



Comparing and Evaluating Document Understanding Models on Austrian Receipts

BACHELORARBEIT

zur Erlangung des akademischen Grades

Bachelor of Science

im Rahmen des Studiums

Software & Information Engineering

eingereicht von

Fabian Hartmann

Matrikelnummer 11824496

an der Fakultät für Informatik
der Technischen Universität Wien
Betreuung: Thomas Grechenig

Wien, 17. Jänner 2022

Unterschrift Verfasser

Unterschrift Betreuung



Comparing and Evaluating Document Understanding Models on Austrian Receipts

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Software & Information Engineering

by

Fabian Hartmann

Registration Number 11824496

to the Faculty of Informatics

at the TU Wien

Advisor: Thomas Grechenig

Vienna, 17th January, 2022

Signature Author

Signature Advisor



Comparing and Evaluating Document Understanding Models on Austrian Receipts

BACHELORARBEIT

zur Erlangung des akademischen Grades

Bachelor of Science

im Rahmen des Studiums

Software & Information Engineering

eingereicht von

Fabian Hartmann

Matrikelnummer 11824496

ausgeführt am
Institut für Information Systems Engineering
Forschungsbereich Business Informatics
Forschungsgruppe Industrielle Software
der Fakultät für Informatik der Technischen Universität Wien

Betreuung: Thomas Grechenig

Wien, 17. Jänner 2022

Erklärung zur Verfassung der Arbeit

Fabian Hartmann

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 17. Jänner 2022

Fabian Hartmann

Kurzfassung

Geschäftsdokumente sind ein wesentlicher Bestandteil der täglichen Kommunikation zwischen Unternehmen und Verbrauchern. Bei jedem Austausch von Waren oder Dienstleistungen wird in Österreich ein Kaufbeleg erstellt. Diese Belege enthalten Informationen über das Unternehmen, den Gesamtbetrag und das Datum. Viele Unternehmen betreiben erhebliche Anstrengungen, diese Informationen für nachfolgende Aufgaben wie beispielsweise die Buchhaltung zu digitalisieren. In Kombination mit Document Understanding-Systemen ist eine automatisierte Verarbeitung der relevanten Informationen möglich. Document Understanding ist ein aktives Forschungsthema, das Methoden der Verarbeitung, des Lesens und der Informationsextraktion aus Dokumenten beschreibt.

Diese Arbeit beschäftigt sich mit dem aktuellen Stand der Technik zur Extraktion von Informationen aus Dokumenten und konzentriert sich dabei auf Kassenbelege und Rechnungen. Derzeitige wissenschaftliche Arbeiten befassen sich mit verschiedenen Möglichkeiten der Kodierung von Dokumenten für Machine Learning Modelle. Diese Bachelorarbeit vergleicht die verschiedenen Verfahren der Kodierung und die Leistung der entsprechenden Modelle auf dem Datensatz Scanned Receipt OCR and Information Extraction (SROIE).

Weiters wird untersucht, inwieweit sich dieses Wissen von englischen Quittungen auf ihre österreichischen Pendanten übertragen lassen. Um dies zu erreichen, wurde ein qualitativ hochwertiger Datensatz österreichischer Quittungen kuratiert, mit Transkriptionen und Ground-Truth-Annotationen. Obwohl alle drei Modelle, die auf dem SROIE-Datensatz trainiert wurden, einen F1-Score von mehr als 0,9 (LayoutLM: 0,96; BERT: 0,95; PICK: 0,94) auf dem SROIE-Testset erreichen konnten, zeigt die Leistung auf österreichischen Quittungen (LayoutLM: 0,60; BERT: 0,38; PICK: 0,19) die Notwendigkeit eines gesonderten Trainings auf österreichischen Belegen.

Keywords: *Document Understanding, Informationsextraktion, Österreichische Kassenbons, LayoutLM, BERT, PICK*

Abstract

Business Documents are a vital part of everyday communication between companies and consumers. Each exchange of goods or services creates a purchase receipt in Austria. These receipts contain information regarding the company, total amount, and the date. Many companies are making significant efforts to digitize this information for downstream tasks like accounting. Combining this with Document Understanding systems enables the automated processing of relevant information. Document Understanding is an active research topic that describes the methodology of processing, reading, and extraction of information from documents.

This thesis explores the state-of-the-art approaches to extracting information from documents, focusing on receipts and invoices. Current research investigates different ways of encoding the document for machine learning models. This thesis compares the different encodings and the performance of the corresponding models on the dataset Scanned Receipt OCR and Information Extraction (SROIE).

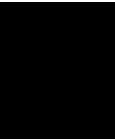
Furthermore, it explores the ability to transfer this knowledge from English receipts to their Austrian counterparts. To achieve this, a high-quality dataset of Austrian receipts was curated, with transcriptions and ground truth annotations. Even though all three models trained on the SROIE dataset could achieve an F1 score of more than 0.9 (LayoutLM: 0.96; BERT: 0.95; PICK: 0.94) on the SROIE test-set, the performance on Austrian receipts (LayoutLM: 0.60; BERT: 0.38; PICK: 0.19) displays the need for custom training on Austrian receipts.

Keywords: *Document Understanding, Information Extraction, Austrian Receipts, LayoutLM, BERT, PICK*

Contents

Kurzfassung	ix
Abstract	xi
Contents	xiii
1 Introduction	1
1.1 Problem Statement	1
1.2 Expected Results	2
1.3 Methodological Approach	2
2 Document Understanding	3
2.1 Optical Character Recognition	3
2.2 Document Analysis	4
2.3 Document Image Classification	5
2.4 Information Extraction	5
2.5 Understanding receipts	7
3 State of the Art	9
3.1 Template Based Approaches	10
3.2 BERT	12
3.3 Language Models with Layout	15
3.4 Grid Based Approaches	22
3.5 Graph Based Approaches	25
3.6 Others	31
3.7 Comparison	32
4 Austrian Receipts	35
4.1 Dataset	36
5 Evaluation	41
5.1 SROIE Preprocessing	43
5.2 Implementation	46
5.3 Results	48
	xiii

5.4 Discussion	50
6 Conclusion and Future Work	53
List of Figures	55
List of Tables	57
Bibliography	59
Weblinks	69
Source-Code	71



Introduction

1.1 Problem Statement

Humans create documents to convey and store information. Businesses make and receive documents as part of acquiring or selling goods [31]. These business documents usually contain a predefined set of data. For instance, invoices have the name of the company and the total amount printed on them [10]. Many companies go to considerable lengths to digitalize this information for tasks further downstream. In Germany, a study for Invoice Reading Systems found that the average cost of processing one invoice sums up to 9€[31]. Using an algorithm eliminates the need for tedious and time-consuming manual work [85]. Business documents like receipts share some layout, and the location within the document conveys a lot of information. Therefore, reading the text on a page is not enough to understand the document [74]. Relying on position only ignores the fact that different companies use different layouts. A slight alteration of the layout can hinder the algorithm from extracting information successfully. A combination of exploiting the positional information as well as the semantics of words is necessary to understand business documents completely [74]. Furthermore, visual information can be included to improve the extractions on receipt images [84].

Because most of the approaches discussed in the literature focus on receipts in the English language, their performance on Austrian receipts is unknown [85]. The difference in Language and structure can influence the performance. A model must identify relevant fields like the total amount or the company reliably. If this is not the case, the model may introduce more errors and create extra work in tasks utilizing the extracted information. In accounting, for example, if expenses do not sum up correctly, time-intensive error-search is needed to fix it.

1.2 Expected Results

This report aims to deliver a theoretical and a practical result. It investigates different approaches and summarizes the state of the art. This work explores different ways to extract information from document images like scanned receipts and invoices. It looks at the underlying research topic of document understanding and the shortcomings of classical NLP approaches to extract the relevant data from these documents. There is a substantial amount of research in the field of document understanding and a variety of similar approaches with unclear relationships to each other. Therefore, this thesis aims to explore the literature and categorize and compare proposed models. Moreover, it aims to help structure future work and give a good starting point for future research.

The practical result focuses on the effectiveness of state-of-the-art algorithms on Austrian receipts. The main question of this work explores the effectiveness and accuracy of an algorithm trained on international data on Austrian receipts (Kassabons). It aims to determine how effective such a system would be if utilized in Austria. Since the layouts for English receipts are not that different from their Austrian counterpart, a model trained on many English receipts should work quite well. In addition, most of the relevant text does not differ that much between linguistically. For example, the date or the total amount have the same format and do not depend on the language.

1.3 Methodological Approach

The first part of the thesis is dedicated to the differences in the various approaches found in the literature, grouping them into categories to enable a more precise overview. It distinguishes the underlying ideas and how they are built and trained. Due to the long history of document analysis and information extraction, the goal is an overview of recent advances and exciting approaches.

The creation of a sample of Austrian receipts is the second part of this thesis. Due to the scarcity of publicly available receipts and the high effort involved in annotating the images, this dataset is of limited size. Because the receipts were collected from personal shopping, there are some restrictions on the variety. Nevertheless, one goal during data collection was to create a set of receipts with different layouts and origins (companies).

The third part of the thesis concerns implementing and training state-of-the-art models. In the course of that, BERT [13], LayoutLM [85], and PICK [87] were implemented. These three approaches cover different aspects of current work in document understanding. The training of the models uses the SROIE dataset to learn about receipts.

Last but not least, the implemented models get evaluated on the Austrian Receipts dataset. To compare the performance, measurements like Precision and Recall are calculated. Due to the limited size, empirical values have limitations; therefore, a more qualitative approach is considered. The mistakes of the models get analyzed in more detail and hypothesized on why the model was unable to predict the field correctly.

Document Understanding

This thesis follows the work of Subramani et al. [74] and considers document understanding to be the automated reading, processing, and extraction of information on documents. This includes documents like invoices, receipts, contracts, and many more [52].

The workshop on document intelligence [Web5] and the International Conference on Document Analysis and Recognition (ICDAR) [Web1] display the scientific interest in analyzing and understanding documents. They give a platform for current research to explore different areas in automatically processing documents. There are a variety of problems in document understanding fueled by the various applications in healthcare, legal and financial areas [4]. This section aims to give an overview of document understanding and explores different problems in this field. It includes and extends problems discussed in the works of Baviskar et al. [4] and Subramani et al. [74].

2.1 Optical Character Recognition

Optical Character Recognition (OCR) is the process of detecting and classifying characters in an image [49]. It simulates reading the text on a document image and transcribing it into a computer-readable format. OCR consists of 5 stages, resulting in the transcription of text segments with their corresponding location in the image [49]. The first two steps consider digitization and pre-processing. These steps create a digital image (scanning or taking a picture) and pre-process it for the subsequent steps. Common pre-processing steps include thresholding or de-skewing to mitigate potential pitfalls in later stages [49]. The third step consists of two sub-tasks: text detection and segmentation. Text detection aims to localize the bounding boxes of text segments within the image [34]. The result should contain as little background as possible. This sub-task can also be achieved with text localization (finding candidate text regions) and text verification (verifying the candidates as text and non-text areas) [9]. Text segmentation deals with the challenging problem of separating the text instances into single lines and single characters [9]. The

fourth stage, text recognition, classifies the cropped image into a character based on its visual appearance [49]. The last step, post-processing, tries to correct false transcriptions and groups the characters into words and text segments.

Deep learning approaches can get incorporated in multiple stages in the system [46]. Many recently proposed systems use neural networks in text detection, either by treating the problem as an object detection problem or by identifying sub-text components (pixels, characters) within the image [46]. Because these networks require a lot of training data, real datasets are too small, as they only contain a few thousand images [9]. Therefore, most approaches use synthetic images, where the text gets incorporated into the picture and purposefully skewed and transformed not to overfit the network [9]. The realistic datasets are helpful in evaluating the system and compare models.

2.2 Document Analysis

Document Layout Analysis is the task of detecting and annotating regions of interest within a document [74]. The task is best illustrated by the problems it tries to solve.

Different datasets focus on various challenges in document layout analysis. For example, text line segmentation or character segmentation represents a document layout analysis problem, where the regions of interest are either lines or characters [5]. The International Conference on Document Recognition and Analysis (ICDAR) published a variety of datasets exploring different challenges. The dataset of 150 pages of medieval manuscripts investigates the semantic structure of documents with complex layouts [73]. The goal is to annotate the document with four classes: main-text-body, decorations, comments, and background. Two datasets published in 2019 explore table detection and recognition on modern and archival documents respectively [23]. The challenge consisted of identifying the table regions and the table structure. The difficulty lies within the variations of the tables and noise, like handwritten annotations overlapping the tables [23]. The model of Barman et al. [3] tackles a different type of layout analysis. Their model breaks historical newspapers into different classes (serial, weather, death notice, stocks, and without annotations). Newspapers are difficult because of the complex layouts. There are multiple rows, heterogeneous elements with different layouts characteristics, and various types of content [3].

Overall, document layout analysis systems try to identify the different regions of interest within the document image [5]. Such systems are useful in document retrieval and information extraction systems to find important regions within the document.

There are three common approaches to partition the documents into the desired segments [5]. The Bottom-Up strategy iteratively combines fine-grained segments into more extensive regions. It usually starts at fine levels of the document (such as pixels or characters) and combines them into bigger areas until it reaches a stopping criterion [5]. The Top-Down strategy works in reverse and splits the document into smaller elements at each step. Hybrid approaches combine the two strategies [5].

A variety of recent approaches on document layout analysis consider visual or textual information or a combination of both. For example, a model on newspaper segmentation consisting of a U-Net architecture combines visual and textual embeddings after the first convolution, resulting in a more profound representation [3]. The combination outperformed models on purely textual or visual information. Alternatively, CascadeTabNet detects tables and performs structure recognition solely on visual information [58].

