1.2 Visual Question Answering

Question Answering (QA), also known as Machine Reading Comprehension (MRC), is a common information retrieval and NLP task. The goal is to build a system which automatically answers human-posed questions in natural language. Many classic NLP tasks can be reformulated as QA (e.g. machine translation, summarization, natural language inference, sentiment analysis, semantic role labeling, relation extraction, to name a few) [44, 45]. Similarly to Key Information Extraction, QA can extended to incorporate other modalities, such as visual information in Visual Question Answering (VQA) [46]. A VQA system could therefore be used for document KIE by asking natural language questions.

In its simplest form, a document can be represented as a text obtained from OCR. Approach [47] proposed by Damodaran et al. uses ensemble of pre-trained language models by asking natural language questions like "what is the total amount" to extract the key information from documents.

2.1.3 Layout Analysis

Document Layout Analysis (DLA) is typically defined as an object detection problem: given a document/page, find the minimum bounding boxes (or other area representation [48, 49]) covering different layout elements such as *Paragraph*, *Title/Heading*, *Table*, *Figure* or *Caption*.

2.1.4 Document Classification

Document class can be defined as a set of documents, that share similarities in expressions, style, form or contents [25]. Document classification has a wide range of use-cases including document retrieval and filtering for downstream tasks in document analysis. Survey by Chen and Blostein [50] provides a great overview of the document image classification landscape.

2.2 Overview of Information Extraction Datasets

Obtaining a suitable dataset deemed problematic. Key information extraction datasets are rarely shared due to the private nature of the business documents. And while there are publicly datasets (described in Table 2.1), they are small, they lack vendor-level (template) annotations and most of them are not from the domain of interest (invoices, orders,...). To my knowledge, there is not a publicly available dataset of business documents with annotations for localized key information extraction that also contains template annotations[8].

[8]There are some document datasets that should contain the layout classes, however they are small [30] and none of them was possible to download [30, 51].

Table 2.1: Overview of datasets related to KI(L)E from semi-structured business documents, *ftypes* stands for the number of field types *mp* for multipage.

name	document type	docs	ftypes	source	mp	lang.	type
WildReceipt [52]	receipts	1740	25	photo	no	en	KILE
Ghega [29]	patents/datasheets	246	11/8	scan	yes	en	KILE
EPHOIE [53]	chinese forms	1494	10	scan	no	zh	KILE
CORD [54]	receipts	11000	42	photo	no	ind	KILE
DeepForm [11]	invoices, orders	1000	6	scan	yes	en	KILE
Kleister Charity [55]	financial reports	2788	8	scan	yes	en	KIE
Kleister NDA [55]	NDA documents	540	4	scan	yes	en	KIE
SROIE [22]	receipts	973	4	scan	no	en	KIE

The Table 2.1 contains an overview of the datasets available for Key Information Extraction (and Localization). Some of the mentioned datasets are further described below. Some of the datasets mentioned below are not directly usable for KIE, but they could be potentially extended by adding the missing annotations.

We highlight the issue of missing datasets and propose alternative sources of data in the positional paper which was published along this thesis [12] (see Appendix C). Rest of this section further describes datasets relevant to the business document information extraction.

SROIE (Scanned receipts OCR and key information extraction) [22] contains 1000 scanned images of receipts. The dataset comes with three types of annotations used for three different challenges: a) annotations of texts with their related positions for text localization task b) annotations of words present in the receipt for the OCR detection task and c) annotations (without positional information) of 4 field types for key information extraction.

FUNSD (Form Understanding in Noisy Scanned Documents) [56] contains a subset of 200 fully annotated documents from RVL-CDIP [57] dataset. Annotations consist of interlinked semantic entities (groups of words that belong together). Each entity is annotated with a bounding box, its textual content and links with other entities. Also, a label from "question", "answer", "header" or "other" is assigned to each entity.

DeepForm [11] benchmark by Weights&Biases consists of 20000 political advertisement forms (invoices, orders,..) from 2012, 2014 and 2020 US elections. A subset of 1000 documents was annotated with 6 semantic field types. Each document is provided as a PDF file, authors also provide OCR texts. Annotations contain the ground truth text and also the positional information usable for KILE.

CORD (A Consolidated Receipt Dataset for Post-OCR Parsing) [54] is a dataset of 11000 photos of Indonesian receipts from shops and restaurants. Each

annotation consists of a bounding box, textual contents and its label: five superclass and 42 subclass labels are used.

Kleister NDA[9] and Kleister Charity[10] [55] are datasets for end-to-end key-information extraction from long business documents. Charity dataset consists of 2788 financial reports from charity organizations. NDA dataset contains 540 non-disclosure agreements. The goal is to extract 8 (respectively 4) normalized attributes from each document. The annotations do not contain positional information of the target values.

WildReceipt [52] is a collection of 1740 pictures of English receipts with emphasis on variability of different templates. It is annotated for 25 key information categories with positional information about the information location.

Ghega dataset [29] consists of 136 patents and 110 data-sheet documents. Each dataset is further separated into classes by patent source and component type respectively. OCR outputs and deskewed page images are also provided by the authors. Annotation contains 8 attributes for the datasheets and 11 attributes for the patents. Annotations include positional information and also related blocks, that are relevant to the target value.

EPHOIE dataset [53] contains 1494 images of scanned Chinese school examination papers. The dataset combines printed and handwritten documents. 10 categories of values are annotated including their positions.

DocVQA [46] consists of 50000 natural language questions over 12767 industry documents collected from the UCSF Industry Documents Library[11]. It contains various types of documents with the majority being letter, form and report document types.

XFUND[12] [42] is a synthetic dataset that extends FUNSD [56] to other languages. It contains human-labeled forms in 7 languages, with 149 documents in the train, and 50 documents in test split for each language. The layout of the forms was taken from public documents (in the respective language) from the internet while the content was filled by human annotators with synthetic data.

[11] `https://www.industrydocuments.ucsf.edu/`

2.3 Other Related Work

We expect the reader to be acquainted with the basic concepts used in the deep learning field without explicitly repeating them here. Great learning materials to recommend are for example the famous "Deep learning with Python" [58] by Francois Chollet or "Deep Learning" [59] by Ian Goodfellow.

Following subsections briefly touch the building blocks that relate to the proposed solution in Chapter 4 such as the architecture of ResNet/UNet models or a short introduction to contrastive learning.

2.3.1 ResNet

ResNet [9] is a fully convolutional neural network architecture introduced by Microsoft Research in 2015. ResNet has proven its capabilities by winning the ILSVRC & COCO competitions in 2015. The ResNet architecture tackles the issue of wanishing gradient by combining blocks of multiple convolutional layers with a skip connection that directly connects the block's input with the block's output bypassing the block's layers.

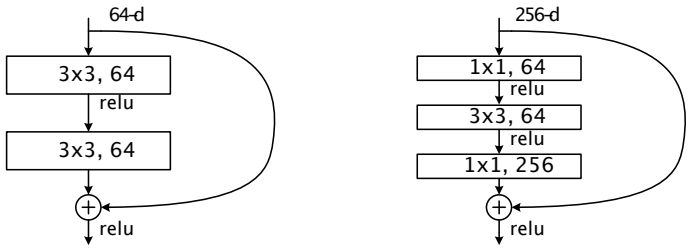


Figure 2.9: Structure of the ResNet [9] block. A block used in a shallower ResNet34 is shown on the left, ResNet50 block on the right.

The ResNet structure of the ResNet block has allowed the authors to effectively train even very deep[13] neural network architectures.

The implementation of the proposed method in Chapter 4 uses a modified ResNet50 neural network readily available pretrained on ImageNet [60] dataset in Tensorflow [61]. The Tensorflow implementation was originally designed for the image classification task, which means that the fully-convolutional network ends with a global average pooling and a fully connected layer[14]. Some of the latter parts of the network were removed in order to use it for the proposed approach.

2.3.2 UNet

UNet [10] is a fully convolutional encoder-decoder architecture originally developed for the medial image segmentation task. Encoder gradually reduces

[13] Considered deep in 2015 terms.

[14] This part of the network is often called *top* in Tensorflow.

the spatial dimensions while continuously increasing the number of channels in the subsequent layers. Decoder, symmetrical to the encoder, expands the deep feature channels back into larger spatial dimensions. Upsampled features are also combined with connections from the encoder to preserve the high-resolution image components. Thanks to the encoder-decoder architecture, UNet has a large receptive field while preserving the high-resolution features by incorporating the "skip" connections from the encoder.

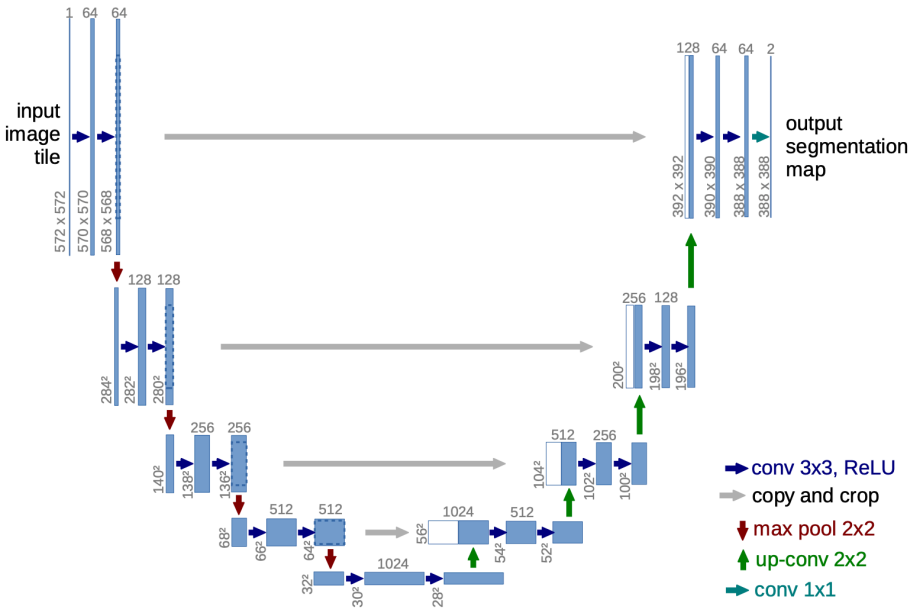


Figure 2.10: Encoder-Decoder architecture of UNet [10].

2.3.3 Contrastive Learning

Contrastive learning is a form of self-supervised representation learning which aims to project the data into a representation (embedding) space, in which objects of the same label are close, while objects from different labels are apart. SimCLR [62], the work by Chen et al., aims to learn visual representations of unlabeled images by learning a representations of augmented pairs of the same image.

By unsupervised pretraining and then finetuning the ResNet50 architecture on only 1% of the ImageNet dataset labels, authors beat AlexNet [63] trained on a fully-supervised ImageNet dataset.

The proposed approach in Chapter 4 is heavily influenced by the the paradigm presented in SimCLR. It naturally comes to mind, that a similar approach could be potentially used to represent different field instances

within the documents. And once we have the field representations, we can potentially use them for information extraction.

Chapter 3

Problem Statement

The goal is to design a system for *Key Information Localization and Extraction* (KILE 3.1) from newly received document with the help of database of previously-processed documents. The learning should be done in one-shot fashion, incrementally improving system's performance with each processed document to quickly adapt to new and changing document layouts. As noted by Hamza et al. [2]: "It is obvious that if the system has processed a similar document before, then it is a real waste of time not to take advantage of such knowledge."

The challenge of this task lies in the variability of the processed documents. Documents differ not only among different document types and and vendors, but also within the documents of the same template. This intra-class variance is visualized in Figure 3.1.

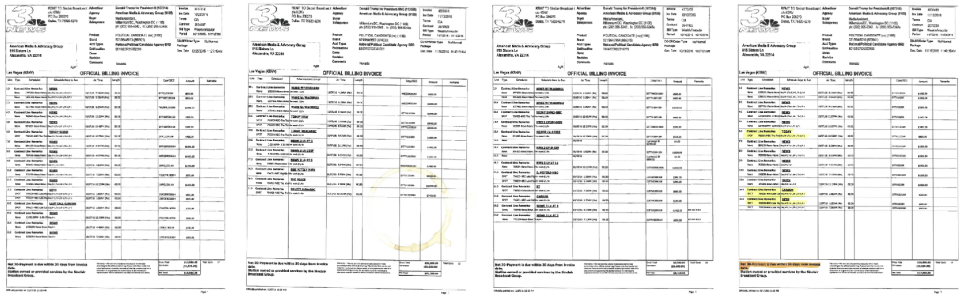


Figure 3.1: Example of invoices sharing the same document (layout) class. Notice the variability caused by imperfect scanning and other visual imperfections. Note the intra-class variability caused by tables of different length. Source: DeepForm [11] dataset, modified.

Figure 3.2 visualizes the positions of different fields on the documents that belong to the same document class. Note that some fields remain generally

fixed, while other fields change their positions across documents.

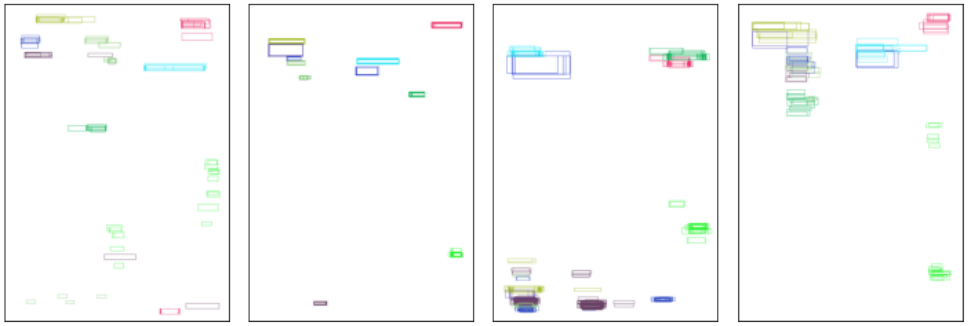


Figure 3.2: Visualization of 4 different layout classes from the training dataset. Note that the location of some field types (more or less) remain fixed, while other field types such as *amount total* move across the document (green boxes).

An idealized one-shot information extraction system should have the following attributes:

adaptation speed The system should be able to adjust quickly to a new document template. Users typically expect that the system learns from only one to two samples.

transformation invariance System should recognize and transfer knowledge between documents of the same layout regardless of the actual image representation. Documents might be colorful, black-and-white, noisy, different size, rotated, contain drawings, highlights and more.

source invariance It should not matter whether the document is a native PDF, scanned image or photographed picture.

multiple sources Ideal system should benefit from combining a knowledge of multiple related documents.

multipage documents Information extraction should be applicable even for multipage documents with varying numbers of pages.

many to many Some field types can occur multiple times both on source and target document. The system should be able to predict multiple instances of the same field type as well as to extract the data in tabular structures.

confidence score Each prediction should be accompanied with a calibrated confidence score estimating the posterior probability $P(\text{correct} \mid \text{text})$ to seamlessly combine it with an output of another model.

3.1 Problem Definition

Let $I_D \in \mathbb{R}^{\mathrm{H}\times\mathrm{W}\times 3}$ be a image of a document page D of height H and width W, and let $a_i = \{k_i, (x_0, y_0, x_1, y_1)\}$ be an annotation of a field within pixel-space of I_D. $k \in \mathcal{K}$ denotes the classification into a fixed set of field types and (x_0, y_0, x_1, y_1) denotes the bounding box of this field type instance (defined by the coordinates of its upper left and lower right corners). The full-page annotation A_D consists of a sequence of annotations $(a_1, a_2, \dots a_{N_D})$, where N_D denotes number of annotations of document page D. The problem can be classified as *Key Information Localization and Extraction* (KILE).

Let $\mathcal{DB} = \{(I_{D_1}, A_{D_1}), (I_{D_2}, A_{D_2}), \dots (I_{D_M}, A_{D_M})\}$ be a set of M already processed document images with their annotations. Given a newly received document image I_D, the task is to predict N_D fields present on the received document, i.e. $\hat{A}_D = (\hat{a}_1, \hat{a}_2, \dots, \hat{a}_{N_D})$ given the to the $\mathcal{DB}$.