2.3 Document Image Classification

Document image classification is strongly related to text and image classification. Text classification is a classical problem in Natural Language Processing that deals with assigning labels to sentences, paragraphs, or whole texts [32]. The algorithms classify text segments with a label from a predefined set of classes [32]. Text classification has a wide range of applications, from spam detection and sentiment analysis to news categorization [51]. Text data can come from different sources like chats, reviews, and social media. Sentiment analysis, for example, extracts the intention or opinions behind the sentences [51]. Product reviews carry information about how much the author liked or disliked the product. Based on the text, a classification system can predict the number of stars a given review had.

Image Classification follows the same idea but tags images instead of texts [33]. The ImageNet datasets [64], for example, consists of over one million images with 1000 class labels, from goldfish over baseball to volcano.

Document Image Classification deals with labeling an image of a document with a class [42]. For example, a system could aim to classify a document image as either a letter, scientific report, or invoice document. Liu et al. [42] proposed four different categories of systems for document image classification: textual-based methods, structural-based methods, visual-based methods, and hybrid methods. Textual-based methods exploit the textual information extracted from the document image, which translates to text classification. The visual-based methods relate to the image classification task, as they only consider the visual appearance. The structural-based methods rely on the layout of the document and contain template matching-based methods and graph matching-based methods. Because documents have written text, a visual appearance, and a structure, hybrid approaches can combine the different types of information [42]. For example, document image classification approaches can fuse textual and visual information to improve their performance [2].

2.4 Information Extraction

Information Extraction is the process of analyzing text to identify semantic entities and their relationships [17]. It aims at making the structure and information within the text explicit and usable (recorded in databases) [17]. Information extraction systems are especially useful in areas with a lot of data and frequent semantic entities [17].

On unstructured classical text, there are multiple components in information extraction, from Named Entity Recognition (NER) to Semantic Analysis and cross-document understanding [17]. Named Entity Recognition (NER) aims to recognize "descriptive entities" within the text [1]. It can identify and classify predefined domain-independent semantic entities like locations or organizations or domain-specific entities like drugs and diseases [1]. NER can be a part of various natural language processing tasks, such as text understanding and information extraction [39]. Given a sequence of text-tokens, NER outputs a list of all entities found within the text, with the corresponding token sequence [39]. The definition of a token can vary between algorithms and applications. Many systems use either characters, word-pieces, or whole words as their tokens [39]. An example from the CoNLL03 dataset [67] illustrates the problem. This dataset aims to identify persons, locations, and organizations in the text.

U.N. official Ekeus heads for Baghdad.

The corresponding named entities are:

- Ekeus [Person]
- U.N. [Organisation]
- Baghdad [Location]

Recently, deep learning-based NER systems achieved state-of-the-art results [39]. Most of these systems treated NER as a sequence tagging problem, where each token has a corresponding label (entity-type). The approaches use either a word-level representation (each word corresponds to a word embedding, pre-trained on an extensive collection of text), a character-level representation (useful to exploit sub-word information such as suffix and prefix), or a hybrid approach [39]. Convolutional neural networks, recurrent neural networks, and most notably transformer architecture showed promising results in NER [39].

The problem of syntactic analysis extends NER and aims to identify linguistic relationships in sentences like verbs and nouns [17]. This leads to semantic analysis, which seeks to identify the semantic relationships between entities. For example, the previous sentence, "U.N. official Ekeus heads for Baghdad.", could be translated to the relationship: `travelsTo(Ekeus, Baghdad)`. Where `travelsTo` is a database relation of interest [17]. The complexity of these tasks stems from the diversity of patterns in written language [17]. For example, scientific reports and adventure stories are different in their styles. This richness in linguistics makes broad systems for information extractions a challenge [17].

Document Information Extraction extends the notion of information extraction on texts to documents [24]. This information can often be represented as key-value pairs, where each key (field of interest) has a value [24]. Form-like documents are common in everyday business interactions and often contain valuable data like invoice numbers and dates [47]. Document Information Extractions is the process of obtaining these key-value pairs.

The extraction becomes more challenging for visually rich documents as the layout gets more complex[80]. Such documents transmit the information not only in the text but also in visual appearance and relative positioning to other text. Document Information Extraction systems can therefore incorporate positional and visual information in addition to text to improve their effectiveness [85].

2.5 Understanding receipts

This thesis focuses on a specific type of document understanding, namely on receipts. Many receipts have a variety of information on them, for example, company name, telephone number, date, and the total amount. Line item information like unit price and quantity of each item is also present on receipts [75]. This information can be vital for office automation in financial and accounting areas [23].

Modern learning-based approaches require large and well-annotated datasets to be successful [23]. However, sensitive data like customer names, dates, and credit card information on receipts hinders large-scale compilation of training data [6]. The lack of big datasets creates a problem for training and evaluating models. Blanchard et al. [6] considered an automatic approach to generate custom invoices to combat the shortage of examples for learning-based approaches.

There are a few annotated medium-scale datasets of receipts. The International Conference on Document Analysis and Recognition (ICDAR) published a challenge for scanned receipt OCR and information extraction (SROIE) [23]. The challenge consists of three parts: Text Localization, OCR, and Key Information Extraction. This thesis focuses on the third part, the extraction of text for key fields from each receipt. The dataset consists of 1000 scanned receipt images with ground truth annotations. The receipts contain mainly English words and originate from Malaysia. Figure 5.1 illustrates an example from the SROIE dataset. Each receipt is represented by three documents in the dataset:

1. An image of the receipt
2. Transcription of the receipt
3. Text extraction for four entities (company, address, total, and date)

The challenge evaluated the performance of different systems based on the F1 score. This evaluation metric is used throughout the entire thesis. The F1 score corresponds to the harmonic mean of precision and recall. The SROIE challenge considers an exact match between the predicted text and the ground truth text as a true positive (TP). The number of extracted entities represents the predicted positives ($TP + FP$). The number of ground truth (gt) entities denotes the actual positives ($TP + FN$).

$$\begin{aligned} Precision &= \frac{TP}{TP + FP} = \frac{\# \text{ exact matches}}{\# \text{ predicted entities}} \\ Recall &= \frac{TP}{TP + FN} = \frac{\# \text{ exact matches}}{\# \text{ gt entities}} \\ F1 &= \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \end{aligned}$$

The CORD dataset [54] consists of over 11 000 Indonesian receipts obtained through crowd-sourcing. The first 1000 receipts were published in December 2019 with a picture of the receipt and ground truth word labels. In contrast to SROIE, CORD considers 30 different entity types, from the price and quantity of a unit to the total amount and amount paid by cash or card and various others. Due to privacy and legal issues, the authors went through an extensive process to blur sensitive information in the image and remove the corresponding fields in the ground-truth annotations. Hence, the resulting images have major parts, like the company name and location, blurred.

The WildReceipts dataset [75] is the result of a web search for receipt and invoices images. The authors manually removed unreadable, incomplete, and non-English receipts and annotated them with 25 entity types. The resulting 1740 receipt images often display receipts with folds from non-front views. This makes the WildReceipt dataset more challenging than scanned documents only. The annotations provided by the dataset contain errors because they are based on the extraction of an OCR engine.

CHAPTER 3

State of the Art

This chapter aims to explore and categorize different approaches for extracting information from receipts. The focus lies on approaches which were successful in the information extraction task of the SROIE challenge. It further includes other exciting techniques relevant for extracting information on receipts. Because of the similarities between invoices and receipts, systems focusing on information extraction on invoices are still relevant and included.

This thesis divides the systems into several categories to facilitate comparisons and highlight differences. Earlier approaches for the extraction of information considered the use of templates. Each template has most of the relevant fields, either in a fixed position or relative to a keyword. Identifying the template of an invoice helps to extract this information. Current approaches often incorporate advances in natural language processing (NLP) to boost their ability to understand receipts. These approaches adopt state-of-the-art systems for named entity recognition (NER). Grid-based solutions aim to exploit spatial information and apply convolutional neural networks from a variety of successful approaches in computer vision tasks. Graph-based systems encode the document as a graph and exploit information from neighboring nodes to extract important fields.

3.1 Template Based Approaches

Template-based approaches work on the premise that receipts or invoices from the same company share the same layout. The idea is to find an existing annotated receipt from this company and extract the information based on the position, text, and structure [70]. One obvious drawback is the necessity of an annotated receipt for each vendor. If this is not available, these approaches fail to generalize on new document layouts [53]. The use of example annotated documents introduces a different problem. Keeping an up-to-date template for each vendor is error-prone and not scaleable [43].

Schuster et al. [70] developed Intellix, a system based on this concept. Its goal is to identify annotated documents from the same template and use the position and content of these extracted entities to obtain the information on new invoices. An end-user monitors the extracted information and provides feedback if necessary.

The first step of Intellix is template detection, which aims to find similar documents. It uses the text search engine Lucene [Web6] to look through the annotated examples efficiently [14]. Lucene is based on a bag-of-words representation using TFIDF (term frequency x inverse document frequency) [27] and enables the search for the top k nearest neighbors. Because the same words alone do not translate to a similar template, some layout information is useful. The authors proposed a new feature type, named wordpos, to compare documents by their positional appearances of words. Intellix projects a 20x30 grid on top of the document, resulting in a column x and row y for each word (using the center of the bounding box). Wordpos encodes each word with `txt_x_y` and adds it to the representation of the document [14]. For example, if the text `Invoice` is in row 1 and column 15, `Invoice_15_1` is the resulting encoding used for searching the examples. Using Lucene with this representation enables the system to search for terms in specific positions. If receipts share many words in the same place (resulting in the same wordpos), they are probably from the same template. This search is prone to shifts in the document. Even a small shift of the document in any direction can change the column or row of several words, resulting in different wordpos. This, in turn, prevents Lucene from finding documents from the same template.

The second step is to use these examples from the same template and extract the information from the new invoice. Intellix applies three different methods for extraction followed by a decider [70]. The first method, Fix-Field Indexer, relies on the correct template detection. If all the documents similar to the new document (according to the wordpos representation) share the same company name, the original document probably stems from the same company. Some relevant fields like Date or Invoice Number are in a fixed place within a template. The Fixed-Position Indexer extracts those by using the absolute position within the document. The third method, Context-Based Indexer, takes relative positions from other words into account. For example, the total amount is often on the right of the word "Total". All these Indexers list their possible extractions with a confidence score, and the decider determines the extraction.

The third step is about incorporating human feedback into the system. Intellix achieves this by adding a document with the corrected extractions to its example pool if the obtained information was incorrect [18]. The system can find the corrected document in future searches and utilizes the user feedback to adapt its predictions on new documents.

This implementation heavily relies on the fact that a template for this document is already available [15]. This is not always the case, as invoices from new vendors can come in. The system does not perform well on unseen templates. The authors tested the ability of few-example learning on a private invoice dataset and showed an extraction effectivity of 78% F1 score with one document per template. If no template document was available, the F1 score was only 22%. The average effectivity of the system was 88% [15]. The poor performance on unseen documents shows that the system needs at least a few examples and human feedback to perform at its best.

A different template-based approach developed by Rusinol et al. [63] uses the same premise as Intellix by using templates for each vendor and creating a model for each relevant field. The system implements the same idea of the Context-Based-Indexer by giving each shared word between the templates and the new invoice a vote for where the entity should be. The proximity of the word to the target field determines the weight of the vote. The document frequency furthermore influences the weight. The more documents include one word, the less important the word is to extract the field of interest. Extended work [11] on this system experimented with using a-priori weights for the votes. The method more heavily emphasized words close and to the left or top of the target field. This adaption improves the performance when only a few samples are available due to favoring votes in the local context and discarding words with little relationship to the actual field [11].

3.2 BERT

BERT (Bidirectional Encoder Representation from Transformers) [13] is a deep learning approach achieving state-of-the-art results on a variety of natural language processing (NLP) tasks, including NER.

BERT works fundamentally different to template-based approaches. The main idea is to use unsupervised pre-training on a large corpus of texts to get a pre-trained language model (LM), which is able to improve many NLP tasks [13]. This pre-trained model can be fine-tuned to extract information from receipts using labeled data [85]. Extracting information from receipts can be tackled as a token classification problem. Every word in the text is either part of an entity or not. Serializing the text in the document and labeling each word with the corresponding class, enables the system to extract the relevant fields.

The BERT workflow for token classification problem [13]:

1. Break the input sequence into tokens using WordPiece [82].
2. Use BERT to create contextualized embeddings for each token
3. A linear classifier labels each token with a label.

The architecture for BERT stems from the Transformer encoder introduced by Vaswani et al. [76]. The Transformer encoder consists of a stack of encoder layers. Figure 3.1 depicts the architecture of one encoder layer. Each layer consists of self-attention and a fully connected feed-forward network. The authors proposed residual connections and layer normalization for each sublayer.

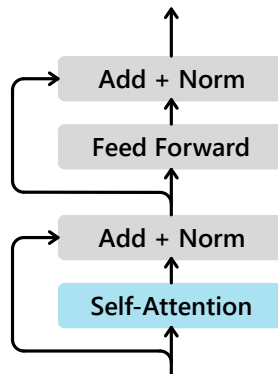


Figure 3.1: Architecture of one encoder layer in the Transformer encoder [76].

This architecture does not use recurrence or convolutions but relies entirely on the self-attention mechanism to model dependencies between inputs [76]. The attention mechanism proposed for the Transformer enables the model to learn dependencies and relationships between words and to attend to different words in the input sequence.

The attention function creates a context vector for each input. This vector entails information from each word in the input sequence through the weighted sum according to the attention score. Each position in the encoder can attend to all positions in the previous layer. Figure 3.2 illustrates the idea behind the Scaled-Dot-Product Attention.

For each position in the input, the self-attention mechanism calculates a Key (K), Value (V), and Query (Q) vector through multiplication with the learnable matrices M_K , M_V , and M_Q respectively. The Q_i and K_j vectors are used to create an attention score a_{ij} . The attention scores measure how much weight the corresponding V_j gets in the weighted sums over all values. This weighted sum is the resulting contextualized embedding vector for the i th input [76].