In the language of computer vision, this task could be expressed as multiple-object detection. For the purpose of this work, we only limit ourselves to localization of at-most one instance of each field type within each document. While this is not exactly aligned with the overall objective of an information extraction system, it simplifies the evaluation and comparison of the proposed methods. All proposed methods in Chapter 4 were designed to be easily extended to predict multiple instances of the same field type.

For the sake of evaluation, we define the template class of a document as a function that transforms structured input data into a graphical document representation. Each document page D therefore belongs into one template class t where $t \in \mathcal{T}$ is a set of all documents that share the template class and $\mathcal{T}$ represents all template classes.

3.2 Utilized Dataset

Experimental dataset consists of 3223 annotated[15] single-page invoices. As described below, documents were manually classified into 732 template classes. Each document D belongs to exactly one template class t. Each template class $t \in \mathcal{T}$ in our dataset contains at least 2 documents. It is worth noting, that the definition of a document template is highly problematic, as

- vendors reuse and customize existing templates,
- documents are produced in different languages,
- templates evolve over time.

[15]Field-level annotations were provided by Rossum. For legal and privacy reasons, the dataset cannot be published as part of this work. We aim to publish a public dataset (without any customer data) in a future publication.

The line between what is, and what is not the same template is often hard to define. Overall, the documents which were created with the same software around the same time with the similar visual features are considered to share the template class.

To manually annotate the dataset into template classes, a custom annotation tool was developed in order to efficiently recommend pairs of the documents to the annotator. The annotation process itself is then just binary labelling of image pairs and answering whether the two documents share the template class or not. This process is described in more detail in Section 6.1. Different document layout classes are not mixed among the different dataset splits. Training dataset should not contain any documents that share their template with any of the documents in the test/validation datasets. The information about dataset splits are shown in Table 3.1.

Table 3.1: Template class sizes in each dataset split. Intervals are inclusive.

split	docs	class	2-2	3-3	4-7	8-15	16-23	24-31	32-39	40-48
$\mathcal{D}_{train}$	2222	492	492	307	202	66	20	7	2	1
$\mathcal{D}_{valid}$	494	120	120	70	44	12	0	0	0	0
$\mathcal{D}_{test}$	507	120	120	68	49	9	0	0	0	0

The dataset contains 11 field types. Selected field types rarely overlap each other and represent a meaningful set of fieldtypes to measure the performance of information extraction systems. Dataset contains both field types, that typically stay at a fixed position within the template class (*document id*), as well as fields that often shift due to elements of variable sizes (*amount total*). This is illustrated in Table 3.2.

field type	amount total	bank num	date issue	document id	phone num	recipient addr.	recipient name	sender addr.	sender dic	sender ic	sender name
avg std x	18.3	16.9	8.9	7.7	31.4	15.0	21.7	16.7	11.0	12.6	23.4
avg std y	42.3	11.1	8.1	9.5	28.3	8.8	10.1	27.8	15.0	14.4	46.5

Table 3.2: Mean standard deviation across centers of fields within dataset's clusters. Shown separately for x and y coordinates.

As seen in Table 3.3, not every document contains all field types. In cases, where the document contains more than one field of the given class, only the first occurrence of the field (when sorted first by vertical, and then horizontal position) was selected.

Table 3.3: Number of field occurrences and average field type counts over documents in different dataset splits.

	$\mathcal{D}_{train}$		$\mathcal{D}_{valid}$		$\mathcal{D}_{test}$	
field type	count	per doc.	count	per doc.	count	per doc.
amount total	2265	0.98	460	0.99	445	0.98
bank num	1296	0.56	323	0.69	291	0.64
date issue	2255	0.98	463	1.00	453	1.00
document id	2273	0.99	458	0.98	453	1.00
phone num	1837	0.80	379	0.82	369	0.81
recipient addr.	657	0.29	131	0.28	114	0.25
recipient name	2294	1.00	465	1.00	454	1.00
sender addr.	671	0.29	131	0.28	114	0.25
sender dic	1826	0.79	382	0.82	381	0.84
sender ic	1931	0.84	397	0.85	385	0.85
sender name	2283	0.99	459	0.99	453	1.00

3.3 Evaluation Metrics

The desired output of the considered key information extraction system is either text prediction, or its normalized value. This, however, makes the evaluation cumbersome for several reasons. The prediction results will be always upper-bounded by performance of the OCR model. And in case of the final information extraction system also by the normalization business rules (to normalize dates, amounts, ...). Employing OCR model just for the sake of evaluations complicates development and model checkpointing while training.

The studied task can be also formulated as "object detection based on semantic segmentation". So instead of complicating the evaluation with text extraction and normalization, we solely evaluate the position of the predicted bounding box against its gold annotation.

$$IOU = \frac{\text{Area of Intersection of two boxes}}{\text{Area of Union of two boxes}} \in [0, 1] \tag{3.1}$$

The IOU metric from Eq. (3.1) was selected for its simplicity and wide use in object detection. As described below, we threshold the IOU between gold annotation and prediction to decide whether the field was correctly localized.

It is worth noting, that while thresholded IOU generally captures whether the predicted position of the field was correct or not, it can be rather a pessimistic measure with respect to the downstream task. Our considered pipeline uses the predicted bounding box to generate a crop from the page image. This crop is then fed into an OCR system to extract the text. However, the OCR prediction on crop with text surrounded by whitespace is the same as with

a bounding box tightly aligned. Due to this fact, there can be a slight discrepancy between measured IOU and the downstream task performance.

The ultimate goal is to save the manual work of the person interacting with the information extraction system. This includes correcting the predicted field positions, adding the missing predictions, as well as removing false positives. To further simplify the evaluation, we denote a predicted bounding box as "correctly localized" if the IOU (Eq. (3.1)) is over 0.35. Selected threshold gives the system robustness against the issue mentioned earlier while still capturing whether the field was correctly localized.

We also allow the annotation/prediction to be "none", representing that the field is not on the document. For a pair of annotation a and prediction $\hat{a}$ of the same field type k three types of errors might occur: *wrong* - model predicted a bounding box that did not match the gold annotation ($IOU < 0.35$), *extra* (*false positive*) - model predicted a value (dreamed it up) on a document where should be none and *miss* (*false negative*) - model was supposed to predict a value but it did not predict anything.

3.4 Evaluation Procedure

This thesis aims to develop a system capable of utilizing knowledge from a database of already-processed documents to identify and extract information from a newly incoming (query) document. The evaluation procedure should reflect the performance on the target task as closely as possible. The results shall provide insight into the performance of the evaluated system both when no similar document is in the database as well as when we add documents from similar templates. By measuring the performance after adding similar documents, we can observe whether the model's performance actually improves with more samples of relevant documents. Or whether the model is able to generalize even without seeing a similar document.

An experimental protocol similar to the approach proposed by [4] is used. We first initialize the document database with no documents from the tested template class, and then simulate adding related documents one-by-one. The performance is evaluated for each newly added document. Selection of the new (query) document given the relevant dataset is described in Equation (3.2).

Let S be a dataset used for evaluation. For template class t with documents $(D_{(t,0)}, D_{(t,1)}, \ldots)$, start with full dataset without all documents from evaluated template class $(S \setminus \{D_{(t,0)}, D_{(t,1)}, \ldots\})$ and obtain the new (query) document $Q_{(t,i)}$ and a new dataset $S_{(t,i)}$ (used in $\mathcal{DB}$) as follows:

$$\begin{aligned} Q_{(t,0)} &= D_{(t,0)}, S_{(t,0)} = S \setminus \{D_{(t,1)}, D_{(t,2)}, \ldots\} \\ Q_{(t,1)} &= D_{(t,1)}, S_{(t,1)} = S_{(t,0)} \cup \{D_{(t,0)}\} \\ Q_{(t,2)} &= D_{(t,2)}, S_{(t,2)} = S_{(t,1)} \cup \{D_{(t,1)}\} \\ &\vdots \\ Q_{(t,|t|-1)} &= D_{(t,|t|-1)}, S_{(t,|t|-1)} = S_{(|t|-2)} \cup \{D_{(t,|t|-2)}\} \end{aligned} \tag{3.2}$$

Given the template class t, the score is measured for each query document $Q_{(t,0)}$, $Q_{(t,1)}$, $\ldots Q_{(t,|t|-1)}$. The $Q_{(t,0)}$ represents a new document and $S_{(t,0)}$ a dataset with no documents that share the layout class with $Q_{(t,0)}$. For $Q_{(t,1)}$, there will be one document from template t already present in $S_{(t,1)}$.

To obtain a single score over all field types and all template classes in the dataset split, we evaluate the scores for different integer values $n \in \{0, 1, \ldots\}$. Since the templates contain different number of documents, we denote $Q_{(t,m)}$ as a document from template t where $m = min(n, |t| - 1)$ is the last document being added into the document database which consists of annotated documents $S_{(t,m-1)}$. This is described in the Equation (3.2).

For query document $Q_{(t,i)}$, system predicts annotations $(\hat{a}_{(q,1)}, \hat{a}_{(q,2)}, \ldots)$. We define $mACC@n$ (mean micro accuracy at n) as the average score (over individual field predictions) with at most n documents from the template class in the database for each predicted document.

$$mACC@n = \frac{\sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} HIT(a_{(q,k)}, \hat{a}_{(q,k)}) \mid q = D_{(t,min(|t|,n))}}{|\mathcal{T}| * |\mathcal{K}|} \tag{3.3}$$

The metric proposed at (3.3) is designed to provide a single score across all field predictions given n documents of a template class for a newly added document present in the evaluation database. To accommodate for true negatives (where the model correctly predicted none[16]), thresholded IOU (HIT) is defined as:

$$HIT(a, \hat{a}) = \begin{cases} 1 & \textit{if } IOU(a, \hat{a}) > 0.35 \\ 1 & \textit{if } UNION(a, \hat{a}) = 0, \\ 0 & \textit{otherwise.} \end{cases} \tag{3.4}$$

Note that the last option also captures the cases when one of the annotation/prediction is *none*. To evaluate performance of the system for a single field type, we call the metric presented in (3.3) simply as $ACC@n$ since there is no micro-averaging over different field types.

[16] If both annotation and prediction are empty $UNION(a, \hat{a}) = 0$.

Chapter 4

Proposed Method

Many of the approaches in one-shot information extraction are inspired by the traditional layout-based extraction systems where the incoming document is first classified, and the selected extraction template is then used for the information extraction.

Yet, we can make a strong argument why these systems can never work perfectly. *The notion of similarity between two documents is inherently defined by the utilized extraction method.* For optimal performance, the retrieval and information transfer must be interconnected.

Let's show this on an example. Database contains two extracted documents $D_{(A,1)}$ and $D_{(B,1)}$ from different templates A and B. We have an incoming (target) document $D_{(B,2)}$ of the same template as $D_{(B,1)}$ that we want to extract. The document $D_{(B,2)}$ perfectly matches the structure of $D_{(B,1)}$, but it was poorly scanned and the document is rotated within the scanned image. Document $D_{(A,1)}$ is, on the other hand, correctly aligned and its visual features loosely follow $D_{(B,2)}$.

$D_{(B,2)}$ is used as a query for document retrieval using simple visual similarity. Most visually similar document $D_{(A,1)}$ is retrieved. The positions of fields are copied from $D_{(A,1)}$ to $D_{(B,2)}$ using a naive copy-paste annotation transfer. The performance of this prediction will likely be poor, since different fieldtypes may be present in the copy-pasted locations.

However, if the extraction algorithm was robust against document rotations, it could extract the information nearly perfectly given a document from the same template was correctly retrieved. But since the retrieval method relied only on visual features, the best source document $D_{(B,1)}$ for this extraction method could have never been retrieved.

This simple argument shows my motivation to define (and train) both retrieval, and transfer steps in conjunction.

4.1 Template Matching Baseline

This approach consists of two steps: a) *document retrieval* – given a query document, retrieve the most relevant document from the database based on the similarity measures described below. And b) *information transfer* – copy field locations from the retrieved document and use them as the annotation prediction for the query document.

4.1.1 Document Retrieval

Several document retrieval methods were implemented. All proposed document representations use negative L2 distance as similarity metric. Proposed document retrieval methods are:

visual Document image is downsampled into shape $(31 \times 43 \times 3)$. Individual pixel values are flattened resulting into a single 3999 dimensional "float32" page embedding. This approach is similar to the approach proposed in [29].

dejavu Features are obtained from early (3rd) convolutional layer of a proprietary document segmentation model. Features are pooled and flattened into 4640 dimensional "float32" page embedding. The reasoning behind this embedding is that the representation should be more robust than pure visual similarity.

oracle From the correct layout cluster, retrieve the most similar (based on *dejavu* representation) document. This simulates an upper bound for document layout retrieval based on visual similarity.

4.1.2 Information Transfer

For an incoming query (target) document, we retrieve the most similar (source) document using one of the methods described above. We reuse the annotation information from the source document and apply it on the target document using an information transfer method.

copypaste A simple approach to the field transfer between a source and target document. Use all field annotations (which consist of field types and their respective bounding boxes) from the source document and copy them onto the target document.

4.2 Field-Level Representation Learning

The previously described approach — albeit its simplicity — represents a very strong baseline. This might not be obvious at first glance, but in reality, the processed documents in the dataset are mostly digital-born, and thus precisely aligned. The visual similarity-based retrieval therefore works very well as the documents share the overwhelming majority of their visual features.

Copypaste transfer has a very high accuracy for fields with static positions and constant sizes such as *document id*. However, it falls short on:

- documents with variable-sized elements,
- documents where the mapping of fields is not one-to-one,
- artifacts and shifts introduced by scanning or pre-processing,
- representation and retrieval of multi-paged documents.

The method proposed below is inspired by the recent adoption of self-supervised methods such as the approach described by Chen et al. in SimCLR [62]. The main idea is to train a CNN backbone f_{CNN} that projects the input document image $I_D \in \mathcal{R}^{\text{H}\times\text{W}\times 3}$ into a representation $R_D \in \mathcal{R}^{\mathcal{H}\times\mathcal{W}\times\mathcal{S}}$ where $\mathcal{S}$ is the dimensionality of the representation-space and $\mathcal{H} \leq \text{H}$, $\mathcal{W} \leq \text{W}$ are its spatial dimensions. The spatial size of the representation space is smaller than the input image to simplify the experiments, but in general, it could have the same width and height as I_D.

The reasoning behind training the f_{CNN} projection is that we wish the network to learn a higher-level representation of the pixels that make up each field. Similar to a human choosing a similar field when presented with an example, the network should be able to encode the field's representation by recognizing the style of the input pixels and their neighborhood. The trained representation can then be used to localize same field on a different document.

Let f_{cut} be a function, that combines the representation R_D with annotation of a single field $a = (k, b)$ and reduces R_D into a subset of superpixels that belong to a. The bounding box $b = (x_0, y_0, x_1, y_1)$ of field annotation y is first linearly rescaled into feature-space coordinates $b^r = (x_0^r, y_0^r, x_1^r, y_1^r)$. Rescaled coordinate values are rounded down for x_0^r, y_0^r and up for x_1^r, y_1^r to capture the whole area of the annotation within the document. Given the $h^r = y_1^r - y_0^r$ and $w^r = x_1^r - x_0^r$ are the height and width of b^r, it is used to select a patch of features c from R_D of shape $w^r \times h^r \times S$.

The last function f_{pool}, takes in a subset of superpixel representations from R_D, and applies average pooling across the spatial dimensions resulting into a 1D vector of real numbers $q \in \mathcal{R}^{\mathcal{S}}$.