The following equation in matrix notation formalizes the Scaled-Dot-Product attention:

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

d_k refers to the number of dimensions of the key value. Dividing by the square root of d_k aims to counteract the growing magnitude of the dot-product with higher dimensions [76].

The Transformer (and therefore BERT) uses Multi-headed attention, which enables the model to attend to different pieces of information from different inputs. Multi-headed attention applies the attention mechanism multiple times on projections of Q , K , and V and concatenates the result [76].

Because the attention mechanism is unaware of the positional relationship between tokens, the Transformer architecture proposes using 1D-positional encodings to model the positional information and enable the model to be aware of the order and 1D position of tokens within the input sequence [76]. To avoid confusion with concepts introduced later, this thesis refers to the positional embeddings from the Transformer and BERT as 1D positional embeddings. The Transformer architecture combines the learnable embeddings for the text tokens with the 1D-positional embeddings to create the input for the encoder.

BERT extends the input for the encoder further by adding segment embeddings. The additional embedding facilitates new pre-training objectives and enables a variety of downstream tasks. These embeddings are not of interest for the token classification problem. Figure 3.3 illustrates the composition of BERT encoder input.

BERT uses the encoder from the Transformer to pre-train a capable language model for downstream tasks, such as named entity recognition. The authors trained the bidirectional encoder on unlabeled data with two training objectives [13]. The masked language model is a pre-training task where the data generator randomly masks some input tokens. The objective of the model is to recover the masked tokens according to the context and surrounding tokens. The Next Sentence Prediction task enables the model to understand the relationships between two sentences. The model aims to identify if sentence B does

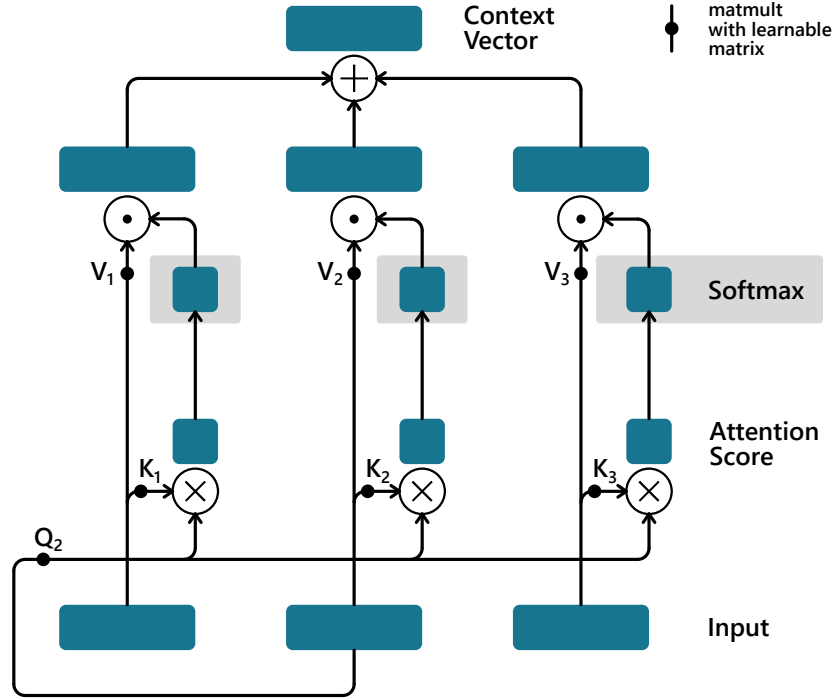


Figure 3.2: Illustration of the attention mechanism calculating the context vector of the second input [76].

appear after sentence A in the original text or if the data generator mixed them up. The successful deployment in multiple downstream tasks displays the effectiveness of the unsupervised pre-training for language understanding systems [13].

The pre-trained BERT needs task-specific fine-tuning to extract information from receipts. Applying BERT in this context requires the serialization of the text within the document. BERT can treat the problem as a token classification task to extract fields of interest from the text. A linear token-classifier can be used on top of the contextualized embeddings to fine-tune pre-trained BERT for extracting information from receipts. Fine-tuning and testing BERT on the SROIE dataset results in a promising 92% F1 score [85].

3.3 Language Models with Layout

Because BERT was successful in a variety of tasks and showed promising results for NER, researchers proposed different adaptations of the BERT architecture to improve the performance on document understanding tasks.

One major shortcoming of the original language model is the loss of 2D information. BERT only uses the serialized text and can not infer the location of a word in the document. The lack of 2D positional information makes it impossible for BERT to determine if two words share the same vertical or horizontal position, which is important for a structured document [85]. Different modifications try to incorporate the 2D positional information to enable the model to learn spatial relationships.

Another drawback of BERT is the lack of visual information encoded by the language model. Some adaptations try to include visual information to improve the performance on visually rich documents.

3.3.1 LayoutLM

LayoutLM [85] is a language model, which aims to improve the classic BERT-Model with layout information. The idea is to extend the powerful pre-training used by BERT to include layout information. The language model aims to learn language and layout information simultaneously.

BERT only relies on the text and 1D-position embeddings (at which position the word occurred in the serialized text of the document). LayoutLM extends this notion by adding embeddings for the 2D position to learn spatial relationships of words within documents. The document layout information captured with the 2D positional embeddings is important for most structured texts [85]. The goal of these embeddings is to enable the language model to learn from absolute positions and relative distances between words.

In LayoutLM each token gets a bounding box (x_0, y_0, x_1, y_1) from the corresponding word, where (x_0, y_0) represents the top left corner and the (x_1, y_1) to the lower right. The 2D positional embedding is the sum of four embeddings.

$$\text{2D positional embeddings} = E_x(x_0) + E_x(x_1) + E_y(y_0) + E_y(y_1)$$

The X and Y coordinates have a separate embedding table (E_x and E_y), which means x_0 and x_1 use the same table to look up the corresponding embeddings, and y_0, y_1 use the table for the Y coordinates. The input to the Transformer encoder is therefore the classical input for BERT (token-, segment and 1D positional embeddings) plus the 2D positional embeddings. Figure 3.3 compares the embeddings for LayoutLM with BERT and other Language models discussed later on.

The visual information extends the text and positional information and can be essential in downstream tasks to distinguish between fonts, color, and visual features. This information on a word level is helpful to determine how important a word or region is

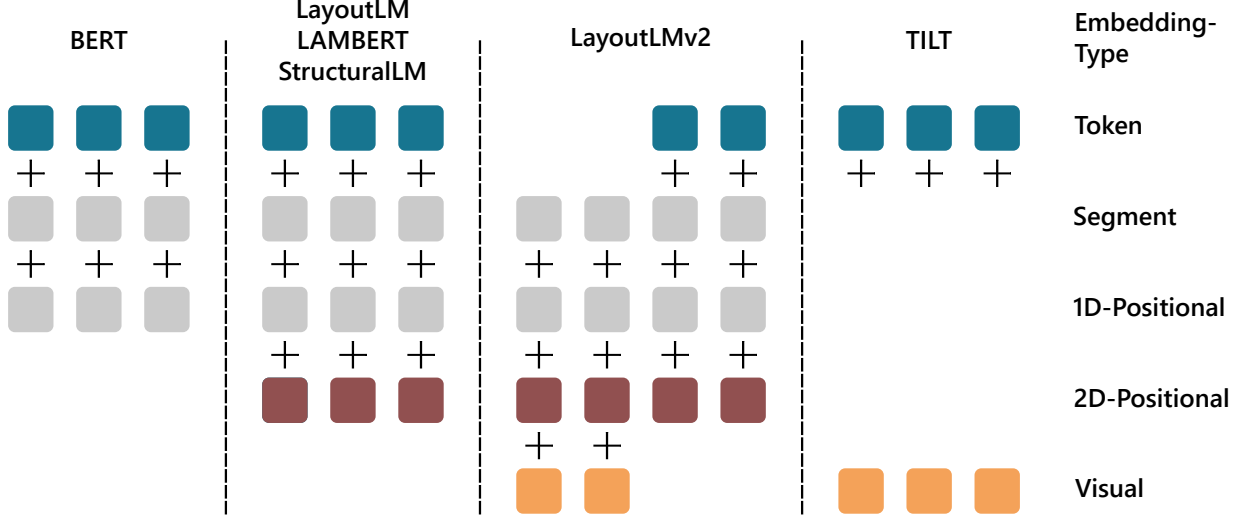


Figure 3.3: Comparison of the composition of the input for the encoder between different language models. BERT combines Token, Segment and 1D-Positional embeddings [13]. LayoutLM [85], LAMBERT [90] and StructuralLM [38] extend this with 2D-Positional embeddings. LayoutLMv2 [84] and TILT [57] incorporate Visual embeddings as well.

within the document [85]. LayoutLM incorporates the image embeddings by utilizing a Faster R-CNN model [61]. Using the bounding boxes for each word, the image of the receipt gets split into multiple sub-images, creating one image for each word. The result of the Faster R-CNN on these images represents the visual information for the word. The authors of LayoutLM proposed using the ResNet-101 model [21] as the visual backbone for the Faster R-CNN model to capture visual features. In contrast to the 2D positional embeddings, the visual embeddings are not added to the input of the encoder but rather used in addition to the output of the LayoutLM model. This creates a late fusion of visual information with the contextual embeddings produced by LayoutLM. In the case of token classification, the linear classifier can use the information of both embeddings to label each token. Figure 3.4 illustrates the setup for information extraction on receipt images. One disadvantage of this approach is the lack of visual information incorporated during the pre-training phase of LayoutLM [84].

LayoutLM is pre-trained on an unlabeled dataset with 11 million document images to learn textual and layout information (without visual embeddings). The authors used the IIT-CDIP Test Collection [37] as the training dataset and applied the Tesseract OCR engine to retrieve text and corresponding bounding boxes from the scanned documents.

The team proposed a new pre-training objective, the Masked Visual-language Model (MVLM), inspired by the popular NLP-Pre-training approach, masked language model.

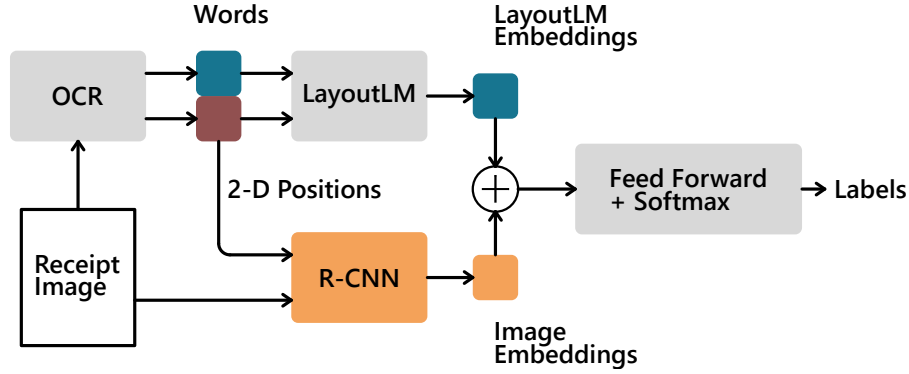


Figure 3.4: Illustration of LayoutLM for information extraction on receipt images with late fusion of visual embeddings. [85]

The original approach masked out some tokens in the sequence and tried to predict the missing token. The new MVLM approach works similarly by masking the text token but keeping the missing token’s bounding box. This way, the language model does not only learn about language contexts but also 2D positional relations.

The authors also used the Multi-label Document Classification loss during pre-training to improve the model’s knowledge about different layouts and document types. The Multi-label Document classification works by comparing the predicted document class from the language model with the document tag provided in the dataset.

To test the effectiveness of LayoutLM, the authors used three different document understanding benchmarks (FUNSD [25], RVL-CDIP [19], and SROIE Dataset [23]).

They experimented with different initializations of the Transformer encoder by using BERT or RoBERTa [44]. The experiments showed that using RoBERTa as the initial state improves performance slightly (approximately 2.5%) compared to BERT. LayoutLM outperformed the basic BERT and RoBERTa models in all three of these benchmarks without using the image embeddings. This improvement over the standard NLP models displays the necessity to include the layout when understanding scanned documents. In the FUNSD and RVL-CDIP datasets, adding the Image embeddings to the final layer improved the predictions.

The evaluation of LayoutLM on information extraction on receipts illustrates the need for 2D positional information. LayoutLM without image embeddings was able to achieve 0.95 F1 score on the SROIE dataset, a significant improvement over BERT with 0.92. The precision and recall on the SROIE dataset are only marginally better with image embeddings.

3.3.2 LayoutLMv2

LayoutLMv2 [84] is an extended version of LayoutLM. LayoutLMv2 is also based on the Transformer encoder and employs 2D positional embeddings on top of traditional 1D positional embeddings. There are three significant differences between LayoutLM and LayoutLMv2.

1. LayoutLMv2 incorporates visual embeddings into the encoder input and creating a multi-modal encoder.
2. It applies spatial aware self-attentions. The idea is to use 2D relative position biases added to the classical attention score.
3. LayoutLMv2 introduces two new pre-training objectives.

For the visual embeddings, the authors proposed the ResNeXt-FPN model [40] as the visual backbone for the encoder. To create the visual embeddings for the encoder, LayoutLM processes the whole image with the visual backbone. The output feature map of the visual encoder gets average-pooled to a size of 7x7 and split into 49 visual tokens. These visual tokens get concatenated to the input. Figure 3.3 illustrates that the textual and visual tokens are not present in the same input but rather split into two separate parts. Because the visual tokens stem from a 7x7 grid on the feature map, the positional embeddings correspond to the bounding-boxes of the cells in the grid. To help the encoder to differentiate between visual and text tokens, the visual tokens use a different segment embedding.

For spatial aware self-attention, LayoutLMv2 uses the classical attention score and adds a learnable relative position bias for the 1D and 2D positions [84]. Depending on the relative distance between two tokens, the attention score can change and therefore shift the attention to specific input tokens. The following equation calculates the new attention weights and figure 3.5 illustrates the use of the positional bias in the self-attention mechanism. The contextualized embeddings h are the weighted sum of the Values V according the the adapted attention weight a' .

$$\begin{aligned}\alpha_{ij} &= \frac{1}{\sqrt{d_k}} Q_i^T K_j \\ a'_{ij} &= \alpha_{ij} + b_{j-i}^{(1D)} + b_{x_j-x_i}^{(2D_x)} + b_{y_j-y_i}^{(2D_y)} \\ h &= \text{softmax}(a')V\end{aligned}$$

Where x_j and x_i are the top left corner coordinates of the i -th bounding box. The learnable embedding tables $b^{(1D)}$, $b^{(2D_x)}$, and $b^{(2D_y)}$ encode the bias in the attention score.

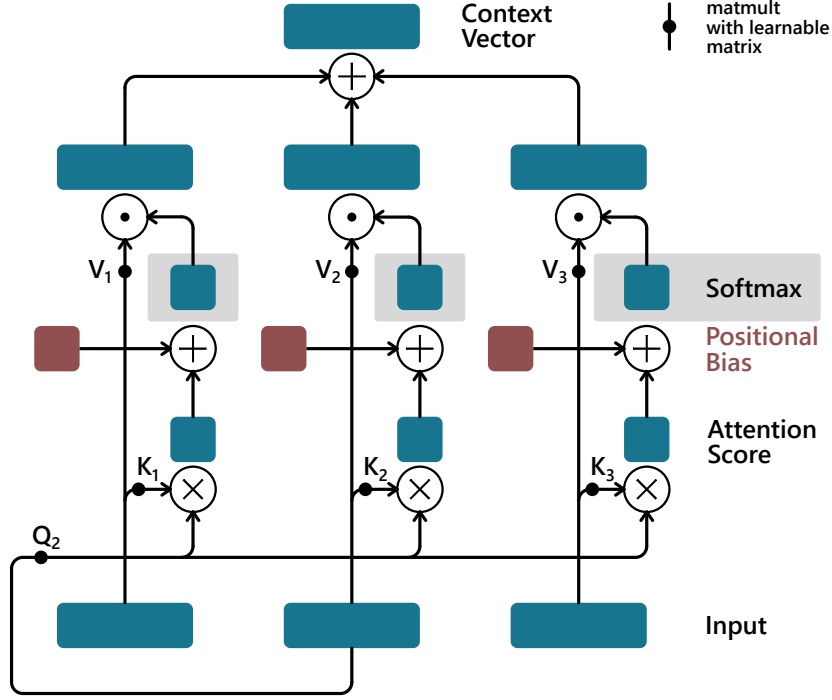


Figure 3.5: Illustration of the attention mechanism with Positional Bias added to the attention score [84].

The new pre-training objectives introduced by LayoutLMv2 aim to improve the connection between visual embeddings and textual and layout information. The pre-training task Text-Image Alignment covers some random text tokens in the image and uses a final layer on top of the encoder-output to predict for each token if the corresponding location in the image is covered. The Text-Image matching pre-training strategy randomly swaps the image with an image from another page. The model aims to predict if the image belongs to the document.

The authors showed that LayoutLMv2 could achieve an F1 score of nearly 98% on the SROIE dataset [84]. This increase compared to LayoutLM with 95% is significant, displaying the effect of the modifications.

3.3.3 LayoutXLM

LayoutXLM [86] is a multilingual extension of LayoutLMv2, intending to bridge language barriers found in cross-lingual benchmarks. The pre-training documents stem from 53 languages (with English, German and Japanese being the most common). Apart from some minor adjustments in the pre-training phase to accommodate the needs for multiple languages, LayoutXLM uses the same architecture as LayoutLMv2. The evaluation on

seven languages with 50 forms each in the test set showed that LayoutXLM performs significantly better than other bilingual models in semantic entity recognition. Fine-tuning on an dataset consisting of only English documents, resulted in a performance hit in every other language.

3.3.4 StructuralLM

StructuralLM [38] is a different extension from LayoutLM. It uses the same architecture but different 2D positional embeddings. Instead of creating a bounding box for each word, StructuralLM considers the detected cell (text segments) from OCR as a semantic unit. Every token in this cell gets the same 2D positional embeddings. The idea behind this change is to minimize the difference between words from the same cell.

Furthermore, it uses an additional pre-training strategy, Cell Position Classification (CPC). For the CPC pre-training objective, StructuralLM removes the 2D positional information from the input and lets the model predict the location of the cell within the document.

The authors did not evaluate their model on the SROIE dataset but showed promising results in form-understanding on the FUNSD dataset.

3.3.5 LAMBERT

LAMBERT [90] is an extension of LayoutLM, which uses spatial aware self-attention similar to LayoutLMv2. The authors modified the 2D position embeddings to use unlearnable embeddings. They used a combination of sin and cos of the coordinates, similar to the initially proposed positional embeddings in the transformer [76]. They introduced an adapter layer in front of the layout embeddings to soothe the encoder into accepting the 2D positional embeddings. The adapter layer starts with low values to avoid presenting the model with an input it was not trained on. The model achieved an F1 score of 98.2% on the SROIE dataset.

3.3.6 TILT

TILT [57] does not use BERT as the base architecture but rather extends the text-to-text transformer (T5) architecture [60]. This architecture enables the model to treat the information extraction task as a sequence generation instead of a token classification problem. Token classification architecture has a fundamental problem: they can not output something, which does not appear in the input. Generative approaches, such as the T5 architecture, do not suffer from this limitation. For example, this enables the model to correct common OCR errors. The T5 architecture extends the original Transformer architecture (including the decoder) by dropping the 1D positional embeddings at the input. They encode relative positional information by using spatial aware self-attention. TILT adapts this attention approach by incorporating 2D spatial biases during self-attention, similar to LayoutLMv2 [84]. Furthermore, the authors proposed using a U-net

architecture [62] with ROI pooling [61] to generate image embeddings for each input token (using the corresponding bounding boxes). These embeddings get added to the text embedding and fed into the transformer. This results in an early fusion of visual and textual information. For pre-training, they used more than 1M documents with a masked language model objective. To avoid overfitting during fine-tuning, the authors used augmentation on text (Case augmentation), position (spatial augmentation), and vision (affine vision augmentation). The authors showed the effectiveness of this approach with an F1 score of 98.1% on the SROIE dataset [57].

3.4 Grid Based Approaches

Unlike previous methods, which were designed primarily for text sequences with adaptations to incorporate layout and visual information, the following grid-based approaches do just the opposite. They use methods from computer vision problems and modify them to incorporate text information. The idea is to use a grid representation of the document, incorporating textual information and employing convolutional neural nets. The different grid representations explored in this section aim to retain 2D positional information.

3.4.1 Chargrid

Chargrid [28] aims to exploit fully convolutional neural networks to extract relevant information from documents. Instead of serializing the text within the document and creating a 1D representation, Chargrid transfers the characters to a grid, directly exploiting 2D information. The algorithm constructs the Chargrid by starting with a zero-initialized grid, the size of the image. For each character, the one-hot encoded representation gets assigned to all pixels within the bounding box. This representation retains spatial and textual information. It furthermore encodes some visual information like font size by giving more pixels the same encoding. Figure 3.6 illustrates the grid-creation for Chargrid and compares it to other approaches. The system scales this representation down to a standard size and creates a document encoding using fully convolutional neural networks.

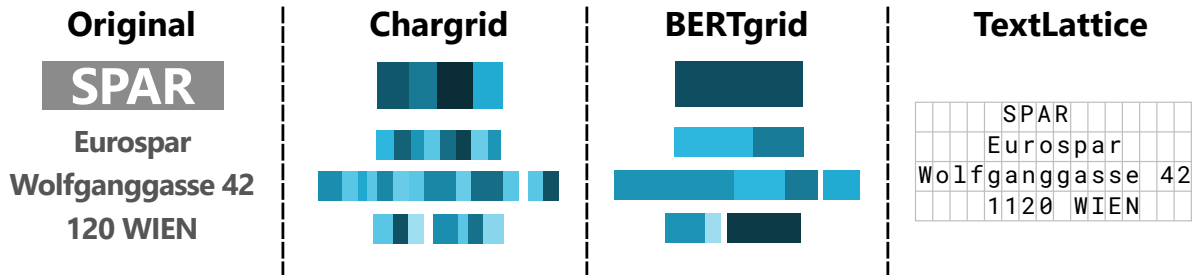


Figure 3.6: Illustration of Grid-creation by different approaches. Chargrid [28] works on character-level whereas BERTgrid [12] uses token-level embeddings. The system Tag, Copy and Predict [79] uses the TextLattice representation where each character occupies the same space.

The architecture of Chargrid is based on U-net [62]. The U-net architecture consists of two parts. The first part, the encoder, uses convolutions with max-pooling and downsampling. At each downsampling step, the number of feature channels is doubled. The decoder (the second part) uses upsampling with skip connections from the corresponding encoder layer and 1x1 convolutions [62]. The authors of Chargrid proposed the use of two decoders for training. The first is for semantic segmentation and classifies each cell in the Chargrid representation. The second decoder performs bounding-box regression.

To incorporate visual information into the model, the authors propose a second model, named Chargrid-hybrid, which uses two encoders, one for the Chargrid input and one for the image input. The decoders incorporate the skip connections from both the Chargrid and the visual encoder.

The authors trained and evaluated their model on a private 12k dataset of invoices with bounding-box annotations for each class. Their evaluation found that Chargrid performed well and significantly better if compared to an image-only version. The fusion of image and character encoding in the decoder did not increase the performance considerably. It is difficult to compare Chargrid results with other algorithms as the authors used a private dataset and a different evaluation metric based on the number of modifications, deletions, and additions required to obtain the ground truth [28].

3.4.2 BERTgrid

BERTgrid [12] extends the idea of Chargrid, but instead of using character encodings, it uses the language model BERT to create contextualized word embeddings. The text in the document gets serialized, resulting in a string representation of the document with no 2D position encoded. The text contains a 1D position regarding the reading order of the document. Applying BERT to the serialized text results in token-embeddings. These embeddings get projected onto the grid the same way as in Chargrid. But instead of having a different embedding for each character, characters within the same token have the same embedding. Figure 3.6 illustrates the difference between Chargrid and BERTgrid in terms of granularity in the encoding.

Wordgrid is an adaptation of BERTgrid where instead of using contextualized word embeddings based on the serialized text, the system uses word2vec [50] to get word-embeddings. Furthermore, they created two models which combine Chargrid with either BERTgrid or Wordgrid, respectively. For this fusion, they proposed an architecture concatenating the two representations after the first convolutional layer.

The authors custom trained BERT on 700k unlabeled data and evaluated the method using the same evaluation set as in Chargrid. The combination of BERTgrid and Chargrid outperformed the other models on nearly all entity types.

3.4.3 BERTgrid with visual information

ViBERTgrid [41] and VisualWordGrid [29] extend the idea of BERTgrid by adding visual embeddings to the encoder.

VisualWordGrid [29] explores two adaptations of BERTgrid. One model fuses the visual and textual information the same way as the proposed fusion in Chargrid. Two encoders work on the image and grid representation respectively and the decoder fuses the skip-connections from both encoders into a multi-modal representation. In contrast to Chargrid, VisualWordGrid uses word-embeddings instead of character-embeddings. The second approach fuses the visual and textual representation before encoding. Instead

of just concatenating the two representations, the authors proposed to only use the visual information if no text is present on the corresponding pixel. Because the authors evaluated their models on a private dataset, comparison to the other approaches is challenging.

ViBERTgrid [41] creates the same grid representation from the text as BERTgrid. The proposed network concatenates the BERTgrid representation to an intermediary layer of the U-Net architecture working on the image, resulting in the multi-modal backbone of ViBERTgrid. Instead of only performing pixel-wise segmentation, they proposed a word-classification branch. Using the bounding boxes for each word, they use ROI-Align [20] to generate a vector for each word from the multi-modal feature map. The system performs classification for each word on the combination of this feature vector and the corresponding contextualized BERT embedding. The authors demonstrated the performance with an F1-score of 96% on the SROIE dataset.

3.4.4 Tag, Copy or Predict

The authors of Tag, Copy or Predict [79] introduced a different way of creating a grid. Instead of each character overlaying multiple pixels, each character gets precisely one cell. They named this representation TextLattice and constructed it line by line with a heuristic about how many columns there are within the document. This representation avoids duplicated information. Figure 3.6 illustrates an example of this representation. Additionally to the semantic tagging, they introduced a Copy or Predict brach to correct OCR errors based on the expected (predicted) token. This branch uses a pointer generator network introduced by See et al. [71]. Evaluated on the SROIE dataset, the model achieved an F1 score of up to 0.97.

3.5 Graph Based Approaches

The reading order of business documents is often nonlinear in terms of just reading from top to bottom, left to right [7]. A graph representation of the document can describe the nonlinearity found in tables and structured layouts. In contrast to the previous models, graph based approaches attempt to model the relationship of the text segments directly. These approaches utilize graph neural networks, which are machine-learning-based methods that operate on these graphs [89]. To facilitate comparison between different graph based approaches, a short introduction of graph neural networks prefaces the different systems in the literature. The papers from Zhou et al. [89] and Wu et al. [83] give a comprehensive overview of current graph neural networks.