Given an input document image I_D, and annotation a of a single field, we can obtain the field representation $q_{(D,a)}$ as

$$q_{(D,a)} = f_{\text{pool}}(c_{(D,a)}) \qquad c_{(D,a)} = f_{\text{cut}}(R_D, a) \qquad R_D = f_{\text{CNN}}(I_D) \tag{4.1}$$

A representation is created for each annotation of $a \in A_D$ as visualized in Figure 4.1. Extra *unk* ($|A_D| + 1$)th class covers all document pixels without any annotation. We define feature matrix $F_D \in \mathbb{R}^{(|A_D|+1)\times\mathcal{S}}$ of document D as a matrix containing representations $q_{(D,a_k)}$ of fields within the document. We denote a_k as an annotation of field type k. A field representation $q_{(D,a_k)}$ for the annotation a_k can therefore be calculated as

$$q_{(D,a_k)} = f_{\text{pool}}\Big(f_{\text{cut}}\big(f_{\text{CNN}}(I_D), a_k\big)\Big) \tag{4.2}$$

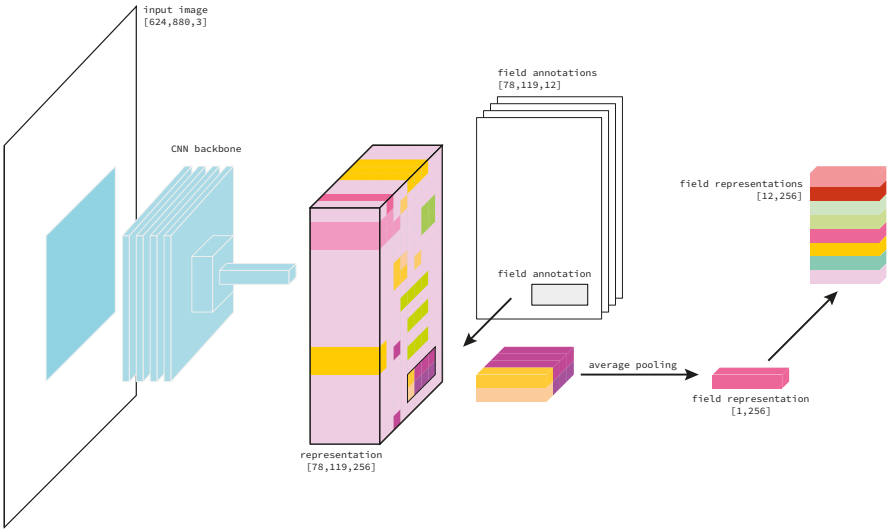


Figure 4.1: Feature extraction using the proposed approach. Field representation is created for each annotation on the input document. Backbone f_{CNN} transforms the input image I_D into a representation space R_D where $S = 256$ and where the document contained all 11 fieldtypes. Field annotations are used to create field representations by averaging their respective superpixel cuts across spatial dimensions as described earlier.

4.2.1 Document Retrieval

Let R_Q be a predicted representation of an incoming query document D_Q, and F_S a matrix with field representations of already-processed document D_S. $\mathcal{K}$ denotes the field types contained within F_S and r_{ij} a superpixel on coordinates (i, j) within R_Q. We define the distance m between matrices F_S and R_Q as:

$$m(R_Q, F_S) = \frac{1}{|\mathcal{K}|} \sum_{q_k \in F_S} \underset{r_{ij} \in R_Q}{\operatorname{argmin}} ||q_k - r_{ij}||_2 \tag{4.3}$$

Database $\mathcal{DB}$ stores all previously annotated documents and their predicted field representations. When a new document image I_Q of a document

D_Q is received, it is first projected into representation space $R_Q = f_{\text{CNN}}(I_Q)$. A source document D_S (along with its annotations A_S and field representations F_S) is retrieved from the database $\mathcal{DB}$ using negative similarity metric m defined in Equation (4.3). The field representations F_S are used for the information extraction on D_Q.

4.2.2 Information Transfer

To obtain a prediction for an incoming query document D_Q given annotated source document D_S, we reuse the field representation matrix F_S retrieved from the database. Now, given a representation $q_{(S,a_k)} \in F_S$ of fieldtype k, we calculate distance matrix P_k, where for each superpixel r_{ij}: $P_k^{ij} = ||q - r_{ij}||_2$. We predict an annotation $\hat{a} = (k, \hat{b})$ from distance matrix P_k in three steps:

1. Binarize the P_k on threshold θ (i.e. 0.5).
2. Select the largest connected component.
3. Construct a bounding box from the connected component.

This process is further visualized and described in Figure 4.2.

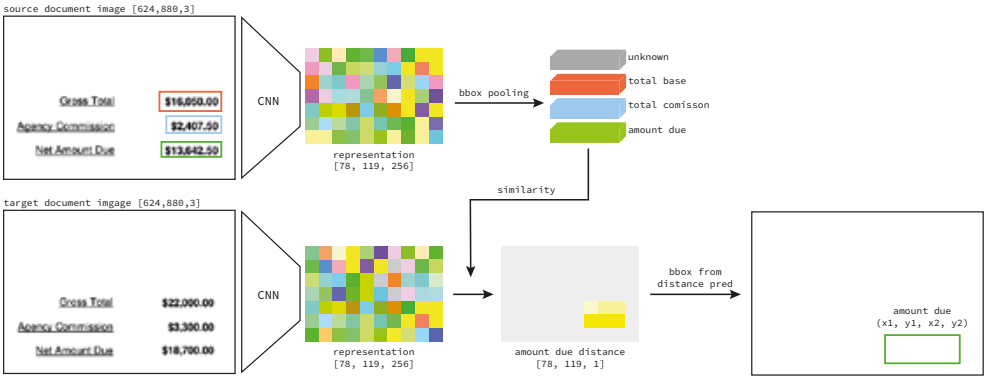


Figure 4.2: Prediction of a field (*date due*) within the query document D_Q (bottom) given a source document D_S (top). The upper part (field representations) is generally precomputed and retrieved from the $\mathcal{DB}$ when the query document arrives.

4.2.3 Model Training

For training, a generator that yields pairs of documents of the same template class was implemented. For each training batch, we select $n/2$ distinct template classes where n is the batch size. From each template class, we randomly select two different documents. The batch therefore contains n documents in

$n/2$ pairs from different template classes. One training epoch consists of going through all layout classes.

In the forward pass, documents D_1 and D_2 of the same template are projected using backbone CNN network f_{CNN} into their representations R_1 and R_2. To simplify the computation of cosine similarity, all vectors within R_1 and R_2 are L2-normalized along the depth dimension ($\mathcal{S}$).

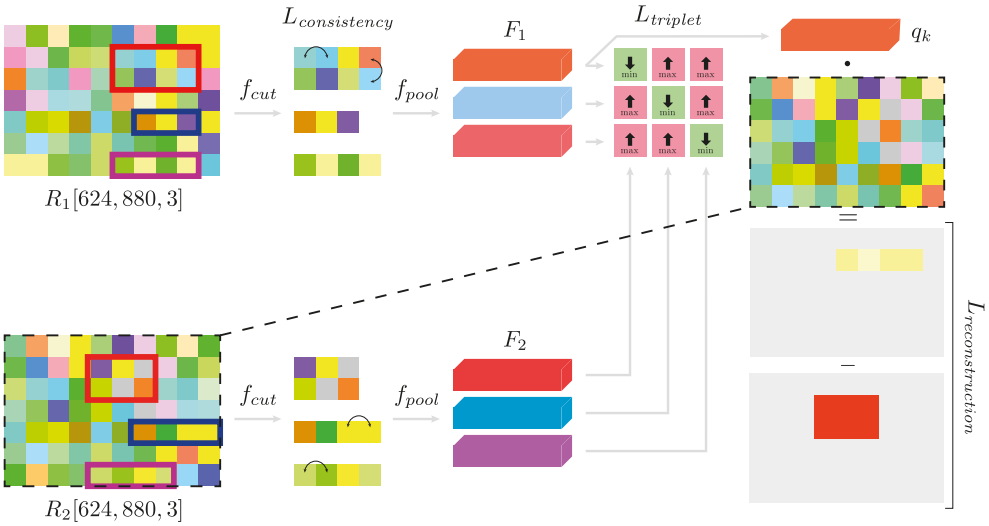


Figure 4.3: Overview of the training and involved losses over a single pair of images. Arrows in L_{triplet} show which distances are minimized/maximized.

Three distinct loss functions (as visualized in Figure 4.3) are used in the training:

Triplet Loss We calculate F_1 and F_2 from R_1 and R_2 as described earlier. Given a pair of representations $q_{(D_1,a_{k_i})} \in F_1$ of field type k_i and $q_{(D_2,a_{k_j})} \in F_2$ of field type k_j, we wish to to minimize the distance between representations if $k_i = k_j$, and maximize it otherwise.

Triplet loss, as proposed in [64], aims to minimize the distance between a positive sample $q_p = q_{(D_1,a_{k_i})}$ and anchor $q_a = q_{(D_2,a_{k_i})}$ while simultaneously pushing a negative sample $q_n = q_{(D_1,a_{k_j})}$ away at least by margin m. In all experiments, margin m was set to 1.

$$\mathcal{L}_{\text{triplet}} = \sum_{q_a,q_p,q_n} [m + ||q_a - q_p||_2 - ||q_a - q_n||_2]_+ \tag{4.4}$$

The implemented model uses *Triplet Hard Loss* [65] which selects the hardest positive and negative samples within the batch when forming the triplets. The anchor q_a and positive q_p are selected as pairs of the same field class, but from two different documents. This is what motivates the network to keep the field representations similar across documents of the same template.

The triplet loss is calculated for each pair of documents within the training batch. The final loss is calculated as average over $n/2$ pairs within the batch.

Multiple experiments were also done with a contrastive loss similar to the loss presented by Chen et al. in SimCLR [62]. In the end, the exact implementation of the used triplet loss did not significantly influence the results, hence the simpler-to-use[17] *Triplet Hard Loss* was selected.

Consistency loss Due to the nature of f_{pool}, when $\mathcal{L}_{\text{triplet}}$ was used alone, the representations within the super-pixel crops were not forced to be consistent. This is because $\mathcal{L}_{\text{triplet}}$ averages the annotation's super-pixel representations, while what we also want is to have consistency within all super-pixels that compose the each field's representation (before averaging). We want this since the field representation is compared with each of the superpixels of the representation space during prediction. And consistency heavily influences the results.

For a document D annotations $A_D = (a_1, a_2, \ldots)$, and their respective crops denoted as $C = (c_1, c_2, \ldots c_{|A_D|})$ obtained using f_{crop} from R_D, $r \in c_i$ marks a single superpixel within the crop c_i. The consistency loss for one document is calculated as:

$$\mathcal{L}_{\text{consistency}} = \frac{1}{|D_A|} \sum_{c_i \in C} \frac{\sum_{r_j \in c_i} \sum_{r_k \in c_i} ||r_j - r_k||_2}{|c_i|^2} \tag{4.5}$$

In other words, we minimize the pairwise distance between all pairs of the super-pixels within each annotation crop of R_D. Since this term grows quadratically with the size of each crop, extracted regions are randomly subsampled to preserve the GPU memory if they contain more than 100 elements.

The consistency loss is calculated for each document and then averaged over all documents within the batch to obtain single loss value.

Reconstruction loss is a supervised loss that directly optimizes the downstream prediction task where we want to predict a position field on newly received document using another document's field representation. Given field representation q_k from a source document and a representation space R of the target document, we predict the distance matrix $\hat{P}$ as previously described in Section 4.2.2. This is done for each field representation q_k from the source document. The predicted distance matrix $\hat{p}_k$ is then compared with a binary mask b_k that was obtained from the target document's gold annotation a_k. We calculate the reconstruction loss $\mathbb{RL}$ as:

$$\mathbb{RL}(D_A, D_B) = \sum_{b_k \in D_A, q_k \in F_B} \mathcal{D}(b_k, ||R_A - q_k||_2) \tag{4.6}$$

[17] And readily available in TensorFlow Addons.

For $\mathcal{D}$, focal loss [66] was used instead of binary cross entropy since the predictions are generally sparse. The reconstruction loss for each pair of documents in a training batch is calculated and averaged as:

$$\mathcal{L}_{\text{reconstruct}} = \frac{\mathbb{RL}(D_1, D_1) + \mathbb{RL}(D_1, D_2) + \mathbb{RL}(D_2, D_1) + \mathbb{RL}(D_2, D_2)}{4} \tag{4.7}$$

A simplified overview of all three losses involved on a single training pair of documents is visualized in Figure 4.3.

The total loss $\mathcal{L}$ is calculated for each pair of the images within batch. If there are multiple pairs within the training batch, this loss is averaged. Loss $\mathcal{L}$ for a pair of training documents can be represented as:

$$\mathcal{L} = \mathcal{L}_{\text{triplet}} + \mathcal{L}_{\text{consistency}} + \mathcal{L}_{\text{reconstruct}} \tag{4.8}$$

The best model is selected from each training based on the validation transfer accuracy over pairs of documents from the same cluster (basically $mACC@1$ with the *oracle* retrieval method) at the end of each batch.

4.2.4 Backbone Model

The convolutional architecture used as f_{CNN} is based on ResNet50 [9]. The model was pretrained on the Imagenet [60] dataset and modified for the proposed segmentation task. Last 32 pooling layers were removed (including the classification head) and upsampling step with a residual connection was added as inspired by the U-Net [10] encoder-decoder architecture. The upsampling was followed by several 3×3 and 1×1 convolutions to obtain the desired feature space of desired depth $\mathcal{S}$. The modified model consists of $10,789,120$ trainable parameters. It is worth noting, that the predicted segmentation mask is not upscaled to the full resolution of I_D, but rather to a smaller super-pixel space R_D.

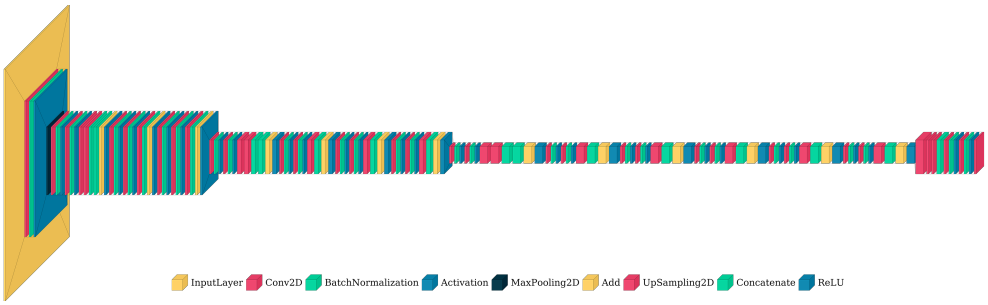


Figure 4.4: Architecture of the proposed model based on ResNet50.

Adam optimizer [67] with default parameters was used for all trainings. Using different optimizers (namely RMSProp, Yogi [68], AdaGrad [69]) did not show any improvement in the training performance. Due to GPU limitations, batch size of 6 was used (6 images from 3 pairs).

Chapter 5

Experiments

All experiments within this chapter were done on models trained on the training dataset and selected based on their validation performance. Validation dataset was also used to obtain hyperparameters such as the threshold level. All presented results were performed using a test split to obtain an unbiased estimate of the performance.

5.1 Baselines

Table 5.1: Test micro average accuracy (mACC@n) of copypaste transfer on documents retrieved using *dejavu*, *visual* and *oracle* document representations.

mACC@n	0	1	2	3	4	8	16
visual	0.085	0.709	0.799	0.803	0.806	0.808	0.807
dejavu	0.103	0.723	0.796	0.809	0.811	0.814	0.813
oracle	-	0.729	0.797	0.809	0.810	0.814	0.812

Table 5.1 compares the *visual* and *dejavu* retrieval methods (defined in 4.1.1). Oracle document retrieval confirms, that both previously mentioned methods are powerful baselines for document retrieval. The results also verify the quality of the document class annotations – *dejavu* retrieval paired with *copypaste* transfer with no relevant documents in the database shows the lack of class-independent generalization of this approach.

The micro averaged accuracy presented in Table 5.1 does not illustrate different performance over fields of different types. This is shown in Table 5.2. The performance differs greatly among fields that typically remain at fixed locations (for example *document id*) and fields (such as *amount total*) that are greatly affected by the shifts caused by variable-sized elements on the invoices.

The positional field variance of different field types across within the clusters was previously illustrated in Table 3.2.

ACC@n	0	1	2	3	4	8	16
amount total	0.051	0.461	0.632	0.615	0.635	0.652	0.644
bank num	0.096	0.743	0.786	0.789	0.810	0.826	0.813
date issue	0.091	0.775	0.850	0.875	0.875	0.875	0.875
document id	0.116	0.791	0.858	0.875	0.858	0.850	0.850
phone num	0.088	0.628	0.744	0.736	0.773	0.773	0.773
recipient addr.	0.190	0.718	0.793	0.812	0.727	0.781	0.750
recipient name	0.166	0.791	0.825	0.833	0.866	0.858	0.858
sender addr.	0.238	0.750	0.827	0.875	0.787	0.843	0.843
sender dic	0.072	0.783	0.808	0.848	0.828	0.808	0.828
sender ic	0.064	0.800	0.860	0.851	0.860	0.860	0.860
sender name	0.125	0.739	0.789	0.831	0.823	0.831	0.823

Table 5.2: Performance of the *dejavu* retrieval with the *copypaste* transfer.

5.2 Document-level Similarity

A model to extract document representation was trained using the protocol described in 4.2.3. Given a predicted representation of the incoming query document and database containing (precomputed) field representations of other already-processed documents, we retrieve the closest source document as described in 4.2.1 and transfer the information using the approach described in 4.2.2.

The experiments were first performed on document representations obtained from activations from last layer of a proprietary segmentation model. The results are then improved by training a dedicated feature extraction model using the approach described in 4.2.3.

5.2.1 Reused Document Representations for Document-Level Retrieval

To try the idea of the proposed approach as a proof of concept, a multimodal semantic segmentation model pretrained on invoices (called *segnet*[18] in the further text) was used to obtain the representation for each document. The model was not trained on the documents from the test set used in this thesis, but it is very likely that the the model's training set contains documents of the same layout. The training dataset was also an order of magnitude larger

[18]No resemblance with the known SegNet [70] segmentation architecture.

(10,000 documents) as compared with dataset used to train the model below. This makes the performance not directly comparable.

The representations were obtained from the model by removing its last classification layer (which directly predicts the field segmentation masks on the target document). Only differences from the representations as described in the proposed approach is the different size of the representation space (206× 292) and also its dimensionality of 96 channels.

The last difference from the previously described approach in Section 4.2.3 is a different distance metric. Euclidean distance was used instead of the cosine similarity since it led to better results[19]. The results for different fields are described in Table 5.3.

ACC@n	0	1	2	3	4	8	16
amount total	0.458	0.667	0.644	0.701	0.669	0.695	0.712
	+0.40	+0.21	+0.01	+0.09	+0.03	+0.04	+0.07
bank num	0.457	0.532	0.513	0.506	0.487	0.506	0.519
	+0.36	-0.21	-0.27	-0.28	-0.32	-0.32	-0.29
date issue	0.625	0.717	0.758	0.783	0.758	0.783	0.775
	+0.53	-0.05	-0.09	-0.09	-0.12	-0.09	-0.10
document id	0.667	0.842	0.808	0.833	0.833	0.833	0.825
	+0.55	+0.05	-0.05	-0.04	-0.03	-0.02	-0.03
phone num	0.345	0.446	0.505	0.470	0.470	0.500	0.500
	+0.26	-0.18	-0.24	-0.27	-0.30	-0.27	-0.27
recipient addr.	0.139	0.514	0.533	0.630	0.562	0.567	0.586
	-0.05	-0.20	-0.26	-0.18	-0.17	-0.22	-0.16
recipient name	0.708	0.733	0.758	0.733	0.692	0.708	0.708
	+0.54	-0.06	-0.08	-0.10	-0.18	-0.15	-0.15
sender addr.	0.139	0.543	0.567	0.630	0.562	0.567	0.586
	-0.01	-0.21	-0.26	-0.25	-0.23	-0.28	-0.26
sender dic	0.568	0.720	0.690	0.690	0.670	0.680	0.670
	+0.50	-0.06	-0.12	-0.16	-0.16	-0.13	-0.16
sender ic	0.667	0.723	0.760	0.740	0.710	0.760	0.740
	+0.60	-0.08	-0.10	-0.11	-0.15	-0.10	-0.12
sender name	0.517	0.613	0.630	0.613	0.613	0.622	0.613
	+0.39	-0.12	-0.16	-0.22	-0.21	-0.21	-0.21

Table 5.3: Scores for representations obtained from a proprietary segmentation model. The deltas represent the difference over the *dejavu* baseline with a copypaste annotation transfer. Positive numbers represent an advantage of using the segmentation model instead of the template matching baseline.

[19]The proprietary model used ReLU activations and was not optimized for cosine similarity of the extracted features.

It is immediately obvious, that even at $n = 0$, the model is able to generalize. Even without retrieving a document from the same template class, the representations of different field classes are similar. This is not surprising, since the model was trained for this task. The poor performance on the fields such as *sender address* is worth noting since it also affects the approach evaluated below.

The fact that the model improves when $\mathcal{DB}$ contains the documents of the same template is interesting. It shows that the representations differ across different documents even though the features were extracted from a layer close to the final classification head and even though the model was trained for generalization.

5.2.2 Trained Document Representations for Document-Level Retrieval

A modified ResNet50 model was trained using the approach presented in the Section 4.2. The model was trained for 36 epochs[20], the best model was selected based on validation score calculated at the end of each epoch.

The results in Table 5.4 show that the trained model was able to outperform the *dejavu* baseline on the *amount total* field, which represents a field that changes a lot across the documents. The comparison also highlights some of the issues of the model, such as the unsatisfactory performance on the *sender address* and *recipient address* fields.

While the reason for the poor performance of the *recipient address* is not clear, it is probably caused by the unbalanced nature of the training dataset. Only 30% of the documents contain *recipient address* annotation. At $n = 3$ all of the errors were a miss (model predicted nothing). The poor performance on this field type can also be caused by the nature of the *recipient address* which is typically a multi-line text. And since the f_{pool} averages the representations inside the annotation crop, the crop will likely also contain background. The averaged field representation can therefore be very similar to the background.

Example of document extraction with above-the-average results is shown in Figure 5.1. The figure visualizes all of the prediction steps that are done to obtain the final bounding box predictions.

[20]Each epoch consists of a single pass over all training dataset layout classes.

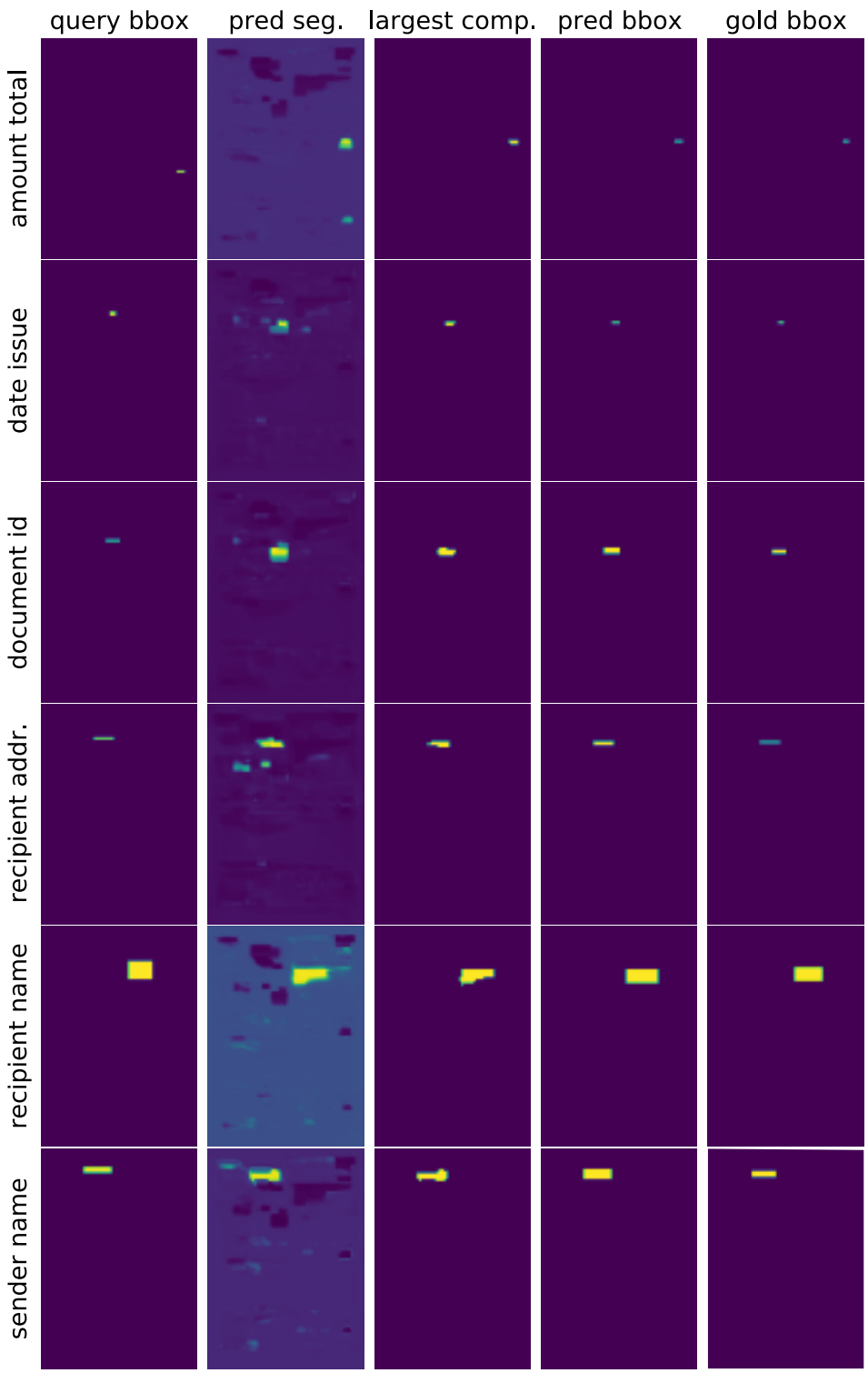


Figure 5.1: Prediction using trained document-level transfer.

ACC@n	0	1	2	3	4	8	16
amount total	0.193	0.709	0.797	0.778	0.788	0.754	0.754
	+0.14	+0.24	+0.16	+0.16	+0.15	+0.10	+0.11
bank num	0.482	0.805	0.773	0.789	0.800	0.800	0.813
	+0.39	+0.06	-0.01	+0.00	-0.01	-0.03	+0.00
date issue	0.151	0.650	0.683	0.708	0.683	0.675	0.683
	+0.06	-0.13	-0.17	-0.16	-0.19	-0.20	-0.19
document id	0.395	0.742	0.750	0.792	0.817	0.808	0.808
	+0.28	-0.05	-0.11	-0.08	-0.04	-0.04	-0.04
phone num	0.194	0.470	0.541	0.495	0.510	0.490	0.500
	+0.10	-0.15	-0.20	-0.24	-0.26	-0.28	-0.27
recipient addr.	0.677	0.375	0.333	0.333	0.345	0.259	0.286
	+0.48	-0.34	-0.46	-0.48	-0.38	-0.52	-0.46
recipient name	0.525	0.783	0.792	0.817	0.825	0.817	0.817
	+0.35	-0.01	-0.03	-0.02	-0.04	-0.04	-0.04
sender addr.	0.839	0.312	0.185	0.200	0.276	0.222	0.250
	+0.60	-0.44	-0.64	-0.68	-0.51	-0.62	-0.59
sender dic	0.351	0.505	0.510	0.505	0.515	0.485	0.485
	+0.28	-0.28	-0.29	-0.34	-0.31	-0.32	-0.34
sender ic	0.527	0.775	0.733	0.770	0.790	0.780	0.780
	+0.46	-0.03	-0.13	-0.08	-0.07	-0.08	-0.08
sender name	0.500	0.697	0.714	0.731	0.731	0.731	0.723
	+0.38	-0.04	-0.07	-0.10	-0.09	-0.10	-0.10

Table 5.4: Test score for trained ResNet50 projections. Training for 36 epochs. The deltas show the difference over the *dejavu* retrieval with the *copypaste* transfer. Positive numbers represent improvement.

5.3 Trained Document Representations for Superpixel-Level Retrieval

This alternative approach extends the document-level retrieval proposed in 4.2.1. $\mathcal{DB}$ contains all field representations from all previously extracted documents without storing the relationship to the source document[21]. When a new document Q is received, it is projected into R_Q. Each $r_{ij} \in R_Q$ is individually used as a query. h most similar field representations are retrieved for each super-pixel in R_Q.

The classes of the returned representations are used to classify each of the super-pixels in R_Q. Each of the h retrieved representations is weighted by its distance from r_{ij}. A distance matrix (of the same shape as the matrix used in the document-level approach) is constructed by a voting scheme that takes in mind the field types of the retrieved representations as well as their distances to the pixels in R_D. The steps to obtain the bounding box from the distance matrix are the same as described previously.

ACC@n	0	1	2	3	4	8	16
amount total	0.192	0.158	0.192	0.200	0.208	0.208	0.208
bank num	0.342	0.358	0.425	0.442	0.433	0.450	0.458
date issue	0.312	0.333	0.525	0.533	0.567	0.550	0.542
document id	0.508	0.508	0.675	0.708	0.733	0.717	0.717
phone num	0.000	0.000	0.008	0.008	0.017	0.008	0.008
recipient addr.	0.183	0.183	0.183	0.183	0.183	0.167	0.167
recipient name	0.625	0.600	0.583	0.625	0.625	0.633	0.633
sender addr.	0.192	0.158	0.175	0.175	0.183	0.175	0.175
sender dic	0.275	0.275	0.350	0.375	0.358	0.367	0.383
sender ic	0.400	0.425	0.542	0.542	0.575	0.575	0.575
sender name	0.542	0.575	0.575	0.575	0.608	0.600	0.608

Table 5.5: Extraction results using the superpixel-level retrieval.

Super-pixel level retrieval meets most of the requirements of an ideal information extraction as described in 3. It's potentially biggest strength over the document-level approach defined previously is the fact, that it allows to extract all field types on the query document without the need to retrieve a source document (which might lack the field representations desired to be extracted from the target document).

But as shown in Table 5.5, the results are far from the baseline performance. Retrieving more representations and using the proposed voting scheme did not improve the performance. Example of the distance matrix obtained from the mentioned approach is shown in Figure 5.2.

[21] Imagine list of tuples composed of (*representation vector*, *field type*)

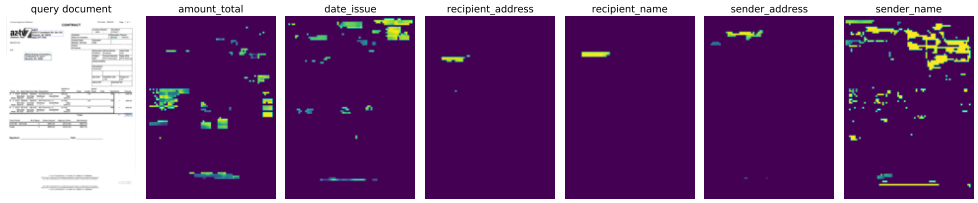


Figure 5.2: Distance matrix predictions using superpixel level retrieval.

The poor performance can be explained by the fact, that since each pixel is classified independently, the distance prediction is relatively noisy which hinders the bounding box detection. The second (and probably more severe) issue happens when user annotates the documents inconsistently, or even marks whitespace as a target field. The whitespace representation will be stored into the database under the designed field type and will corrupt all future extractions.