Several works in different research areas successfully deployed graph neural networks. From Physics (System Modeling) to Social Recommendation Systems and Image Classification [89], graph neural networks can exploit these problems' naturally occurring graph structure and deliver good results. According to a survey of graph neural networks [83], graph convolutional networks are helpful in node, edge, and graph classification problems. The idea of graph convolutions stems from the success of convolutional neural networks (CNN) in computer vision. A CNN can extract meaningful local features from image data in various parts of the image. They are shift-invariant and exploit local connectivity [83]. Graph convolutional methods extend the 2D convolutions onto graphs. The goal is to use adjacent nodes to aggregate information from the graph structure [89].

There are two commonly found categories for convolutional operators: the spectral methods (based on the graph Fourier transform) and the spatial method (based on a node's neighbors) [89].

Spectral-based methods are founded in graph signal processing. The eigenvalue decomposition on the graph Laplacian matrix enables the graph Fourier transform and the translation into the spectral domain [83]. Graph convolutional neural networks learn the weights of a filter that gets convoluted with the graph in the spectral domain. Extended work on these spectral CNN reduces the computational complexity associated with eigenvalue decomposition by using a Fourier transform approximation [83].

The spatial-based methods are rather diverse, but the Message Passing Neural Network (MPNN) framework generalizes some of these models [89]. The MPNN considers graph convolution as a message-passing process. The idea is to aggregate messages from each neighbor and then update the representation for each node [16].

The following three functions summarize the MPNN framework. Each node v is represented with a hidden state h_v^t at timestep t . The message passing phase accumulates the messages from all the neighboring nodes $w \in N(v)$ through the message function M_t . The next hidden state h_v^{t+1} of node v gets updated according to the previous state h_v^t and the messages from the current timestep m_v^{t+1} with the vertex update function U_t . These updates get performed for T time steps.

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw})$$

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1})$$

A readout phase concludes the MPNN network. The readout function R computes a vector for the entire graph based on the last hidden states h_v^T of each vertex.

$$\hat{y} = R(\{h_v^T \mid v \in G\})$$

The functions M_t , U_t and R are learnable differentiable functions. The authors showed that some of the most common spatial and spectral-based methods are instantiations of the message passing neural nets.

The graph attention network (GAT) [77] is a slight adaptation of MPNN, where the network incorporates the attention mechanism into the message curation step. It computes the subsequent representation of the node by attending to its neighbors through self-attention. The MPNN can represent the GAT with the following functions [83] where $W_{1...3}^t$ and a are learnable parameters:

$$\alpha(h_v, h_w) = \text{softmax}(\text{LeakyReLU}(a[W_1^t h_v \parallel W_2^t h_w]))$$

$$M_t(h_v^t, h_w^t, e_{vw}) = \alpha(h_v, h_w) W_3^t h_w^t$$

Another way of learning with graphs is to learn the graph structure itself. The work from Jiang et al. [26] explores an unsupervised approach to learn the optimal graph structure without relying on human established features. The idea is to utilize graph convolutional networks in applications where the graph data is not directly available. Graph learning aims to create the graph with a soft adjacency matrix representing the pairwise relationship between two nodes. The network aims to minimize an unsupervised loss function that encourages small relationships between nodes with a large distance $\|h_v - h_w\|$. The authors connected graph learning with a spectra-based graph convolutional operation and showed promising results on various datasets.

The following pages introduce and compare different graph based approaches to document understanding and information extraction on invoices and receipts. There are four main design decisions when applying a graph based systems on documents.

1. **Definition of a node:** A common practice is that each word or text segment corresponds to a node.
2. **Node representation:** Each node has a vector representation consisting of various information sources.
3. **Definition of an edge:** The system has to specify which nodes are connected.
4. **Convolutional Operation:** The process of extracting information from the graph structure.

3.5.1 An invoice reading system

Lohani et al. [45] proposed a graph neural network for information extraction on receipts. The approach treats each word in the document as a node. For each word, the system calculates a feature vector consisting of word-embeddings, boolean and numerical features.

To create the word-embeddings, the authors opted for Byte Pair Encoding (BPE) [72] over Word2Vec [50] because of the ability to deal with out-of-vocabulary words. BPE breaks the words into sub-words to encode the meaning of the word. The system uses the embeddings from BPEmb [22], which are pre-trained on Wikipedia. The boolean features contain information about the word, for example, if the word is a date, city, country, or just containing numerics. The numerical features entail spatial information to its nearest neighbors (left, right, top and bottom). The concatenation of these three types of features results in a 317-dimensional vector representing each word within the document.

The system creates the document-graph by adding an edge between nodes if they are spatial neighbors on the document in either the horizontal or vertical direction. Therefore, the resulting graph has at most four outgoing edges on any of the nodes (one in each direction). Figure 3.7 displays an example of such a graph.

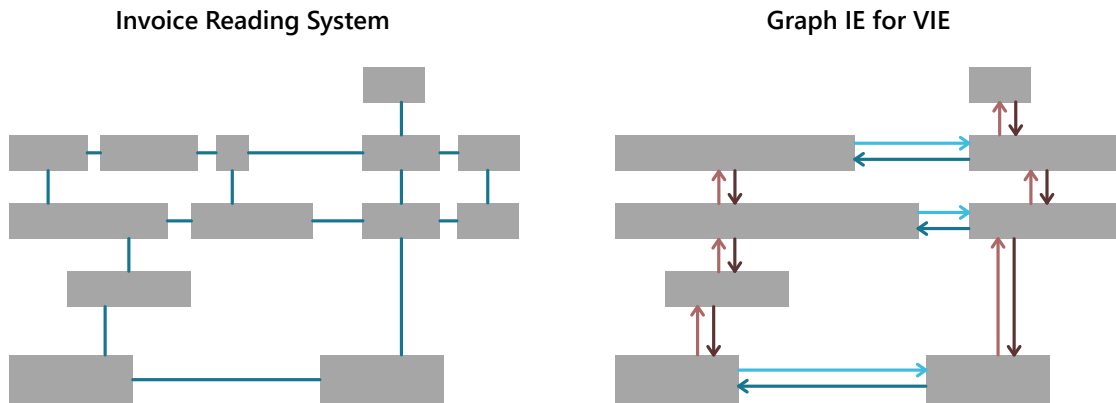


Figure 3.7: Comparing different approaches to model the graph. The invoice reading system [45] treats each word as a node and creates an undirected edge between horizontal and vertical neighbors. GraphIE [59] treats text segments as nodes and has directed edges between nodes. The edge from top to bottom is different from bottom to top.

The network then performs node classification using a spectral-based approximation for graph convolution followed by a softmax activation. The result is a class label for each word in the document. The system uses four layers of graph convolutions while doubling the number of filters in each layer.

The authors performed training and evaluation on a private dataset consisting of 3100 invoices. They could achieve an F1 score of 93% on 28 classes (from the company name to product quantity).

3.5.2 GraphIE

GraphIE [59] is a different approach using graph neural networks. Instead of using words as the semantic unit, GraphIE uses cells (text segments) extracted by the OCR engine. Each text segment gets represented by a node in the resulting graph of the document. The system applies BiLSTM with word-level embeddings and averages them over the text segments to obtain a node representation vector. Similar to the previous approach, they append spatial information (transformation of the bounding box) to the feature vector.

The system designs the graph using four types of edges. One for the right neighbor, one for the left neighbor, and one for upper and lower respectively. Figure 3.7 compares this approach to the word-level approach discussed earlier. This results in a directed graph with multiple edge types.

GraphIE uses the extended version of the MPNN framework for graph convolution presented by Schlichtkrull et al. [69]. The extended model allows for directed edges, enabling non-symmetric message functions depending on the direction. The aggregation function allows the system to use multiple edge types. For decoding the graph, they used BiLSTM+CRF [36] on each text segment to classify each word.

The authors did not evaluate their model on invoices or receipts but showed good results for visual information extraction on reports for drug-related side effects.

3.5.3 Graph Convolution for MIE

A network proposed by Liu et al. [43] explores graph attention networks on visually rich documents. Similar to GraphIE, this approach also considers text segments as the semantic unit. Each node in the resulting graph represents a text segment. Because they construct a fully connected graph, every node connects to every other node. Therefore, the proposed system does not fully utilize the potential of graph convolution [87]. The approach not only creates node embeddings but also uses edge embeddings based on the relative distance and aspect ratios between nodes. The system adapts the graph attention network (GAT) for graph convolutions. In contrast to the GAT network, this system creates the messages based on node-edge-node triples instead of the neighboring node alone. The attention mechanism scores the importance of each message for this node and creates a weighted sum representation of all the messages. This weighted sum updates the hidden state of the node. For classification, this approach uses BiLSTM+CRF [36] on the word embeddings from word2vec [50] in addition to the graph-embeddings. Overall, the authors showed that their approach yields an F1 score of 84% on an International Purchase Receipts dataset with 1500 documents.

3.5.4 PICK

Contrary to the previous graph-based approaches, PICK [87] does not rely on hand-crafted graphs and features. The system uses the graph learning approach discussed earlier, where the graph gets constructed using a soft-adjacency matrix. There is no need to specify relationships between nodes beforehand, as the system tries to learn the strength of a connection between nodes based on the node-embeddings. The idea is to catch relations between nodes even if they are not spatially close on the image itself.

PICK considers text-segments as the semantic unit, therefore, each node represents one text-segment. A document with N text-segments can be represented with s_1, \dots, s_N and b_1, \dots, b_N where s_i represents the i th text segment in the document with bounding box b_i .

PICK consists of four modules.

1. The encoder is responsible for creating text, image, and node-embeddings. This component contains the Transformer encoder and the CNN.
2. The graph-learning module creates a soft adjacency matrix from the node-embeddings.
3. The graph-convolution module is a stack of convolutional layers aggregating information from neighboring nodes.
4. The decoder uses the text-embeddings and the node-embeddings to classify each character.

For the text embeddings, the author proposed the use of a transformer encoder on the character level for each text segment. The text-embedding for the node is the average over the resulting contextualized character-embeddings. A convolutional neural network creates the image-embeddings for each node based on the sub-image according to the bounding box. The initial node representation v_i^0 is the sum of text-embeddings and visual embeddings.

Based on these node-embeddings, the graph learning module creates a soft adjacency matrix for the graph convolutions. The graph learning module is unaware of any spatial positioning as the node-representation does not encode 2D positions. The graph is constructed solely based on image and textual information.

PICK uses a variation of the MPNN convolutions to distribute information between nodes. The authors utilized edge embeddings to encode spatial relationships and aspect ratios between the two nodes. The directed edge embedding between node i and j :

$$\alpha_{ij}^0 = W_\alpha [x_{ij}, y_{ij}, \frac{w_i}{h_i}, \frac{h_j}{h_i}, \frac{w_j}{h_i}, \frac{T_j}{T_i}]^T$$

Where x_{ij} and y_{ij} are the horizontal and vertical distances between the nodes, w_i and h_i are the width and height of the bounding box and T_i represents the number of characters in the text-segment. W_α is a learnable weight matrix. These edge embeddings are the first encoding of 2D position in PICK, until now the model operates solely on the text and visual information.

For the l th convolution on the graph, PICK creates a hidden feature vector h_{ij}^l for each edge:

$$h_{ij}^l = \sigma(W_1^l v_i^l + W_2^l v_j^l + \alpha_{ij}^l + b^l)$$

The weight matrices W_1^l , W_2^l and b^l are learnable for each convolutional layer. ReLU is used for the activation function σ .

To create the resulting node embeddings from the convolutional layer, the system calculates the weighted sum of the hidden edge embeddings h_{ij} according to the soft adjacency matrix A from the graph learning module and a learnable weight matrix W_3^l .

$$v^{l+1} = \sigma(A_i h_i^l W_3^l)$$

Each convolutional layer also updates the edge embedding based on the hidden feature vector:

$$\alpha_{ij}^{l+1} = \sigma(W_4^l h_{ij})$$

For decoding, PICK uses a BiLSTM + CRF [36] on the character embeddings from the transformer encoder plus the corresponding node-embedding from the graph convolutions. So each character in the document gets a class-label, which is used to extract fields of interest from the document.

Figure 3.8 displays how PICK processes the receipts to extract information.

The authors evaluated their approach on a variety of datasets, including SROIE. They could achieve an F1 score of 0.96 for extracting the fields from the receipts.

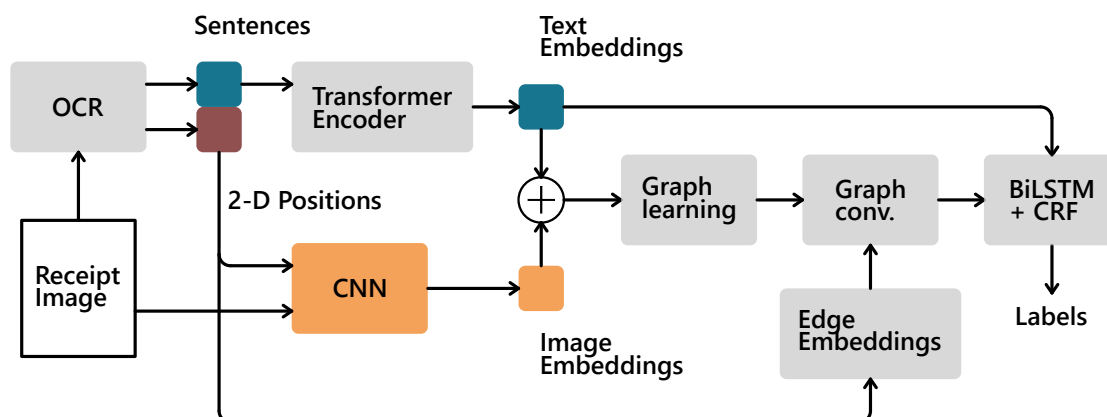


Figure 3.8: Illustration of PICK for information extraction on receipt images. [87]

3.6 Others

Some honorable approaches which do not fit any of the previous categories.