5.4 Other Experiments

The following subsections explain other selected experiments that could be interesting to the reader. The subsection 5.4.1 shows that all three proposed losses improve the training performance. Subsection 5.4.2 shows that transfer learning is applicable even across the tasks of image classification and business document segmentation. The subsection 5.4.3 describes an experiment that aimed to improve the performance by including multimodal inputs.

5.4.1 Loss Ablation Study

The results with different combinations of training losses are shown in Table 5.6. It describes the results of a model (best validation result) trained for 60 epochs.

$\mathcal{L}_{\text{reconstruct}}$	$\mathcal{L}_{\text{consistency}}$	$\mathcal{L}_{\text{triplet}}$	*mACC@3*
✓	×	✓	0.0403
×	×	✓	0.0852
✓	×	×	0.1212
×	×	✓	0.1277
✓	✓	×	0.3717
✓	✓	✓	0.5114

Table 5.6: Test *mACC@3* scores for ResNet50 inspired architecture trained for 60 epochs. Combination of all three losses provides the highest score.

5.4.2 Transfer Learning

The utilized model architecture was based on the ResNet50 [9] with weights pretrained on Imagenet [60] dataset. The newly added layers (later upscaling parts) were initialized randomly. Figure 5.3 shows the validation score when training a randomly initialized model versus model based on Imagenet weights.

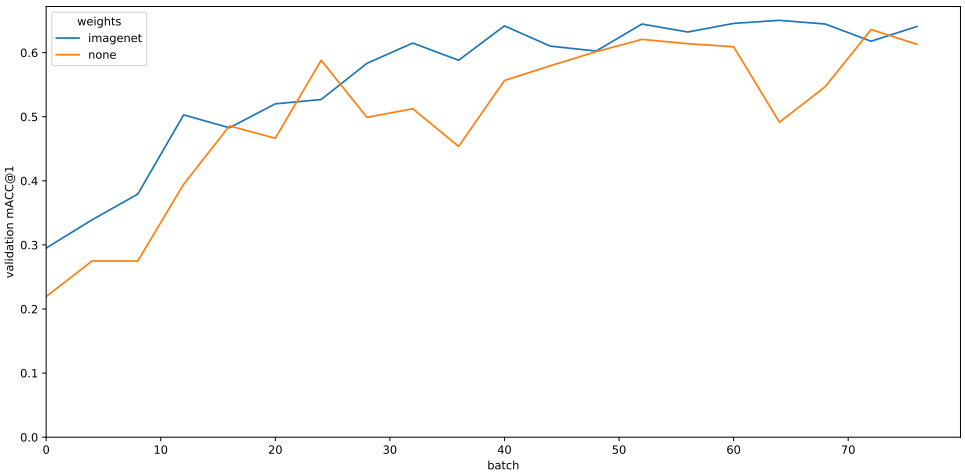


Figure 5.3: Comparison between a model with weights pretrained on Imagenet and a model with weights randomly reinitialized.

This is rather an interesting observation since the ImageNet domain is very different from the domain of business documents. Yet, the transfer learning with finetuning improves the overall performance.

5.4.3 Multimodal Inputs

Current trend in Document Understanding follows the realization, that IE models can often benefit from combining different modalities. This includes text combined with its positions to incorporate the layout information, or even with its page image to provide further visual clues for the model. In an experiment inspired by the approach presented in [33], multimodal input features were used. For each document, all page texts were first extracted either directly from PDF or by OCR if missing.

Instead of presenting the f_{CNN} solely with the page image $I_D \in \mathcal{R}^{W \times H \times 3}$, the input was extended by adding 48 channels containing one hot encoded characters (deaccented alphabet letters, numeric characters and some special characters). The network was presented (and trained) with a multimodal matrix M_D of $\mathcal{R}^{W \times H \times 51}$. Since the number of weights has changed for the first layer, it was randomly initialized as compared to the standard image-only implementation.

Table 5.7: Document-level transfer model trained on multimodal inputs.

mACC@n	0	1	2	3	4	8	16
multimodal	0.199	0.372	0.411	0.424	0.420	0.422	0.423

Interestingly enough, as shown in Table 5.7, the presented approach did not improve the performance at all. It is not entirely clear why since the original belief was that the information about individual letters and their positions would help the network. It is possible that the proposed architecture is simply not suitable for the changed inputs, but it is still surprising.

Chapter 6

Implementation Details

The model training and the annotation tool backend were implemented in Python. All applications, including training, are built into a Docker [71] container to simplify deployment and training across different machines. All models were trained on a single GTX 1080Ti GPU using the TensorFlow [61] deeep learning platform and Keras [72] API.

To enable efficient similarity search over the representation vectors, Faiss [73], a library by Facebook AI, was used.

TensorBoard[22] was used to monitor the training losses as well as to visualize the validation results during training.

To effectively load the samples into the GPU memory, a multiprocess dataset generator was implemented and used. Since most of the training losses were implemented from scratch, the models were trained using the `tf.GradientTape` Tensorflow automatic differentiation as opposed to the standard Keras API.

6.1 Dataset Annotation

Due to the lack of annotated datasets for document information extraction, creating a suitable dataset was necessary. The dataset provided by Rossum consists of documents with the annotated field locations used for the KILE task. However, the information about the template classes was not included. In order to interactively cluster (and to speed up annotation) documents into groups based on the template layout, a custom annotation tool was developed.

In order to limit the number of shown pairs, the annotation tool utilized several attributes to recommend similar documents for annotation. This comes from a observation that documents with same *sender ic*, *sender dic* or same *sender address* are usually issued by the same vendor and might there-

[22] `https://www.tensorflow.org/tensorboard`

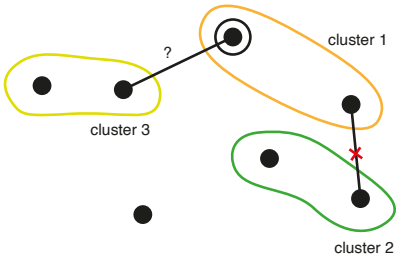
fore share the document layout. These attributes were combined with a visual similarity of the documents to generate candidates for the layout annotation.

Using a technique inspired by agglomerative clustering, a bottom up approach was used to speed up the annotation. Each document in the dataset is first assigned into a single cluster. User is asked to mark the pair of the presented documents as described in Algorithm 1:

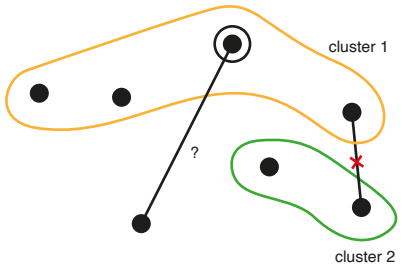
Algorithm 1 Layout annotation process

1: A reference document for annotation is selected.
2: A candidate document is selected based on visual similarity, *vendor id*, *vendor address* or *vendor tax id*. Documents that already belong to the same cluster are filtered out. Also documents from clusters that contain documents with a known negative relationship to the current cluster are filtered out.
3: User approves or rejects the proposed relationship. Clusters are merged if user marks the pair as positive. Information about negative relationship is recorded otherwise.
4: Repeat until user selects another reference document.

The interactive annotation process as mentioned in Algorithm 1 is further visualized in Figure 6.1.



(a) For a selected document (marked by circle), a candidate from cluster 3 is selected based on the visual similarity and shown to the annotator.



(b) User annotates the pair as positive. Cluster 1 and cluster 3 are merged together. New candidate is selected based on visual similarity. Documents from cluster 2 are filtered.

Figure 6.1: Visualization of the interactive clustering algorithm.

A simple Javascript web app with Flask[23] backend was developed for this purpose. User can use keyboard shortcuts **s** (same), **d** (different) and **n** (next) to quickly explore and annotate the dataset. The application is accessible using a browser. The backend runs fully within Docker[24] container. A screenshot of the implemented app is shown in Figure 6.2.

[23] `https://flask.palletsprojects.com/`
[24] `https://www.docker.com/`

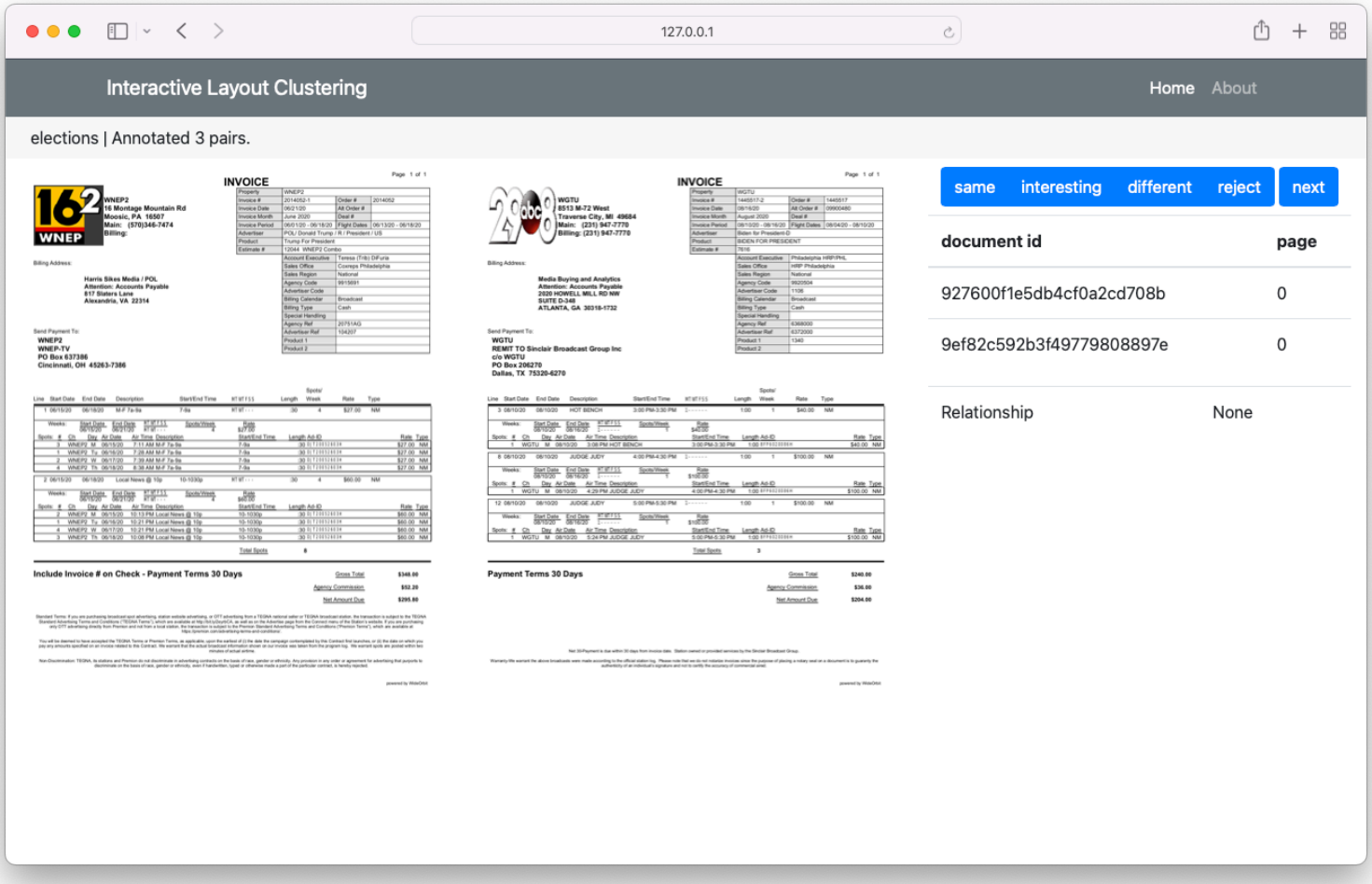


Figure 6.2: Interactive layout annotation tool. Reference page (left) is compared with the document on right.

Conclusions and Future Work

The thesis starts with a comprehensive overview of the document information extraction tasks, datasets and benchmarks. The overview (further enhanced with potential sources of a new dataset and with novel task definitions) was summarized into a position paper [12] submitted for CLEF'22 conference (in review, see Appendix C). We stress the lack of a publicly available dataset of business documents that would also include the layout annotations.

To evaluate the one-shot key information extraction task on business documents, a benchmark was formulated to assess the performance of an one-shot information extraction system with access to a database of previously-annotated documents. A dataset with template annotations was created using a newly implemented annotation tool which significantly simplifies the annotation process. Unfortunately, this proprietary dataset can not be published. A small scale-dataset of publicly available third-party documents is included to illustrate the functionality of the attached code.

A novel representation-learning approach to one-shot document information extraction was proposed. It uses learned field representations to effectively retrieve similar documents and to localize the information within newly received documents. Compared to the *copypaste* baseline, the proposed approach can handle relative shifts in field positions, but underperforms the baseline on fields with fixed positions within the template. The training objective, inspired by contrastive learning, comprises of three loss functions, each improving the final score.

The proposed approach is compared with the baseline methods on the proposed benchmark. It improves the extraction of the *amount total* field — known by its large positional variance — by 24% when having access to a single document of the same template. It should be emphasized that *amount total* is one of the fields with the highest business consequences. The proposed approach could be used to improve the performance of the presented baseline which performs better on other field types. The combination of two engines, however comes with additional computational and maintenance costs.

Future Work

I still strongly believe in the proposed approach where the document is represented as a set of representations of its fields. These representations can be used both for document retrieval, as well as *Key Information Extraction and Localization*. There are still many potential improvements, such as:

- The supervised training used in this thesis could be replaced by a training procedure that would not require layout class annotations. Instead of using two documents of the same layout, two different augmentations of the same image can be used for training. The augmentations could include stochastically masking field contents to motivate the model to represent the field by its surroundings.
- Presenting multiple modalities to the model's input has not improved the performance in the performed experiments. However, this is likely caused by an implementation issue or sub-optimal model architecture w.r.t. to the additional input channels. Multimodal inputs are still worth further exploring.
- The implemented model was evaluated only on 11 field types from each document. This means, that most of the document page was treated as *background*. Note that for fields like *sender address*, there is a prevailing amount of false negative errors. To reduce the bias towards the background class, including more fieldtypes could improve the overall performance despite increasing the number of predicted classes.
- Information extraction methods based on graph convolutional networks [14, 37] have recently shown promising results. Formulating this task as bounding box classification and using graph neural networks is a promising course of research.
- As a followup to the submitted publication [12] (attached as Appendix C), we plan to address the lack of document understanding datasets by publishing a publicly available dataset for business document key information extraction.

Bibliography

[1] Dengel, A. R.; Klein, B. smartfix: A requirements-driven system for document analysis and understanding. In *International Workshop on Document Analysis Systems*, Springer, 2002, pp. 433–444.

[2] Hamza, H.; Belaïd, Y.; et al. Case-based reasoning for invoice analysis and recognition. In *International conference on case-based reasoning*, Springer, 2007, pp. 404–418.

[3] Rusinol, M.; Benkhelfallah, T.; et al. Field extraction from administrative documents by incremental structural templates. In *2013 12th International Conference on Document Analysis and Recognition*, IEEE, 2013, pp. 1100–1104.

[4] d'Andecy, V. P.; Hartmann, E.; et al. Field extraction by hybrid incremental and a-priori structural templates. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, IEEE, 2018, pp. 251–256.

[5] Dhakal, P.; Munikar, M.; et al. One-shot template matching for automatic document data capture. In *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, volume 1, IEEE, 2019, pp. 1–6.

[6] Palm, R. B.; Winther, O.; et al. Cloudscan-a configuration-free invoice analysis system using recurrent neural networks. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, IEEE, 2017, pp. 406–413.

[7] Palm, R. B.; Laws, F.; et al. Attend, copy, parse end-to-end information extraction from documents. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, IEEE, 2019, pp. 329–336.

[8] Majumder, B.; Potti, N.; et al. Representation learning for information extraction from form-like documents. 2020.

[9] He, K.; Zhang, X.; et al. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[10] Ronneberger, O.; Fischer, P.; et al. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[11] Stray, J.; Svetlichnaya, S. Deepform: Extract information from documents. 2020, benchmark. Available from: `https://wandb.ai/deepform/political-ad-extraction`

[12] Skalicky, M.; Simsa, S.; et al. Business Document Information Extraction: Towards Practical Benchmarks. 2022, in review.

[13] Saund, E. Scientific challenges underlying production document processing. In *Document Recognition and Retrieval XVIII*, volume 7874, International Society for Optics and Photonics, 2011, p. 787402.

[14] Holeček, M. Learning from similarity and information extraction from structured documents. *International Journal on Document Analysis and Recognition (IJDAR)*, 2021: pp. 1–17.

[15] Cohen, B.; York, M. Ardent partners' accounts payable metrics that matter in 2020. Technical report, Ardent Partners, 2020, accessed on 15.4.2022. Available from: `https://cbps.canon.com/assets/pdf/ArdentPartners-AP-MTM2020-Canon.pdf`

[16] Han, Y.; Wan, X. Digitization of Text Documents Using PDF/A. *Information Technology and Libraries*, volume 37, no. 1, 2018: pp. 52–64.

[17] Cristani, M.; Bertolaso, A.; et al. Future paradigms of automated processing of business documents. *International Journal of Information Management*, volume 40, 2018: pp. 67–75.

[18] QuickBooks. What is an invoice? Guide, examples, and what to include. online, 1 2021, accessed on 15.4.2022. Available from: `https://quickbooks.intuit.com/r/invoicing/what-is-an-invoice/#invoice-guide-purpose`

[19] Council of European Union. Council Directive (EU) no 2006/112/EC. 2006, `https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32006L0112&from=EN`.

[20] Borchmann, Ł.; Pietruszka, M.; et al. DUE: End-to-End Document Understanding Benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[21] Dengel, A. R. Making documents work: Challenges for document understanding. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, volume 3, Citeseer, 2003, pp. 1026–1026.

[22] Huang, Z.; Chen, K.; et al. Icdar2019 competition on scanned receipt ocr and information extraction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, IEEE, 2019, pp. 1516–1520.

[23] Tito, R.; Mathew, M.; et al. ICDAR 2021 Competition on Document Visual Question Answering. In *International Conference on Document Analysis and Recognition*, Springer, 2021, pp. 635–649.

[24] Antonacopoulos, A.; Clausner, C.; et al. ICDAR2015 competition on recognition of documents with complex layouts-RDCL2015. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, IEEE, 2015, pp. 1151–1155.

[25] Bagdanov, A. D.; Worring, M. Fine-grained document genre classification using first order random graphs. In *Proceedings of Sixth International Conference on Document Analysis and Recognition*, IEEE, 2001, pp. 79–83.

[26] Islam, N.; Islam, Z.; et al. A Survey on Optical Character Recognition System. *CoRR*, volume abs/1710.05703, 2017, 1710.05703. Available from: `http://arxiv.org/abs/1710.05703`

[27] Wewerka, J.; Reichert, M. Towards quantifying the effects of robotic process automation. In *2020 IEEE 24th International Enterprise Distributed Object Computing Workshop (EDOCW)*, IEEE, 2020, pp. 11–19.

[28] Schuster, D.; Muthmann, K.; et al. Intellix–End-User Trained Information Extraction for Document Archiving. In *2013 12th International Conference on Document Analysis and Recognition*, IEEE, 2013, pp. 101–105.

[29] Medvet, E.; Bartoli, A.; et al. A probabilistic approach to printed document understanding. *International Journal on Document Analysis and Recognition (IJDAR)*, volume 14, no. 4, 2011: pp. 335–347.

[30] Cesarini, F.; Francesconi, E.; et al. Analysis and understanding of multiclass invoices. *Document Analysis and Recognition*, volume 6, no. 2, 2003: pp. 102–114.

[31] Lopresti, D.; Wilfong, G. A fast technique for comparing graph representations with applications to performance evaluation. *Document Analysis and Recognition*, volume 6, no. 4, 2003: pp. 219–229.

[32] Esser, D.; Schuster, D.; et al. Automatic indexing of scanned documents: a layout-based approach. In *Document recognition and retrieval XIX*, volume 8297, International Society for Optics and Photonics, 2012, p. 82970H.

[33] Holeček, M.; Hoskovec, A.; et al. Table understanding in structured documents. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 5, IEEE, 2019, pp. 158–164.

[34] Powalski, R.; Borchmann, Ł.; et al. Going full-tilt boogie on document understanding with text-image-layout transformer. In *International Conference on Document Analysis and Recognition*, Springer, 2021, pp. 732–747.

[35] Katti, A. R.; Reisswig, C.; et al. Chargrid: Towards understanding 2d documents. *arXiv preprint arXiv:1809.08799*, 2018.

[36] Denk, T. I.; Reisswig, C. Bertgrid: Contextualized embedding for 2d document representation and understanding. *arXiv preprint arXiv:1909.04948*, 2019.

[37] Liu, X.; Gao, F.; et al. Graph convolution for multimodal information extraction from visually rich documents. *arXiv preprint arXiv:1903.11279*, 2019.

[38] Krieger, F.; Drews, P.; et al. Information extraction from invoices: A graph neural network approach for datasets with high layout variety. In *International Conference on Wirtschaftsinformatik*, Springer, 2021, pp. 5–20.

[39] Vaswani, A.; Shazeer, N.; et al. Attention is all you need. *Advances in neural information processing systems*, volume 30, 2017.

[40] Li, P.; Gu, J.; et al. Selfdoc: Self-supervised document representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5652–5660.

[41] Xu, Y.; Li, M.; et al. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 1192–1200.

[42] Xu, Y.; Lv, T.; et al. LayoutXLM: Multimodal Pre-training for Multilingual Visually-rich Document Understanding. *arXiv preprint arXiv:2104.08836*, 2021.

[43] Devlin, J.; Chang, M.-W.; et al. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[44] McCann, B.; Keskar, N. S.; et al. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*, 2018.

[45] Kumar, A.; Irsoy, O.; et al. Ask me anything: Dynamic memory networks for natural language processing. In *International conference on machine learning*, PMLR, 2016, pp. 1378–1387.

[46] Mathew, M.; Karatzas, D.; et al. DocVQA: A Dataset for VQA on Document Images. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2021, Waikoloa, HI, USA, January 3-8, 2021*, IEEE, 2021, pp. 2199–2208, doi:10.1109/WACV48630.2021.00225. Available from: `https://doi.org/10.1109/WACV48630.2021.00225`

[47] Damodaran, P.; Singh, P.; et al. Zero-shot Task Transfer for Invoice Extraction via Class-aware QA Ensemble. *arXiv preprint arXiv:2108.06069*, 2021.

[48] Antonacopoulos, A.; Bridson, D.; et al. A realistic dataset for performance evaluation of document layout analysis. In *2009 10th International Conference on Document Analysis and Recognition*, IEEE, 2009, pp. 296–300.

[49] Clausner, C.; Antonacopoulos, A.; et al. Icdar2019 competition on recognition of documents with complex layouts-rdcl2019. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, IEEE, 2019, pp. 1521–1526.

[50] Chen, N.; Blostein, D. A survey of document image classification: problem statement, classifier architecture and performance evaluation. *International Journal of Document Analysis and Recognition (IJDAR)*, volume 10, no. 1, 2007: pp. 1–16.

[51] Rastogi, M.; Ali, S. A.; et al. Information Extraction from Document Images via FCA based Template Detection and Knowledge Graph Rule Induction. In *Proceedings of CVPRw*, 2020, pp. 2377–2385, doi:10.1109/CVPRW50498.2020.00287.

[52] Sun, H.; Kuang, Z.; et al. Spatial Dual-Modality Graph Reasoning for Key Information Extraction. *arXiv preprint arXiv:2103.14470*, 2021.

[53] Wang, J.; Liu, C.; et al. Towards Robust Visual Information Extraction in Real World: New Dataset and Novel Solution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[54] Park, S.; Shin, S.; et al. CORD: A Consolidated Receipt Dataset for Post-OCR Parsing. In *Workshop on Document Intelligence at NeurIPS 2019*, 2019.

[55] Stanisławek, T.; Graliński, F.; et al. Kleister: key information extraction datasets involving long documents with complex layouts. In *International Conference on Document Analysis and Recognition*, Springer, 2021, pp. 564–579.

[56] Guillaume Jaume, J.-P. T., Hazim Kemal Ekenel. FUNSD: A Dataset for Form Understanding in Noisy Scanned Documents. In *Accepted to ICDAR-OST*, 2019.

[57] Harley, A. W.; Ufkes, A.; et al. Evaluation of Deep Convolutional Nets for Document Image Classification and Retrieval. In *International Conference on Document Analysis and Recognition (ICDAR)*, 2015.

[58] Francois, C. *Deep learning with Python*. Manning Publications Company, 2017.

[59] Goodfellow, I.; Bengio, Y.; et al. *Deep learning*. MIT press, 2016.

[60] Deng, J.; Dong, W.; et al. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.

[61] Abadi, M.; Agarwal, A.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015, software available from tensorflow.org. Available from: `https://www.tensorflow.org/`

[62] Chen, T.; Kornblith, S.; et al. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, PMLR, 2020, pp. 1597–1607.

[63] Krizhevsky, A.; Sutskever, I.; et al. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, volume 25, 2012.

[64] Schroff, F.; Kalenichenko, D.; et al. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[65] Hermans, A.; Beyer, L.; et al. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[66] Lin, T.-Y.; Goyal, P.; et al. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[67] Kingma, D. P.; Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[68] Zaheer, M.; Reddi, S.; et al. Adaptive methods for nonconvex optimization. *Advances in neural information processing systems*, volume 31, 2018.

[69] Duchi, J.; Hazan, E.; et al. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, volume 12, no. 7, 2011.

[70] Badrinarayanan, V.; Kendall, A.; et al. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, volume 39, no. 12, 2017: pp. 2481–2495.

[71] Merkel, D. Docker: lightweight linux containers for consistent development and deployment. *Linux journal*, volume 2014, no. 239, 2014: p. 2.

[72] Chollet, F.; et al. Keras. 2015. Available from: `https://github.com/fchollet/keras`

[73] Johnson, J.; Douze, M.; et al. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, volume 7, no. 3, 2019: pp. 535–547.

Appendix A

Acronyms

BD	Business Document
CNN	Convolutional Neural Network
DU	Document Understanding
GNN	Graph Neural Network
GPU	Graphical Processing Unit
IE	Information Extraction
KIE	Key Information Extraction
KILE	Localized Key Information Extraction
NLP	Natural Language Processing
OCR	Optical Character Recognition
VRD	Visually Rich Document

Appendix B

Contents of Enclosed Medium

- `readme.txt` ..... readme describing the attached files
- `src` ..... directory with source codes
  - `Dockerfile.annotator` ..... annotation tool dockerfile
  - `Dockerfile.train` ..... dockerfile used for training
  - `Makefile` ..... development server shortcuts
  - `README.md` ..... readme describing how to run the code
  - `data` ..... directory containing datasets
  - `docker-compose.annotator.yml` ..... annotation tool docker-compose
  - `docker-compose.train.gpu.yml` ..... training gpu docker-compose
  - `docker-compose.train.yml` ..... training docker-compose
  - `logs` ..... TensorBoard logs
  - `models` ..... model checkpoints
  - `requirements.txt` ..... python dependencies
  - `transfer` ..... implemented source codes
- `text` ..... thesis text directory
  - `skalimat.pdf` ..... thesis text in PDF format

Appendix C

Business Document Information Extraction: Towards Practical Benchmarks

Business Document Information Extraction: Towards Practical Benchmarks

Matyáš Skalický, Štěpán Šimsa, Michal Uřičář, and Milan Šulc

Rossum.ai
{matyas.skalicky, stepan.simsa, michal.uricar, milan.sulc}@rossum.ai

Abstract. Information extraction from semi-structured documents is crucial for frictionless business-to-business (B2B) communication. While machine learning problems related to *Document Information Extraction* (IE) have been studied for decades, many common problem definitions and benchmarks do not reflect domain-specific aspects and practical needs for automating B2B document communication. We review the landscape of Document IE problems, datasets and benchmarks. We highlight the practical aspects missing in the common definitions and define the *Key Information Localization and Extraction* (KILE) and *Line Item Recognition* (LIR) problems. There is a lack of relevant datasets and benchmarks for Document IE on semi-structured business documents, as their content is typically legally protected or sensitive. We discuss potential sources of available documents including synthetic data.

Keywords: Document Understanding · Survey · Benchmarks · Datasets

1 Introduction

The majority of B2B communication takes place through the exchange of *semi-structured*[1] *business documents* (BD) such as invoices, purchase orders and delivery notes. Automating information extraction from such documents has a considerable potential to reduce repetitive manual work and to streamline business communication. There have been efforts to provide standards for electronic data interchange of BD metadata [5,52,7]. Despite, e.g., electronic invoices taking place rapidly [16], the standards did not get globally adopted, none of them prevails, and most are not inter-operable [21].

Machine learning (ML), natural language processing (NLP), and computer vision problems related to *Document Understanding* and *Document IE* have been studied for decades. Despite the major potential of IE from semi-structured business documents, published research on *Document IE* often focuses on other domains [100,73,13,14,39,38], and many of the traditionally defined tasks and benchmarks do not reflect domain-specific ML aspects and pitfalls of IE from

[1] The term *semi-structured documents* is commonly used in different meanings: Some use it for text files containing semi-structured data [94], such as XML files. We use the term to refer to visually rich documents without a fixed layout [66].

semi-structured BDs. Papers dealing with IE from business documents typically publish their results on private datasets [40,19,34,59,60,47,97,69], hindering reproducibility and cross-evaluation. This is caused by the absence of a large public dataset of semi-structured BDs, noted by several authors [60,77,20,41].

The contributions of this position paper are threefold: first, we provide a review of IE problems, datasets and benchmarks relevant to semi-structured business documents. Second, we identify unaddressed aspects of the tasks and formulate new definitions for *Key Information Localization and Extraction* and *Line Item Recognition*. Third, we stress the lack of a large-scale dataset of semi-structured BDs, and we discuss potential sources of documents for such dataset.

2 Document Information Extraction Problems

2.1 Key Information Extraction (KIE) and Localization (KILE)

Most formulations of KIE come from NLP, where it is usually defined as the extraction of values of a fixed set of entities/classes from an unstructured text source into a structured form [43,96,56,35,96]. Based on the document representation, Garncarek et al. [25] categorize KIE into three groups: (i) sequence-based (working with serialized text [37]), (ii) graph-based (modeling each doc./page as a graph with nodes corresponding to textual segments and their features [17,41] [32]), and (iii) grid-based (treating documents as a 2D grid of token embeddings [40,19]). Sequence-based KIE is closely related to *Named Entity Recognition* (NER) [43] — a sub-task of KIE [96,48] dealing with sequence tagging problems. Borchmann et al. [6] say that (end-to-end) KIE, unlike NER, does not assume token-level annotations are available. The task is also referred to as Slot Filling [60], meaning that a pre-defined slot is filled with the extracted text.

Note that the common definitions of KIE, as well as some of the datasets [73,6], do not require the location of the extracted information within the document. While the localization is typically not crucial w.r.t. to the downstream task, it plays a vital role in applications that require human validation. We extend the definition by explicitly including the localization:

Definition 1 (KILE)**.** Given a document, the goal of *Key Information Localization and Extraction* (KILE) is to localize (e.g., by a bounding box) fields of each pre-defined category (*key*), read out their values, and aggregate the values to extract the key information of each category.

Compared to *Semantic Entity Recognition*, as defined by Xu et al. [93], bounding boxes in KILE are not limited to individual words (tokens).