3.6.1 BiLSTM - RNN

CloudScan [53] is a system based on BiLSTM recurrent neural network (RNN). The system uses an OCR engine to convert invoices into text and creates N-grams of words on the same line. CloudScan computes various handcrafted features for each N-grams, for example, if the text matches a known country or zip code and where the text is in the line. The resulting feature vector gets passed through a BiLSTM network, classifying each N-gram into 32 fields of interest. The resulting extractions get post-processed to handle simple OCR errors and eliminate extractions that do not fit the syntax of the respected field (for example, the total amount needs to be a number). The authors trained the system on more than 300,000 invoices and achieved an F1 score of 0.89 and 0.84 on unseen templates.

Sage et al. [65] designed a similar system but used a less extensive feature vector (only adding spatial information from the bounding box and a category regarding the text structure). The system uses several stacked BiLSTM layers to classify the tokens without any extensive post-processing. Trained on 28,570 purchase orders, the system was able to identify the ID number and the quantity for each line item with an F1 score of 0.934 and 0.821 on unseen layouts.

3.6.2 End-to-End Systems

Contrary to the previous approaches, TRIE [88] does not perform text reading and information extraction separately but creates a unified end-to-end network able to extract the information in one. This approach avoids the following limitations. First, it uses both the visual and textual features extensively and creates powerful multimodal features. Secondly, text reading and information extraction are correlated, which gets exploited with the combined approach. The text reading and information extraction tasks reinforce each other. The authors proposed using a convolutional network to create a shared feature map for text reading (recognition) and information extraction. For information extraction, they used a BiLSTM on the fused multimodal context feature. They showed promising results with an F1 score of 0.82 on the SROIE dataset. Using the ground truth bounding boxes and transcriptions, the system achieved 0.96.

DocReader [30] also attempts to combine the OCR step with the information extraction on the document. It uses a convolutional neural network as an encoder with an attention-based LSTM network as the decoder. Their convolutional neural network gets initialized with a pre-trained OCR model and frozen in the first stage of training to enable the model to start with some idea about characters. The first phase focuses on training the information extraction task before the second stage allows the model to adapt the encoder and improve the OCR and information extraction in a combined manner. The authors trained their model on 1.5 million single-page invoices. They did not evaluate their model on a public dataset but showed it outperforms Chargrid.

VIES [78] is another approach to tackle information extraction on documents in an end-to-end manner. The system consists of three branches, text detection, text recognition, and information extraction. Feature fusion of the detection and recognition branch aims to provide visual and semantic information to the information extraction module. The authors pre-trained the text detection and recognition before jointly training with the information extraction module. The system achieved an F1 score of 91% on the SROIE dataset with the text-detection and recognition module but improved to 96% with the ground truth text.

3.7 Comparison

This chapter introduced a variety of different approaches found in the literature. Because the SROIE dataset is the most prominent public dataset for information extraction on receipts, this performance evaluation focuses on this dataset. Comparison across different datasets is difficult because of inconsistency in layout or language. Table 3.1 compares the discussed approaches according to the performance on the SROIE dataset (if available). It further illustrates the type of information used by the different approaches. Not every approach considers visual or layout information. Some of the approaches rely on the serialization of the text, which in turn requires that the reading order of the document is known. Because structured text and tables can have complex layouts, the reading order is nontrivial [41].

	Textual	2D-Position	Image	serialization	F1
Templated based					
Intellix [70]	x	x			-
Template 2 [11]	x	x			-
Language Models					
BERT [13]	x			x	0.920
LayoutLM [85]	x	x	x	x	0.952
LayoutLMv2 [84]	x	x	x	x	0.981
StructuralLM [38]	x	x	x	x	-
LAMBERT [90]	x	x	x	x	0.982
TILT [57]	x	x	x	x	0.981
Grid based					
Chargrid [28]	x	x	x		-
BERTgrid [12]	x	x		x	-
VisualWordGrid [29]	x	x	x	x	-
ViBERTgrid [41]	x	x	x	x	0.964
Tag, Copy Predict [79]	x	x	x		0.965
Graph based					
Invoice Reading System [45]	x	x			-
GraphIE [59]	x	x			-
Graph Convolution [69]	x	x		x	-
PICK [87]	x	x	x	x	0.961
Others					
Cloudscan [53]	x			x	-
TRIE [88]	x	x	x	x	0.962
DocReader [30]	x	x	x	x	-
VIES [78]	x	x	x	x	0.961

Table 3.1: Comparison between the different approaches regarding the incorporated information and the need of serialization. If an evaluation is available on the SROIE dataset, performance can be compared using the F1 score.

Austrian Receipts

In Austria, nearly every transaction of goods for money requires the company to issue a receipt. The Austrian government regulates the generation and contents of receipts mainly through the federal law *Bundesabgabenordnung (BAO)* and the decree *Registrierkassensicherheitsverordnung (RKSV)*. According to these laws, a company needs to create a receipt if it receives money in the form of cash or card payment in exchange for goods or services (with some minor exceptions) (§132a Abs. 1 BAO). These receipts need to contain at least: (§132a Abs. 3 BAO)

1. name of the company
2. consecutive number (Bonnummer), which identifies the business transaction
3. date of creation of the receipt
4. quantity of goods or service
5. payment amount

Businesses with an annual turnover of at least 15,000 euros or more must record all cash receipts and cash payments individually using a cash register (Registrierkasse) with some exceptions (§131b Abs. 1 BAO). The receipts from these cash registers have additional requirements to increase security against manipulation.

They need to include a machine-readable Code (OCR-, Bar-, or QR-Code) with the following information (§10 Abs. 2 RKSv):

1. identification number for cash register
2. Bonnummer, which identifies the business transaction
3. date and time of creation of the receipt
4. total amounts for each tax rate
5. turnover
6. linking and signature value to increase manipulation security

Because this data is machine-readable, it is accessible to information extraction systems. While some of the data is encrypted, like turnover, other information like date of creation and total amounts for each tax rate is not. This information can be extracted by automated systems, as the encoding has a fixed structure. This thesis omits the machine-readable code, as the focus does not lie on creating the best extractions but on determining how well extraction systems can transfer the knowledge from English receipts to German receipts.

One thing that these laws do not cover is the layout and design of receipts. Therefore different companies use distinct layouts and styles, which makes it hard to define a typical receipt. Different Registrierkassen manufacturers have different templates, which in turn are customizable for the company, resulting in a wide variety of layouts [Aul GmbH]. Figure 4.2 displays a few examples of different layouts. One obvious difference between the displayed receipts is the height. Receipts don't have a fixed width and height and especially the height can differ between companies or even between receipts from the same company. If more items are bought, the list of line items gets longer and therefore the receipt.

4.1 Dataset

In the course of this work, a dataset of Austrian cash register receipts was collected. The Austrian Receipts dataset consists of 120 receipts collected through personal shopping. The dataset shares a similar structure to the SROIE dataset. In addition to the scanned receipt image, the ground truth extractions for major entities (company, address, total, date) are available. Analog to the SROIE dataset, ground truth transcriptions of the receipts are obtainable to focus on information extraction rather than text extractions.

The goal of this dataset is to facilitate the evaluation of models trained on English receipts. Xu et al. [86] did something similar in their work, where they evaluate a form understanding model on various languages. The test sets in their evaluation contain 50 forms for each language, which is comparable to the 120 receipts in this dataset.

4.1.1 Dataset Distribution

Since the receipts are from personal shopping, the variety is limited. One goal in the collection was to increase the number of layouts and vendors to better represent the different receipts in Austria. The dataset contains receipts from 66 companies with no more than 18 per company. Figure 4.1 illustrates the number of vendors over the number of contributing receipts. Most of the companies have at most three documents. The one outlier with 18 receipts is a big grocery chain in Austria. The maximum amount of documents per company was capped at 20. While many receipts from one company may help evaluate the performance on this template, it might skew the evaluation regarding the overall performance on Austrian receipts. The creation date of the receipts in this dataset is in the years 2020 and 2021. Most receipts were issued between May and August of 2021.

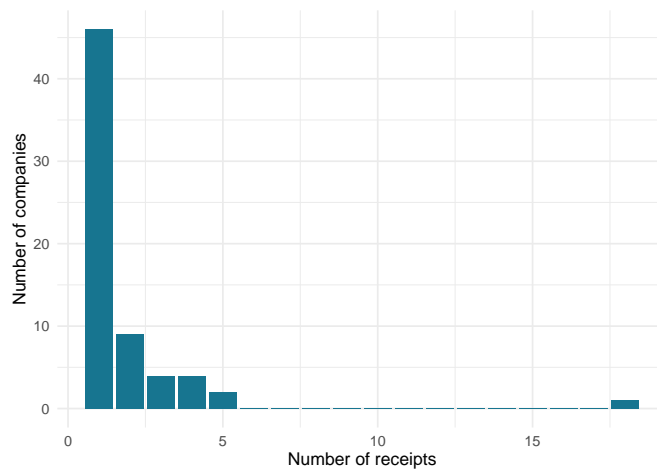


Figure 4.1: Number of companies over the number of receipts in the Austrian Receipt dataset.

4.1.2 Annotations

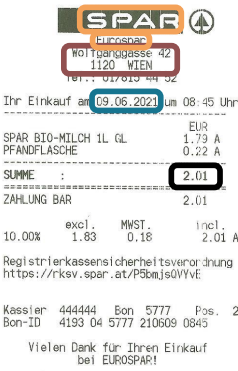
The evaluation of different models on the Austrian Receipt dataset requires ground truth annotations. The annotations (entities and transcription) were created manually to avoid OCR errors and provide a solid starting point for information extraction systems.

For the ground truth annotations of the four entities of interest (company, address, date, and total), the tool LabelStudio [Web7] proved to be useful as it enables the annotation of bounding boxes with labels and transcriptions. The advantage of the bounding boxes comes with training on Austrian receipts, as the location of the ground truth entities makes it easier to create token-level annotations. The section 5.1 elaborates potential pitfalls without positional information for the ground truth extractions. For this thesis, the bounding boxes are not necessary because the Austrian receipts are not used for training, but they could be beneficial in future work.

The company name and the total amount of receipts are sometimes ambiguous. For example, the company *BIPA Parfumerien Gesellschaft m.b.H.* also displays the common shorthand version *BIPA* at the top of their receipts. The scanned receipt image in Figure 4.2 shows that both versions occur in different places on the receipts. The shorthand version does not contain any uncertainty regarding the company, therefore justifying the inclusion of both versions in the ground truth annotation.

The total amount should be unique and clear on each receipt, but discounts, taxes, and can-deposits sometimes introduce ambiguity. If there is uncertainty, the ground truth annotation corresponds to the total amount encoded in the machine-readable code of the receipts.

Analog to the SROIE dataset, ground truth transcriptions of the receipts are available to focus on information extraction rather than text extractions. The tool PPOCRLabel [Web4] eases the transcription of the receipts. It can apply PaddlePaddle OCR and enables the manual correction of the results from there, which shortens the time spent on transcription.



SPAR
Hütteldorfgasse 42
1120 WIEN
Ihr Einkauf am 09.06.2021 um 08:45 Uhr


	EUR
SPAR BIO-MILCH 1L GL.	1.79 A
PFANDFLASCHE	0.22 A
SUMME	2.01
ZAHLUNG BAR	2.01

10.00% excl. MWST. incl. 2.01 A
1.83 0.18

Registrierkassensicherheitsverordnung
https://rksv.spar.at/P3bmjs0VYVE

Kassier: 444444 Bon: 5777 Pos: 2
Bon-ID: 4193 04 5777 210609 0845

Vielen Dank für Ihren Einkauf
bei EUROSPAR!



Billa AG
Hütteldorfgasse 26
1120 WIEN
Datum: 29.05.2021 Zeit: 11:22


	EUR
Innocent Smoothie C	1.99
SUMME	1.99
Gegeben Bar	5.00
Restgeld	3.01

Betrag dankend erhalten

C: 20% MwSt von 1.66 = 0.33

Vielen Dank für Ihren Einkauf!
www.billa.at

Filiale: 01296 Kassa: 1 Bon-Nr: 6356
Pos: 1 Kassier: Fr. Alexandra /9
Re-Nr: 1296-20210529-01-8356



Müller
Hütteldorfgasse / Hütteldorfgasse, 1120 WIEN

Stk	Artikel	Preis	Rabatt	Summe
1	VALERIE TAFELLOEFFE	2,99		2,99a
1	Zwischensumme	0,00		2,99
ZU BEZAHLEN				2,99
	Bargeld	EUR		2,99
	Rückgeld	EUR		-17,01
	MwSt %	Netto	MwSt. Betr.	Brutto
a:	20,00	2,49	0,50	2,99

Vielen Dank für Ihren Einkauf!

Kundenservice: 143 5 912 00 00 www.mueller.at

Umtausch (gegen Ware oder Gutschein) von
original verpackter Ware mit Kassensbon.

Kassen-Id: 5368,1962
Beleg-Id: 200000040201962000001622300597809

MwSt. Nr: 01049904402
F4020196201622300597809

29.05.2021 7:05

Lust wieder deine Haare zu stylen?

20 %
auf alle Styling Produkte von
syoss

Vor alle Kassas des Einkaufszentrums: Keine Namenskennung! (Gepäck) nicht mit anderen
Abrechnungen kombinieren!