2.2 Table Extraction and Line Items

Table Understanding [33] and *Table Extraction* (TE) [26,98] are problems where the tabular structure is crucial for IE. Unlike KIE, which outputs individual fields independently, TE typically deals with a list of (line) items [19,33,60,48,4], each consisting of a tuple of fields (e.g., *goods* and *price*).

Fig. 1. Example of a table structure where field type is not uniquely determined by its column. Source: https://rossum.ai/help/article/extracting-nested-values-line-items.

In simple tables, columns determine the field type and rows determine which item the value belongs to. The table can hence be represented as a grid [68,78]. A bottom-up approach [62,98] can handle more complex tables as in Fig. 1, without relying on a row or column detection. Detected cells or fields can be converted to table structure (determining the line items and columns) in a post-processing step, e.g., spatial clustering [98]. Other works [99,46] tackle the table extraction by directly solving an image-to-markup (e.g., XML or TEX) problem.

We argue that the problem definition should not rely on the structure but rather reflect the information to be extracted and stored. This is close to the problem of detecting the area belonging to a single line item [19]. We define *Line Item* (LI) and the task of *Line Item Recognition* (LIR) as follows:

Definition 2 (LI)**.** A *Line Item* is a tuple of fields describing a single object instance to be extracted, e.g., a row in a table.

Definition 3 (LIR)**.** Given an image of a document page or of a table, the goal of *Line Item Recognition* is to detect all LI present in the section, classify them into a fixed set of classes (e.g., *ordered item*, *discount*, . . .) and for every detected LI, localize and extract key information (as in Definition 1) related to it.

Note that this definition of LIR allows: (i) detection of several tables with different item types, as well as different item types within a single table; (ii) a single field (e.g., a date) to belong to several line items.

2.3 One-Shot Learning for Information Extraction

Layouts of business documents vary greatly, even within a single document type. Thousands of invoice templates are available, and vendors often further adjust them to their own needs. Systems without the ability of fast re-training are at risk of degraded performance when faced with a shift in the incoming data distribution [29], such as when presented with previously unseen layouts.

Improving IE with each processed document is known as a one-shot [20] / online [74] template matching, case-based reasoning [29], or configuration-free IE [69]. This includes systems that reuse annotations of similar documents in the database [20,69] or iteratively build and refine a representation of a document class [29,53,67,17]. Annotations of documents' templates are not part of any public IE dataset of sufficient size.

2.4 Other document IE problems and tasks

Optical Character Recognition (OCR) [72], handwritten OCR [31], scene text recognition [95,2], including (sub)word or text-line level predictions, are standard problems with reviews and comparisons available [28,54,58,36]. While highly relevant to the document IE, this paper aims at the "higher-level" document IE problems, often assuming text extracted from PDF or OCR is available.

Document Layout Analysis (DLA) is typically posed as an object detection problem: given a document page, find the minimum bounding boxes (or other area representation [1,15]) of layout elements such as *Paragraph*, *Heading*, *Table*, *Figure*, or *Caption*. Most DLA datasets [23,1,100,15] contain such layout annotations for scientific and technical publications and magazines.

Extraction of Key-Value Pairs (KVP) refers to recognizing pairs of linked data items where the key is used as a unique identifier for the value. This task usually consists of semantic labeling and semantic linking [92,93]. Contrary to KIE, KVP extraction does not require the set of keys to be fixed. It also assumes that both key and value are present in the document. This may be useful, e.g., to extract data from unknown forms. However, in semi-structured business documents, it is pretty standard that the keys of interests (known in advance) are not explicitly present in the document.

Question Answering (QA), also known as *Machine Reading Comprehension*, is a common problem in information retrieval and NLP. The goal is to automatically answer questions formulated in natural language. Many NLP tasks can be reformulated as QA [51,42]. Similar to KIE, QA can be extended to incorporate visual information to Visual Question Answering (VQA) [50]. VQA system may also interpret and extract content from the figures, diagrams, and other non-textual elements.

KIE can be formulated as an instance of VQA. However, we typically know which classes of key information should be extracted, rendering the natural language interface unnecessary.

3 Semi-Structured Business Document Datasets

Publications on business document IE are often based on private datasets [40,19] [34,59,60,47,97,69]. Due to the documents' sensitive content, authors are typically not allowed to share the experimental data. Large third-party sources like common crawl are publicly available; however, re-publishing such data may pose legal issues. For example, a large common crawl dataset of PDF documents by Xu et al. [93] was not published, while pre-training on it was crucial for the proposed method, and the C4 dataset [64] is shared in the form of code that extracts it directly from Common Crawl.

Publicly available datasets for KI(L)E from BDs are summarized in Table 1: Most of them are relatively small and only contain a few annotated field types. The two largest datasets contain only receipts. The table does not include datasets without KIE annotations — RVL-CDIP [30] (classification), FUNSD [27] and XFUND [93] (entities without fieldtype), NIST [84] (forms

Table 1. Overview of datasets related to KI(L)E from semi-structured BDs.

name	document type	docs	fieldtypes	source	multipage	lang.	type
WildReceipt [76]	receipts	1740	25	photo	no	en	KILE
Ghega [53]	patents/datasheets	246	11/8	scan	yes	en	KILE
EPHOIE [79]	chinese forms	1494	10	scan	no	zh	KILE
CORD [61]	receipts	11000	42	photo	no	ind	KILE
DeepForm [75]	invoices, orders	1000	6	scan	yes	en	KILE
Kleister Charity [73]	financial reports	2788	8	scan	yes	en	KIE
Kleister NDA [73]	NDA documents	540	4	scan	yes	en	KIE
SROIE [35]	receipts	973	4	scan	no	en	KIE

identification) and DocVQA [50] (QA) – and datasets we were not able to download[2] [8,65,101,3].

Borchmann et al. [6] recently joined and re-formulated several existing document IE datasets to build the DUE benchmark for several document understanding tasks on different document domains. However, DeepForm [75] and Kleister Charity [73] are the only subsets of DUE with business documents annotated for KIE.

While there are many existing datasets for Table Detection and LIR [22,100,71] [26,24,15,70,90,87,18,63,99,98,57], some of them are not accessible anymore [26,24] [70,18]. We find only FinTabNet [98] and SynthTabNet [57] to be relevant to us by covering complex financial tables.

4 Where to Get More Documents

Publicly Available Documents. Business documents are typically not shared publicly due to their private content, often including confidential and personal information. There are exceptions to this rule — e.g., institutions such as governments or charities have to make certain financial documents publicly available for transparency reasons. Databases of such documents have already been used to create public datasets for document IE: Several datasets — IIT-CDIP[44] ⊃ RVL-CDIP[30] ⊃ FUNSD[27], and DocVQA[50] — were built from documents from the UCSF Industry Documents Library[3] [83]. Annual Reports of the S&P 500 companies [88] were used to create FinTabNet [98]. Non-disclosure agreements from the EDGAR[4] database [82], collected for the U.S. Securities and Exchange Commission, were used for the Kleister-NDA [73] dataset. The DeepForm dataset [75] consists of documents related to broadcast stations from the FCC Public Inspection Files [86]. Financial records from the Charity Commission [81] were used to create the Kleister-Charity dataset [73]. Several QA datasets [9,55,101] were also collected from open data sources [91,85,80]. Other

[2] For some only the annotations are available, without the original PDFs/images.
[3] A large proportion of the UCSF Industry Document Library are old documents, often written on a typewriter, which presents a domain shift w.r.t. today's documents.
[4] Automated crawling of the site not allowed: https://www.sec.gov/os/accessing-edgar-data

datasets were build via web search [49,76,10], from Common Crawl [93,64][5], Wikipedia [11,12,13], or platforms for sharing scientific papers [39,100,38].

Synthetic documents. Manual annotation is expensive, and the collection of data from public sources may be limited by the presence of personal data or intellectual property. This reasoning calls for leveraging synthetic datasets. Xu et al. [93] manually replaced the content of publicly available documents with synthetic data. Bensch et al. [4] generate synthetic invoice documents automatically. However, we observe the generated invoices have a plain style and do not resemble the distribution of visual layouts of real business documents. Li et al. [57] synthesized a table dataset of four appearance styles based on existing datasets [99,98,46].

We consider three ways to define layouts to be filled with synthetic data: (i) manual design — allowing to create precisely the layouts of interest, but costly at scale, (ii) extraction from public documents followed by sensitive anonymization like in [93], and (iii) using a generative model, e.g., to generate realistic layouts dissimilar to those already present in the dataset — we consider such a problem statement an interesting open research problem.

5 Discussion and Future Work

We argue that the problems of KILE and LIR, as defined in Sec. 2, are crucial for automating B2B document communication, where key information has to be extracted from localized fields and line items. The review of public datasets in Sec. 3 shows that — except for receipts [76,61,35] — semi-structured business documents like invoices, orders, and delivery notes are underrepresented in document IE. Based on manual inspection of selected documents from publicly available sources in Sec. 4, we noticed the distribution of BD differs significantly among different sources. An ideal dataset should cover a large variety of visual styles and layouts and provide diagnostic subsets [6] to differentiate errors in various special cases. Due to high annotation costs and possibly legally protected content of business documents, synthetic data are a potentially affordable alternative for building a large-scale dataset. While synthetic data have been proven successful for OCR [45], the potential of data synthesis for BD IE has not yet been fulfilled: existing attempts either target other tasks and document types [93] or do not reflect the rich visual distribution of semi-structured business documents [4]. An advantage of generating synthetic documents of a given layout is the known layout annotation for benchmarking one-shot information extraction.

To enable benchmarking of information extraction on data and tasks highly relevant to real-world application scenarios, in our future work, we are preparing a large-scale public dataset of semi-structured business documents, following the observations and points made in this paper.

[5] CC-MAIN-2022-05 contains almost 3 billion documents out of which 0.84% are PDFs [89] – however, most of them are not semi-structured business documents.

References

1. Antonacopoulos, A., Bridson, D., Papadopoulos, C., Pletschacher, S.: A realistic dataset for performance evaluation of document layout analysis. In: Proceedings of ICDAR. pp. 296–300. IEEE (2009)
2. Baek, Y., Lee, B., Han, D., Yun, S., Lee, H.: Character region awareness for text detection. In: Proceedings of the IEEE/CVF CVPR. pp. 9365–9374 (2019)
3. Baviskar, D., Ahirrao, S., Kotecha, K.: Multi-layout invoice document dataset (MIDD): A dataset for named entity recognition. Data (2021). https://doi.org/10.3390/data6070078
4. Bensch, O., Popa, M., Spille, C.: Key information extraction from documents: Evaluation and generator. In: Abbès, S.B., Hantach, R., Calvez, P., Buscaldi, D., Dessì, D., Dragoni, M., Recupero, D.R., Sack, H. (eds.) Proceedings of DeepOntoNLP and X-SENTIMENT. CEUR Workshop Proceedings, vol. 2918, pp. 47–53. CEUR-WS.org (2021)
5. Berge, J.: The EDIFACT standards. Blackwell Publishers, Inc. (1994)
6. Borchmann, Ł., Pietruszka, M., Stanislawek, T., Jurkiewicz, D., Turski, M., Szyndler, K., Graliński, F.: DUE: End-to-end document understanding benchmark. In: Proceeedings of NeurIPS (2021)
7. Bosak, J., McGrath, T., Holman, G.K.: Universal business language v2. 0. Organization for the Advancement of Structured Information Standards (OASIS), Standard (2006)
8. Cesarini, F., Francesconi, E., Gori, M., Soda, G.: Analysis and understanding of multi-class invoices. Document Analysis and Recognition **6**(2), 102–114 (2003)
9. Chaudhry, R., Shekhar, S., Gupta, U., Maneriker, P., Bansal, P., Joshi, A.: LEAF-QA: locate, encode & attend for figure question answering. In: Proceeedings of WACV. pp. 3501–3510. IEEE (2020). https://doi.org/10.1109/WACV45572.2020.9093269
10. Chen, L., Chen, X., Zhao, Z., Zhang, D., Ji, J., Luo, A., Xiong, Y., Yu, K.: Websrc: A dataset for web-based structural reading comprehension. CoRR (2021)
11. Chen, W., Chang, M., Schlinger, E., Wang, W.Y., Cohen, W.W.: Open question answering over tables and text. In: Proceedings of ICLR (2021)
12. Chen, W., Wang, H., Chen, J., Zhang, Y., Wang, H., Li, S., Zhou, X., Wang, W.Y.: TabFact: A large-scale dataset for table-based fact verification. In: Proceedings of ICLR (2020)
13. Chen, W., Zha, H., Chen, Z., Xiong, W., Wang, H., Wang, W.Y.: Hybridqa: A dataset of multi-hop question answering over tabular and textual data. In: Cohn, T., He, Y., Liu, Y. (eds.) Findings of the Association for Computational Linguistics: EMNLP. Findings of ACL, vol. EMNLP 2020, pp. 1026–1036. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.findings-emnlp.91
14. Cho, M., Amplayo, R.K., Hwang, S., Park, J.: Adversarial TableQA: Attention Supervision for Question Answering on Tables. In: Zhu, J., Takeuchi, I. (eds.) Proceedings of ACML. Proceedings of Machine Learning Research, vol. 95, pp. 391–406 (2018)
15. Clausner, C., Antonacopoulos, A., Pletschacher, S.: ICDAR 2019 competition on recognition of documents with complex layouts-rdcl2019. In: Proceedings of ICDAR. pp. 1521–1526. IEEE (2019)
16. Cristani, M., Bertolaso, A., Scannapieco, S., Tomazzoli, C.: Future paradigms of automated processing of business documents. International Journal of Information Management **40**, 67–75 (2018)

17. d'Andecy, V.P., Hartmann, E., Rusinol, M.: Field extraction by hybrid incremental and a-priori structural templates. In: 2018 13th IAPR International Workshop on Document Analysis Systems (DAS). pp. 251–256. IEEE (2018)
18. Deng, Y., Rosenberg, D.S., Mann, G.: Challenges in end-to-end neural scientific table recognition. In: Proceedings of ICDAR. pp. 894–901. IEEE (2019). https://doi.org/10.1109/ICDAR.2019.00148
19. Denk, T.I., Reisswig, C.: Bertgrid: Contextualized embedding for 2d document representation and understanding. arXiv preprint arXiv:1909.04948 (2019)
20. Dhakal, P., Munikar, M., Dahal, B.: One-shot template matching for automatic document data capture. In: Proceeedings of Artificial Intelligence for Transforming Business and Society (AITB). vol. 1, pp. 1–6. IEEE (2019)
21. Directive 2014/55/eu of the european parliament and of the council on electronic invoicing in public procurement (Apr 2014), https://eur-lex.europa.eu/eli/dir/2014/55/oj
22. Fang, J., Tao, X., Tang, Z., Qiu, R., Liu, Y.: Dataset, ground-truth and performance metrics for table detection evaluation. In: Blumenstein, M., Pal, U., Uchida, S. (eds.) Proceedings of IAPR International Workshop on Document Analysis Systems, DAS. pp. 445–449. IEEE (2012). https://doi.org/10.1109/DAS.2012.29
23. Ford, G., Thoma, G.R.: Ground truth data for document image analysis. In: Symposium on Document Image Understanding and Technology. pp. 199–205. Citeseer (2003)
24. Gao, L., Yi, X., Jiang, Z., Hao, L., Tang, Z.: ICDAR2017 competition on page object detection. In: Proceedings of ICDAR. pp. 1417–1422 (2017). https://doi.org/10.1109/ICDAR.2017.231
25. Garncarek, L., Powalski, R., Stanislawek, T., Topolski, B., Halama, P., Turski, M., Gralinski, F.: LAMBERT: layout-aware language modeling for information extraction. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) Proceedings of ICDAR. vol. 12821, pp. 532–547. Springer (2021). https://doi.org/10.1007/978-3-030-86549-8_34
26. Göbel, M.C., Hassan, T., Oro, E., Orsi, G.: ICDAR 2013 table competition. In: Proceedings of ICDAR. pp. 1449–1453. IEEE Computer Society (2013). https://doi.org/10.1109/ICDAR.2013.292
27. Guillaume Jaume, Hazim Kemal Ekenel, J.P.T.: Funsd: A dataset for form understanding in noisy scanned documents. In: Accepted to ICDAR-OST (2019)
28. Hamad, K.A., Mehmet, K.: A detailed analysis of optical character recognition technology. International Journal of Applied Mathematics Electronics and Computers **1**(Special Issue-1), 244–249 (2016)
29. Hamza, H., Belaïd, Y., Belaïd, A.: Case-based reasoning for invoice analysis and recognition. In: International conference on case-based reasoning. pp. 404–418. Springer (2007)
30. Harley, A.W., Ufkes, A., Derpanis, K.G.: Evaluation of deep convolutional nets for document image classification and retrieval. In: International Conference on Document Analysis and Recognition (ICDAR) (2015)
31. He, S., Schomaker, L.: Beyond ocr: Multi-faceted understanding of handwritten document characteristics. Pattern Recognition **63**, 321–333 (2017)
32. Holeček, M.: Learning from similarity and information extraction from structured documents. International Journal on Document Analysis and Recognition (IJDAR) pp. 1–17 (2021)