Via Salina
IHR HOTEL AM HALDENSEE / TANNHEIMER TAL


**Entspannungstage
am Haldensee**

4 Tage/3 Nächte mit Genussspeisung
im eleganten Landhauszimmer
Nutzung von Pool & Sauna
mit Frühstücksgeld
1 Flasche House Sekt bei Anreise
1 Wellness-Entspannungsbad

Gültig auf Anfrage und Verfügbarkeit bis 30.6.2021
Weil es kein Urlaub ohne ein bisschen Spaß sein kann!

ab € 449,-
pro Person
inkl. direktem
Seeblick

Tel: 0043 5405 20 10 40 www.via-salina.at



BIPA

Datum: 07.06.2021 Zeit: 17:12

	C	
FA DUSCHE	C	1.80 T
NIVEA CREMEDUSCHE	C	1.79
MENTADENT ZAHNBÜRSTE	C	2.19
COLGATE ZAHNCREME	C	1.99
Zwischensumme	EUR	7.77
25% Nivea Men	25%	-0.45
SUMME	EUR	7.32
Gegeben Bar	EUR	7.50
Restgeld	EUR	0.18

Betrag dankend erhalten

Umtausch nur mit Kassabon innerhalb
von 14 Tagen. Ausgenommen Baby-
nahrung, Lebensmittel, Kapseln
und Medizinprodukte.

C: 20% MwSt von 6.10 = 1.22

T=TIEFPREIS, NICHT KOMBINIERBAR.1

**BIPA Parfümerien
Gesellschaft m.b.H.**
1120 WIEN
NEIDLINGER HAUPTSTRASSE 78-80
050013/01263
UID-Nr: ATU 19434404
www.bipa.at

Filiale: 01263 Kassa: 2 Bon-Nr: 753
Pos: 4 Kassier: SUZANA R /4
Re-Nr: 1263-20210607-02-0753

jö **jö schau, ganz
schön schlau.**

SIE HÄTTEN HEUTE GESAMMELT: 7 ÖS
JETZT UNTER JÖ-CLUB.AT MITGLIED WERDEN

DATE

ADDRESS

COMPANY

TOTAL

Figure 4.2: Example receipts in the Austrian Receipt dataset.

Evaluation

This chapter focuses on the evaluation of state-of-the-art approaches on Austrian receipts. The goal was to determine the effectiveness of BERT, LayoutLM and PICK in terms of precision and recall. Due to the limited data on Austrian receipts, the approach adopted in this paper trains the models on the SROIE dataset and evaluates on their Austrian counterparts. This introduces some limitations as the receipts in the SROIE dataset are in English and follow a different layout. Nonetheless, the performance of models trained on different receipts indicates how well it adapts to unseen layouts and the ability to transfer the knowledge to a new language. This approach is an adaptation to the research by Xu et al.[85], where they explore the performance of models trained in English and evaluated on a multi-lingual form-understanding dataset.

The SROIE [23] dataset has various advantages over CORD [54] and WildReceipts [75]. It is well annotated with minimal errors. Major fields of interest (company, address, date, and total) are available and not blurred and removed. It has high-quality scans of the receipts by minimizing folds and wrinkles and therefore enabling the focus on information extraction. It has ground truth transcription of each receipt, eliminating the need for error-prone OCR systems.

After training the models on the SROIE dataset, an evaluation on the Austrian Receipts dataset should hint at their ability to transfer the knowledge from English to German receipts. The goal is to identify problems and weaknesses in this approach of learning. The SROIE dataset has a training and test split with ground truth annotations for both. For this thesis, the training utilizes the whole training set and uses the SROIE test set as the validation set.

Due to the nature of this thesis, there are limited computational resources for training and evaluation. Google Colaboratory [Web8] is used for executing computational expensive code, as it provides a free environment with GPU support to execute and share python code. The limited execution time and GPU power can still constrain the training of models.

Many systems mentioned in chapter 3 require pre-training on millions of documents. These systems aim to generate an understanding of general document structure through the extensive pre-training process. Because of the large pre-training datasets, training can take up to multiple days on multiple GPUs [85]. The lack of dedicated GPU clusters for training in this thesis makes it infeasible to pre-train models. This only leaves two types of systems. Either there exists a public pre-trained version of the model, or the model does not require massive pre-training to perform competitively. One goal during model selection was to include models from different categories to provide a broader coverage of current approaches. Furthermore, the systems need to perform competitively on the SROIE dataset.

The choice fell on BERT [13], LayoutLM [85], and PICK [87]. All of them show competitive results on the SROIE dataset with an F1 score between 0.92 and 0.98. Additionally, the authors of these approaches provide implementations in PyTorch [55], simplifying the replication of their models and results.

BERT [13] is a strong baseline working on textual information only. The publicly available pre-trained models of BERT are ready for finetuning and achieve state-of-the-art performances on multiple natural language understanding tasks.

LayoutLM [85] is an extension of BERT incorporating 2D layout information into the transformer encoder. It is one of the few language models introduced in section 3.3, which has a pre-trained model publicly available. The performances of this model and the various extensions and adaptations display the importance of LayoutLM for information extraction on receipts. This thesis uses the LayoutLM version without image embeddings, as the performance did not significantly improve with image information according to the original paper [85].

PICK [87] has a different underlying architecture and works with graph learning and graph convolutional neural networks. This system was able to achieve a state-of-the-art result without relying on pre-training or extensive pre and post-processing. In contrast to the other two selected algorithms, PICK extends the attention mechanism to incorporate the idea of representing the document with a graph.

5.1 SROIE Preprocessing

There are some pre-processing steps necessary to use the SROIE dataset for training. The dataset has three files for each receipt: the image, the transcriptions with bounding boxes, and the text extractions. The receipt images are scans and as a result well lit and without any wrinkles. This gives the best possible starting point for OCR and information extraction, no need to pre-process those. The transcription contains text segments with the corresponding bounding boxes. The format of the transcription-file is the following for each text segment: $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4, txt$. The first eight values represent the vertices of the bounding box and the last contains the text. For information extraction, the SROIE dataset contains the ground truth text for the company, address, total, and date in JSON format. Figure 5.1 illustrates an example from the dataset.

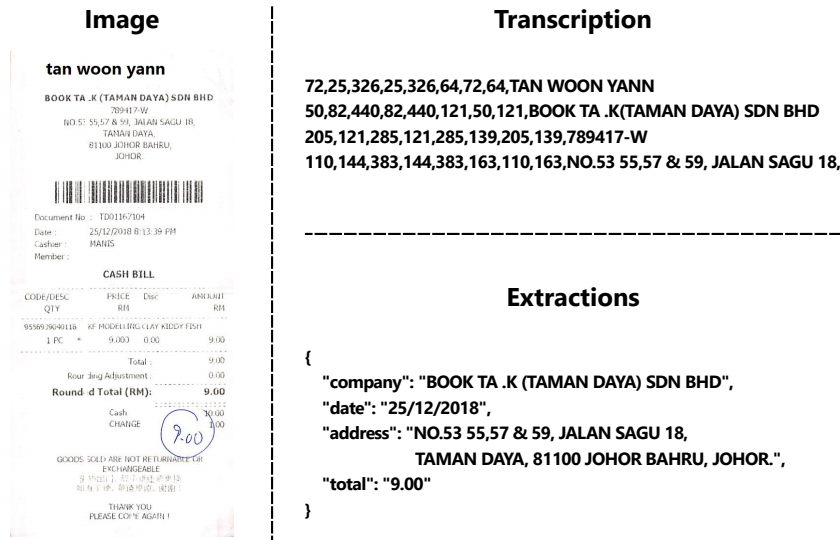


Figure 5.1: An example from the SROIE dataset [23].

All the selected models in this evaluation treat the problem as a token-classification problem. This means, the systems label each token in the document as either being part of an entity or not. Such models require token-level ground truth labels for training. The transcription provided with the SROIE dataset contains all tokens/words but without any label information. The pre-processing therefore involves labeling each word in the transcription with either COMPANY, ADDRESS, TOTAL, DATE, or O (outside). This means matching the transcriptions to the extractions based on the text. For example, in figure 5.1 the text segment with the text *25/12/2018* would get the label *DATE*, because it matches the extraction of this entity.

There are a few challenges while assigning these labels.

1. The text of entities can span multiple text segments.
2. A text segment can contain more than just the text of an entity.
3. The text of an entity can occur multiple times within the receipt.
4. The SROIE dataset contains OCR mismatches between the transcription and the extraction.

There is limited information available on how the different authors tackle this problem to train and evaluate their models on the SROIE dataset. Only the team behind PICK [87] mentions that they manually assigned a label to each bounding box. Annotating or correcting all extractions and transcriptions in the SROIE dataset, which consists of 1000 documents, is time-consuming and was therefore not used for this thesis. The Pre-Processing code used in this thesis is available in the appendix.

To combat the first two challenges, the pre-processing used in this thesis assigns labels if either the extraction fully contains the text of the segment or if the text segment contains the whole extraction. The second case sometimes requires splitting the text segments into multiple parts, so only the text of the entity gets labeled with the corresponding class (the bounding box gets split assuming equal width characters).

The third point is hard to handle, as there is no information about which text segment is the correct one. For example, the text of the entity *TOTAL* on a receipt is *9.00*, but on receipts with only one line-item, this amount can appear multiple times, making it difficult to assign the label *TOTAL* to only one text segment. The pre-processing in this work assigns the labels to all those text segments.

The OCR mismatches in the SROIE dataset represent a challenge in the pre-processing step. The mismatches between the transcription and the text of the entities make the exact matching unsuitable. The two most common mismatches are with whitespaces and punctuation. Either the extraction or the transcription often misses a blank character or differs in punctuation. The example in figure 5.1 has an OCR mismatch in the company text. The transcription is missing a blank character compared to the extraction. The pre-processing of the SROIE dataset used in this work consists of matching the entities to the text segments based on matches ignoring whitespaces. The address field is most prone to punctuation errors and therefore gets matched without them. This compromise ensures a higher rate of identifying the correct text segments without falsely assigning labels. Because some of the text segments deviate more than by whitespace and punctuation from the entity text, the remaining OCR errors were fixed by hand if possible. The most recent papers like TILT [57] perform their experiments on a test set with corrected OCR mismatches and modified ground truth entities. Some of the ground truth total extractions in the SROIE test set include the text *RM* while the training data only

considers the number (without RM) as the total extraction. This thesis only corrects the OCR mismatches and ignores discrepancies in the total entity.

Throughout the training of all three models, the token labels follow the IOB2 tagging scheme [68] as used in PICK and the implementation of LayoutLM and LayoutLMv2. This labeling style indicates the beginning of an entity with the tag 'B-' and uses the 'I-' tag for every other token in the entity. Tokens outside of any entity have the label 'O'. The Table 5.1 illustrates IOB2 tagging on receipt text.

Words	Eurospar	Wolfganggasse	42	Einkauf	am	09.06.2021
IOB2	B-COMPANY	B-ADDRESS	I-ADDRESS	O	O	B-DATE

Table 5.1: Example of IOB2 tagging [68]. The 'B-' tag indicates the beginning and the 'I-' tag the inside of an entity. A word outside any entity is marked with 'O'.

5.2 Implementation

The implementation and evaluation run on the Google Colaboratory [Web8] environment with python and the PyTorch library [55]. PyTorch provides a simple and efficient way to implement machine learning code with GPU acceleration [55].

5.2.1 BERT and LayoutLM

For BERT and LayoutLM, the implementation follows the code published in the huggingface transformer library [81]. The Transformer library supports the distribution of pre-trained models. BERT and LayoutLM are available in this library, eliminating the need to pre-train the models. Furthermore, the library provides tools for tokenization and other common problems in NLP. Throughout this evaluation, the base version of BERT and LayoutLM is used instead of the large variant, as it is comparable in performance on the SROIE dataset (94.6% instead of 95.2%) but with a third of the parameters. The difference in the base versus the large model lies in the number of encoder layers (12 instead of 24) and the hidden size (768 instead of 1024).

This thesis fine-tunes BERT and LayoutLM according to the code published by the team behind LayoutLM with some minor adjustments. The tokenizer provided in the Transformer library converts the text sequence into the input sequence for the model. The resulting input sequence gets padded to a length of 512 tokens. If a receipt creates more than 512 tokens, the original code splits them into multiple inputs without overlap. This thesis discards the excess tokens because only four receipts in the SROIE and three in the Austrian Receipt dataset result in more than 512 tokens. The labels from the original text sequence get aligned to the token sequence by giving the first token of each word the corresponding label. If a word results in multiple tokens, all tokens except the first one receive a padding label and are therefore ignored during fine-tuning. This procedure follows the implementation of LayoutLM.

The training process differs only slightly from the code provided. The memory of the GPU in Google Colaboratory limits the batch size to 8 instead of 16. The lack of improvement on the validation set did not warrant training for more than 50 epochs. The original paper suggested training for 100 epochs on the SROIE training set.

To label each token according to the model's output, the implementation uses the class with the highest logit. Neither the paper nor the code of LayoutLM specifies how it handles multiple extractions per entity type to create the final extractions used for evaluation. The final prediction in this implementation corresponds to the extraction with the highest mean score over all words in the extraction (according to the logits).

5.2.2 PICK

For PICK this thesis makes use of the implementation published by the authors [Web3]. This published model deviates from the paper in two ways.

1. **Image Embeddings:** The paper proposed sub-images for each node/text segment and a CNN to create the image embeddings. The PICK implementation creates the embeddings with the application of the CNN on the whole image with RoI-Align [20] to get the embeddings for each segment. The use of CNN+RoI-Align on the whole image results in contextualized image embeddings, which are based not only on the image within the bounding box but also on the context in which the sub-image appeared.
2. **Fusion of textual and visual information:** The paper suggested the fusion of textual and image embeddings after the transformer encoder. The implementation fused the textual embedding with the image embeddings before processing it with the encoder.

In addition to the changes proposed by the authors in their public implementation, this thesis incorporates three more changes:

1. Minor correction calculating the relational features between two nodes/text segments.
2. Adaptation of the character embeddings for Latin characters.
3. Restriction of transitions in the CRF layer according to the IOB2 tagging scheme. For example, the transition from the label *O* to *I-COMPANY* is not valid, as the beginning of each entity is marked with the *B-* tag.

To fine-tune PICK, each text segment gets translated to a token sequence by encoding each character and padding the maximum text sequence length. The relational features encode the relative positional information between two text segments.