33. Holeček, M., Hoskovec, A., Baudiš, P., Klinger, P.: Table understanding in structured documents. In: 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW). vol. 5, pp. 158–164. IEEE (2019)
34. Holt, X., Chisholm, A.: Extracting structured data from invoices. In: Proceedings of the Australasian Language Technology Association Workshop 2018. pp. 53–59 (2018)
35. Huang, Z., Chen, K., He, J., Bai, X., Karatzas, D., Lu, S., Jawahar, C.V.: ICDAR2019 competition on scanned receipt OCR and information extraction. In: Proceedings of ICDAR. pp. 1516–1520. IEEE (2019). https://doi.org/10.1109/ICDAR.2019.00244
36. Islam, N., Islam, Z., Noor, N.: A survey on optical character recognition system. arXiv preprint arXiv:1710.05703 (2017)
37. Jiang, J.: Information extraction from text. In: Mining text data, pp. 11–41. Springer (2012)
38. Jobin, K.V., Mondal, A., Jawahar, C.V.: Docfigure: A dataset for scientific document figure classification. In: 13th IAPR International Workshop on Graphics Recognition, GREC@ICDAR. pp. 74–79. IEEE (2019). https://doi.org/10.1109/ICDARW.2019.00018
39. Kardas, M., Czapla, P., Stenetorp, P., Ruder, S., Riedel, S., Taylor, R., Stojnic, R.: Axcell: Automatic extraction of results from machine learning papers. arXiv preprint arXiv:2004.14356 (2020)
40. Katti, A.R., Reisswig, C., Guder, C., Brarda, S., Bickel, S., Höhne, J., Faddoul, J.B.: Chargrid: Towards understanding 2d documents. arXiv preprint arXiv:1809.08799 (2018)
41. Krieger, F., Drews, P., Funk, B., Wobbe, T.: Information extraction from invoices: A graph neural network approach for datasets with high layout variety. In: International Conference on Wirtschaftsinformatik. pp. 5–20. Springer (2021)
42. Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., Zhong, V., Paulus, R., Socher, R.: Ask me anything: Dynamic memory networks for natural language processing. In: Balcan, M., Weinberger, K.Q. (eds.) Proceedings of ICML. vol. 48, pp. 1378–1387. JMLR.org (2016)
43. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. In: Knight, K., Nenkova, A., Rambow, O. (eds.) Proceedings of NAACL HLT. pp. 260–270 (2016). https://doi.org/10.18653/v1/n16-1030
44. Lewis, D., Agam, G., Argamon, S., Frieder, O., Grossman, D., Heard, J.: Building a test collection for complex document information processing. In: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 665–666 (2006)
45. Li, J., Wang, S., Wang, Y., Tang, Z.: Synthesizing data for text recognition with style transfer. Multimedia Tools and Applications **78**(20), 29183–29196 (2019)
46. Li, M., Cui, L., Huang, S., Wei, F., Zhou, M., Li, Z.: Tablebank: Table benchmark for image-based table detection and recognition. In: Calzolari, N., Béchet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S. (eds.) Proceedings of The 12th Language Resources and Evaluation Conference, LREC. pp. 1918–1925 (2020)
47. Liu, W., Zhang, Y., Wan, B.: Unstructured document recognition on business invoice. Mach. Learn., Stanford iTunes Univ., Stanford, CA, USA, Tech. Rep (2016)

48. Majumder, B.P., Potti, N., Tata, S., Wendt, J.B., Zhao, Q., Najork, M.: Representation learning for information extraction from form-like documents. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J.R. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL. pp. 6495–6504 (2020). https://doi.org/10.18653/v1/2020.acl-main.580
49. Mathew, M., Bagal, V., Tito, R., Karatzas, D., Valveny, E., Jawahar, C.: Infographicvqa. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1697–1706 (2022)
50. Mathew, M., Karatzas, D., Jawahar, C.V.: Docvqa: A dataset for VQA on document images. In: Proceedings of WACV. pp. 2199–2208. IEEE (2021). https://doi.org/10.1109/WACV48630.2021.00225
51. McCann, B., Keskar, N.S., Xiong, C., Socher, R.: The natural language decathlon: Multitask learning as question answering. CoRR (2018)
52. Meadows, B., Seaburg, L.: Universal business language 1.0. Organization for the Advancement of Structured Information Standards (OASIS) (2004)
53. Medvet, E., Bartoli, A., Davanzo, G.: A probabilistic approach to printed document understanding. Int. J. Document Anal. Recognit. **14**(4), 335–347 (2011). https://doi.org/10.1007/s10032-010-0137-1
54. Memon, J., Sami, M., Khan, R.A., Uddin, M.: Handwritten optical character recognition (ocr): A comprehensive systematic literature review (slr). IEEE Access **8**, 142642–142668 (2020)
55. Methani, N., Ganguly, P., Khapra, M.M., Kumar, P.: Plotqa: Reasoning over scientific plots. In: Proceedings of WACV. pp. 1516–1525 (2020). https://doi.org/10.1109/WACV45572.2020.9093523
56. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticæ Investigationes pp. 3–26 (2007). https://doi.org/https://doi.org/10.1075/li.30.1.03nad
57. Nassar, A., Livathinos, N., Lysak, M., Staar, P.W.J.: Tableformer: Table structure understanding with transformers. CoRR **abs/2203.01017** (2022). https://doi.org/10.48550/arXiv.2203.01017
58. Nayef, N., Patel, Y., Busta, M., Chowdhury, P.N., Karatzas, D., Khlif, W., Matas, J., Pal, U., Burie, J.C., Liu, C.l., et al.: ICDAR 2019 robust reading challenge on multi-lingual scene text detection and recognition—rrc-mlt-2019. In: Proceedings of ICDAR. pp. 1582–1587. IEEE (2019)
59. Palm, R.B., Laws, F., Winther, O.: Attend, copy, parse end-to-end information extraction from documents. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 329–336. IEEE (2019)
60. Palm, R.B., Winther, O., Laws, F.: Cloudscan - A configuration-free invoice analysis system using recurrent neural networks. In: Proceedings of ICDAR. pp. 406–413. IEEE (2017). https://doi.org/10.1109/ICDAR.2017.74
61. Park, S., Shin, S., Lee, B., Lee, J., Surh, J., Seo, M., Lee, H.: Cord: A consolidated receipt dataset for post-ocr parsing. In: Workshop on Document Intelligence at NeurIPS 2019 (2019)
62. Prasad, D., Gadpal, A., Kapadni, K., Visave, M., Sultanpure, K.: Cascadetabnet: An approach for end to end table detection and structure recognition from image-based documents. In: Proceedings of CVPRw. pp. 2439–2447 (2020). https://doi.org/10.1109/CVPRW50498.2020.00294
63. Qasim, S.R., Mahmood, H., Shafait, F.: Rethinking table recognition using graph neural networks. In: Proceedings of ICDAR. pp. 142–147. IEEE (2019). https://doi.org/10.1109/ICDAR.2019.00031

64. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683 (2019)
65. Rastogi, M., Ali, S.A., Rawat, M., Vig, L., Agarwal, P., Shroff, G., Srinivasan, A.: Information extraction from document images via FCA based template detection and knowledge graph rule induction. In: Proceedings of CVPRw. pp. 2377–2385 (2020). https://doi.org/10.1109/CVPRW50498.2020.00287
66. Riba, P., Dutta, A., Goldmann, L., Fornés, A., Ramos, O., Lladós, J.: Table detection in invoice documents by graph neural networks. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 122–127. IEEE (2019)
67. Rusinol, M., Benkhelfallah, T., Poulain dAndecy, V.: Field extraction from administrative documents by incremental structural templates. In: 2013 12th International Conference on Document Analysis and Recognition. pp. 1100–1104. IEEE (2013)
68. Schreiber, S., Agne, S., Wolf, I., Dengel, A., Ahmed, S.: DeepDeSRT: Deep Learning for Detection and Structure Recognition of Tables in Document Images. In: Proceedings of ICDAR. pp. 1162–1167 (2017). https://doi.org/10.1109/ICDAR.2017.192
69. Schuster, D., Muthmann, K., Esser, D., Schill, A., Berger, M., Weidling, C., Aliyev, K., Hofmeier, A.: Intellix–end-user trained information extraction for document archiving. In: 2013 12th International Conference on Document Analysis and Recognition. pp. 101–105. IEEE (2013)
70. Shahab, A., Shafait, F., Kieninger, T., Dengel, A.: An open approach towards the benchmarking of table structure recognition systems. In: Doermann, D.S., Govindaraju, V., Lopresti, D.P., Natarajan, P. (eds.) The Ninth IAPR International Workshop on Document Analysis Systems, DAS. pp. 113–120 (2010). https://doi.org/10.1145/1815330.1815345
71. Siegel, N., Lourie, N., Power, R., Ammar, W.: Extracting scientific figures with distantly supervised neural networks. In: Chen, J., Gonçalves, M.A., Allen, J.M., Fox, E.A., Kan, M., Petras, V. (eds.) Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries, JCDL. pp. 223–232 (2018). https://doi.org/10.1145/3197026.3197040
72. Smith, R.: An overview of the tesseract ocr engine. In: Ninth international conference on document analysis and recognition (ICDAR 2007). vol. 2, pp. 629–633. IEEE (2007)
73. Stanisławek, T., Graliński, F., Wróblewska, A., Lipiński, D., Kaliska, A., Rosalska, P., Topolski, B., Biecek, P.: Kleister: key information extraction datasets involving long documents with complex layouts. In: International Conference on Document Analysis and Recognition. pp. 564–579. Springer (2021)
74. Stockerl, M., Ringlstetter, C., Schubert, M., Ntoutsi, E., Kriegel, H.P.: Online template matching over a stream of digitized documents. In: Proceedings of the 27th International Conference on Scientific and Statistical Database Management. pp. 1–12 (2015)
75. Stray, J., Svetlichnaya, S.: Deepform: Extract information from documents (2020), https://wandb.ai/deepform/political-ad-extraction, benchmark
76. Sun, H., Kuang, Z., Yue, X., Lin, C., Zhang, W.: Spatial dual-modality graph reasoning for key information extraction. arXiv preprint arXiv:2103.14470 (2021)
77. Sunder, V., Srinivasan, A., Vig, L., Shroff, G., Rahul, R.: One-shot information extraction from document images using neuro-deductive program synthesis. arXiv preprint arXiv:1906.02427 (2019)

78. Tensmeyer, C., Morariu, V.I., Price, B., Cohen, S., Martinez, T.: Deep splitting and merging for table structure decomposition. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 114–121. IEEE (2019)
79. Wang, J., Liu, C., Jin, L., Tang, G., Zhang, J., Zhang, S., Wang, Q., Wu, Y., Cai, M.: Towards robust visual information extraction in real world: New dataset and novel solution. In: Proceedings of the AAAI Conference on Artificial Intelligence (2021)
80. Web: Annual reports. https://www.annualreports.com/, accessed: 2022-04-28
81. Web: Charity Commission for England and Wales. https://apps.charitycommission.gov.uk/showcharity/registerofcharities/RegisterHomePage.aspx, accessed: 2022-04-22
82. Web: EDGAR. https://www.sec.gov/edgar.shtml, accessed: 2022-04-22
83. Web: Industry Documents Library. https://www.industrydocuments.ucsf.edu/, accessed: 2022-04-22
84. Web: NIST Special Database 2. https://www.nist.gov/srd/nist-special-database-2, accessed: 2022-04-25
85. Web: Open Government Data (OGD) Platform India. https://visualize.data.gov.in/, accessed: 2022-04-22
86. Web: Public Inspection Files. https://publicfiles.fcc.gov/, accessed: 2022-04-22
87. Web: Scitsr. https://github.com/Academic-Hammer/SciTSR, accessed: 2022-04-26
88. Web: S&P 500 Companies with Financial Information. https://www.spglobal.com/spdji/en/indices/equity/sp-500/#data, accessed: 2022-04-25
89. Web: Statistics of Common Crawl Monthly Archives — MIME Types. https://commoncrawl.github.io/cc-crawl-statistics/plots/mimetypes, accessed: 2022-04-22
90. Web: Tablebank. https://github.com/doc-analysis/TableBank, accessed: 2022-04-26
91. Web: World Bank Open Data. https://data.worldbank.org/, accessed: 2022-04-22
92. Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., Zhou, M.: Layoutlm: Pre-training of text and layout for document image understanding. In: Gupta, R., Liu, Y., Tang, J., Prakash, B.A. (eds.) Proceedings on KDD. pp. 1192–1200 (2020). https://doi.org/10.1145/3394486.3403172
93. Xu, Y., Lv, T., Cui, L., Wang, G., Lu, Y., Florêncio, D., Zhang, C., Wei, F.: LayoutXLM: Multimodal Pre-training for Multilingual Visually-rich Document Understanding. CoRR (2021)
94. Yi, J., Sundaresan, N.: A classifier for semi-structured documents. In: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 340–344 (2000)
95. Yu, D., Li, X., Zhang, C., Liu, T., Han, J., Liu, J., Ding, E.: Towards accurate scene text recognition with semantic reasoning networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12113–12122 (2020)
96. Yu, W., Lu, N., Qi, X., Gong, P., Xiao, R.: PICK: processing key information extraction from documents using improved graph learning-convolutional networks. In: Proceedings of ICPR. pp. 4363–4370. IEEE (2020). https://doi.org/10.1109/ICPR48806.2021.9412927
97. Zhao, X., Wu, Z., Wang, X.: CUTIE: learning to understand documents with convolutional universal text information extractor. CoRR **abs/1903.12363** (2019), http://arxiv.org/abs/1903.12363

98. Zheng, X., Burdick, D., Popa, L., Zhong, X., Wang, N.X.R.: Global table extractor (GTE): A framework for joint table identification and cell structure recognition using visual context. In: Proceedings of WACV. pp. 697–706. IEEE (2021). https://doi.org/10.1109/WACV48630.2021.00074
99. Zhong, X., ShafieiBavani, E., Jimeno-Yepes, A.: Image-based table recognition: Data, model, and evaluation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. (eds.) Proceedings of ECCV. pp. 564–580. Springer (2020). https://doi.org/10.1007/978-3-030-58589-1_34
100. Zhong, X., Tang, J., Jimeno-Yepes, A.: Publaynet: Largest dataset ever for document layout analysis. In: Proceedings of ICDAR. pp. 1015–1022. IEEE (Sep 2019). https://doi.org/10.1109/ICDAR.2019.00166
101. Zhu, F., Lei, W., Huang, Y., Wang, C., Zhang, S., Lv, J., Feng, F., Chua, T.: TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) Proceedings International Joint Conference on Natural Language Processing. pp. 3277–3287 (2021). https://doi.org/10.18653/v1/2021.acl-long.254