The training process extends 50 instead of 30 epochs, as the performance increased significantly on the validation set.

The CRF layer at the end of PICK enables the Viterbi algorithm to obtain the most likely tag sequence [35]. According to this tag sequence, possible extractions are identified according to the IOB2 tagging scheme. In the case of multiple predictions for one entity type, the text which appears most often (or the first one if tied) is used as the final prediction. Employing the most likely tag sequence does not allow for prediction scores like in BERT or LayoutLM. The original paper and code do not state how they tackle this problem.

5.3 Results

This section reports on the results of the three models. The metrics used throughout this evaluation are the same as in the SROIE challenge discussed in section 2.5.

Table 5.2 compares the performance of the models as stated in their papers versus the performance achieved in this implementation and training. The F1 score for BERT on the SROIE dataset stems from the team behind LayoutLM [85], as the original paper [13] did not evaluate on this dataset. These results are not directly comparable as the original papers did not manually fix the OCR mismatches in the test set, whereas this thesis evaluates on a manually corrected dataset. Many of the most recent approaches, like TILT [57] and Lambert [90], use a corrected version of the SROIE dataset. Nevertheless, the performance of the implementation for this thesis is close to the performance in the corresponding papers. Only PICK is unable to achieve the same level as the original paper even with the corrected test set and 20 epochs more training.

The results in table 5.3 show the performance of BERT on the SROIE and the Austrian Receipt dataset. The results correspond to the evaluation using ground truth transcriptions of the receipts to avoid OCR errors. The micro average refers to the recalculated scores on the aggregated contributions across all entity types [48]. BERT performs significantly worse on the Austrian Receipts dataset in all entity types. The recall $\frac{\# \text{ exact matches}}{\# \text{ gt entities}}$ is a lot lower on the Austrian receipts. Only the precision $\frac{\# \text{ exact matches}}{\# \text{ predicted entities}}$ of the date and total entity are above 0.8. The discrepancy in precision and recall for the date and total entity states that BERT does not predict these entity types in some receipts. The high precision indicates, that the prediction the model makes, are often correct (more than 85% on the Austrian Receipts dataset).

Table 5.4 displays the performance of LayoutLM. The results are considerably better than BERT on the Austrian Receipt dataset, but overall performance is not close to the performance on the SROIE dataset. LayoutLM is perfect for the date entity in the SROIE dataset and correctly extracts more than 90% of the dates in the Austrian receipts. The model does not predict the total entity in half of the receipts, but the predictions it produces are correct 90% of the time in the test set.

The PICK model is unable to detect most of the fields of interest in the Austrian Receipts dataset, as displayed in table 5.5. The results on the company entity are similar to the results of the LayoutLM model, but the other entities are significantly worse. The date entity was never correctly extracted by the PICK model.

An additional experiment evaluates the performance of these models on receipts with OCR extracted transcriptions instead of ground truth (gt) transcriptions. Table 5.6 compares the performance of BERT, LayoutLM, and PICK employing the F1 score. This evaluation uses the OCR engine EasyOCR [Web2] to extract the text on the receipts. This OCR engine inevitably introduces errors in the transcription which makes exact matches between ground truth extraction and the predicted text impossible. All models perform worse with the OCR extracted transcription on the SROIE and the Austrian

Receipts dataset. The F1 scores are halved compared to the evaluation with ground truth transcriptions.

	original paper			this thesis		
	Precision	Recall	F1	Precision	Recall	F1
BERT	0.910	0.910	0.910	0.951	0.941	0.946
LayoutLM	0.946	0.946	0.946	0.964	0.960	0.962
PICK	-	-	0.961	0.942	0.937	0.939

Table 5.2: Performance comparison on the SROIE dataset [23] between this implementation versus the performance stated in the corresponding papers.

	SROIE			Austrian Receipts		
BERT @ 50	Precision	Recall	F1	Precision	Recall	F1
COMPANY	0.942	0.931	0.936	0.289	0.235	0.259
ADDRESS	0.916	0.911	0.913	0.204	0.204	0.204
DATE	0.997	0.997	0.997	0.873	0.517	0.649
TOTAL	0.947	0.925	0.936	0.844	0.319	0.463
micro average	0.951	0.941	0.946	0.463	0.321	0.379

Table 5.3: The performance of BERT [13] (fine-tuned on SROIE training set) on the SROIE test set and Austrian Receipts dataset for each entity type.

	SROIE			Austrian Receipts		
LayoutLM @ 50	Precision	Recall	F1	Precision	Recall	F1
COMPANY	0.98	0.971	0.975	0.581	0.571	0.576
ADDRESS	0.925	0.922	0.924	0.292	0.274	0.283
DATE	1.000	1.000	1.000	0.950	0.942	0.946
TOTAL	0.951	0.945	0.948	0.911	0.429	0.583
micro average	0.964	0.96	0.962	0.661	0.558	0.605

Table 5.4: The performance of LayoutLM [85] (fine-tuned on SROIE training set) on the SROIE test set and Austrian Receipts dataset for each entity type.

SROIE				Austrian Receipts		
PICK @ 50	Precision	Recall	F1	Precision	Recall	F1
COMPANY	0.948	0.939	0.944	0.511	0.387	0.440
ADDRESS	0.907	0.902	0.905	0.090	0.062	0.073
DATE	0.997	0.997	0.997	0.000	0.000	0.000
TOTAL	0.916	0.908	0.912	0.375	0.101	0.159
micro average	0.942	0.937	0.939	0.316	0.138	0.192

Table 5.5: The performance of PICK [87] (fine-tuned on SROIE training set) on the SROIE test set and Austrian Receipts dataset for each entity type.

SROIE			Austrian	
	GT	OCR-engine	GT	OCR-engine
BERT	0.946	0.373	0.379	0.160
LayoutLM	0.962	0.390	0.605	0.278
PICK	0.939	0.240	0.192	0.086

Table 5.6: Performance comparison between using ground truth transcriptions and OCR engine results (EasyOCR).

5.4 Discussion

This section interprets the results and explores potential explanations for the behavior of the models.

The models used for this evaluation are close to the performance stated in the corresponding papers. Table 5.2 shows differences. The overperformance of BERT and LayoutLM could stem from the fixed OCR mismatches in the SROIE test-set. Overall the good performance validates the implementation and training of the models.

The following few sections explore the different entity types and their differences between the two datasets. The leftmost image in Figure 4.2 is used as an example to illustrate common errors. The extractions of the various models on this receipt are displayed in table 5.7.

Most receipts display the company name prominently at the top of the document. The SROIE and Austrian receipts share this property. The company name is an important field of interest to assign the document to an expense category. There are some difficulties regarding the extraction. The length can differ drastically, for example the Austrian Receipt dataset contains documents from *SPAR* and *OBI Bau- und Heimwerkermärkte Systemzentrale GmbH*. Especially Austrian receipts often incorporate the company name by displaying a logo or graphics instead of plain text. 50 out of 120 receipts display some kind of company logo. These graphics can change the look of the receipt and the

	Company	Address	Date	Total
Ground Truth	<i>SPAR</i> or <i>Eurospar</i>	<i>Wolfganggasse 42</i> <i>1120 WIEN</i>	<i>09.06.2021</i>	<i>2.01</i>
BERT	-	<i>42 1120 WIEN</i>	<i>09.06.2021</i>	-
LayoutLM	<i>SPAR Eurospar</i> <i>Wolfganggasse 42</i>	<i>1120 WIEN</i>	<i>09.06.2021</i>	<i>2.01</i>
PICK	<i>SPAR</i>	-	-	-

Table 5.7: Extraction results on example Austrian receipt.

position of the company name. Reviewing the mistakes for the different models reveals that most extractions contain at least the company name or part of it. LayoutLM can extract the correct company name more than half the time in the test set. The difference in performance to BERT may be explained by the additional positional information provided to LayoutLM, which might enable the model to determine new lines and font size differences. However, the precision and recall are not even close to the performance on the SROIE test set, indicating that the model is only partially able to transfer the knowledge from English to Austrian receipts.

The difference in performance in recognizing the address reinforces the difficulty in applying transfer learning on receipts. The best model, LayoutLM, is not able to extract even a third of the addresses correctly. Most of the predictions miss some part of the address, like the postal code or house number. An address can be quite long as it contains different elements like street name, house number, and postal code. Furthermore, these elements often span multiple lines on the receipts. The complexity and length of addresses might pose a problem for the models, as only exact matches are considered correct.

The date format in the SROIE dataset is different from the format usually seen on Austrian receipts. Most of the dates on SROIE receipts have the format *25/12/2018* whereas the dates on most Austrian receipts look like *25.12.2018*. A human reader can identify the similarity, but a model trained solely on the first format might not recognize the parallels. The PICK model does not have pre-training on millions of documents and only trains on the first date format. This might be the reason why it struggles with dates in the Austrian Receipts dataset. BERT and LayoutLM have extensive pre-training, which may introduce multiple date formats and their similarities to the model and therefore explain the better performance compared to PICK.

All three models do not even have an extraction for the total amount on half of the Austrian receipts. Similar to the date, the total amount has different formats in the SROIE and Austrian Receipts dataset. Most of the receipts in the SROIE dataset display their total amount with a decimal point, whereas Austrian receipts often use the decimal comma (73 out of 120). This discrepancy could introduce similar problems as discussed for the date entity.

One major difference between the SROIE and the Austrian Receipt dataset is the language. SROIE consists entirely of English receipts, while the Austrian receipts contain German language. Research on named entity recognition (NER) shows that even language models pre-trained on multiple languages have a significant dropoff in performance if evaluated on a language they are not fine-tuned on [56]. Given these results, the language could pose a significant challenge to the models.

The documents in SROIE and Austrian Receipt dataset differ not only in language but also in layout. Many Austrian receipts feature a logo or graphic, which changes not only the appearance but also the positioning of the information on the receipt. This might be a reason, why the models do not perform well on Austrian receipts. Research on few-example-learning with LayoutLM suggests that the model can achieve 90% of its full performance with only 32 documents in the fine-tuning stage [66]. This suggests that fine-tuning LayoutLM on a few Austrian receipts might yield good performance.

Conclusion and Future Work

Document understanding involves various problems, from optical character recognition to classification and information extraction. In particular, information extraction from documents is an active research area with strong influence from machine learning approaches, as shown by the state of the art. Current approaches often include 2D positional or visual information on top of textual information to boost their performance. Additionally, different approaches for document representation are considered. Graph and grid-based approaches try to directly incorporate the 2D positional information by encoding it in their representation. Language model approaches consider pre-training on millions of documents to obtain a general understanding of language and structure.

To evaluate the performance of current state-of-the-art models on Austrian receipts, BERT, LayoutLM, and PICK were implemented and trained on the SROIE dataset. Even though the performances of these models reach an F1 score of more than 0.9 on the SROIE test set, the best model (LayoutLM) achieved 0.6 on the Austrian Receipt dataset. The other models did considerably worse, with BERT reaching 0.379 and PICK only 0.192. The different language and layouts between receipts in the SROIE and Austrian Receipt dataset could be a reason for the performance drop. Therefore future work should consider fine-tuning pre-trained models on only a few examples of Austrian receipts, similar to the research of Sage et al. [66]. In their work, they explored information extraction on the SROIE dataset with a limited number of receipts and showed promising results using the LayoutLM model (90% of its full performance with only 32 documents). To combat the language difference between pre-training and fine-tuning, future research could explore employing a German version of BERT [8].

The performance on OCR extracted transcriptions was significantly worse, with less than half the F1 score. To use these information extraction models in large scale applications, the transcriptions need to be better. Therefore a commercial OCR engine instead of EasyOCR [Web2] could be considered in future work.

List of Figures

3.1	Architecture of one encoder layer in the Transformer encoder [76].	12
3.2	Illustration of the attention mechanism calculating the context vector of the second input [76].	14
3.3	Comparison of the composition of the input for the encoder between different language models. BERT combines Token, Segment and 1D-Positional embeddings [13]. LayoutLM [85], LAMBERT [90] and StructuralLM [38] extend this with 2D-Positional embeddings. LayoutLMv2 [84] and TILT [57] incorporate Visual embeddings as well.	16
3.4	Illustration of LayoutLM for information extraction on receipt images with late fusion of visual embeddings. [85]	17
3.5	Illustration of the attention mechanism with Positional Bias added to the attention score [84].	19
3.6	Illustration of Grid-creation by different approaches. Chargrid [28] works on character-level whereas BERTgrid [12] uses token-level embeddings. The system Tag, Copy and Predict [79] uses the TextLattice representation where each character occupies the same space.	22
3.7	Comparing different approaches to model the graph. The invoice reading system [45] treats each word as a node and creates an undirected edge between horizontal and vertical neighbors. GraphIE [59] treats text segments as nodes and has directed edges between nodes. The edge from top to bottom is different from bottom to top.	27
3.8	Illustration of PICK for information extraction on receipt images. [87] . . .	31
4.1	Number of companies over the number of receipts in the Austrian Receipt dataset.	37
4.2	Example receipts in the Austrian Receipt dataset.	39
5.1	An example from the SROIE dataset [23].	43

List of Tables

3.1	Comparison between the different approaches regarding the incorporated information and the need of serialization. If an evaluation is available on the SROIE dataset, performance can be compared using the F1 score.	33
5.1	Example of IOB2 tagging [68]. The 'B-' tag indicates the beginning and the 'I-' tag the inside of an entity. A word outside any entity is marked with 'O'.	45
5.2	Performance comparison on the SROIE dataset [23] between this implementation versus the performance stated in the corresponding papers.	49
5.3	The performance of BERT [13] (fine-tuned on SROIE training set) on the SROIE test set and Austrian Receipts dataset for each entity type.	49
5.4	The performance of LayoutLM [85] (fine-tuned on SROIE training set) on the SROIE test set and Austrian Receipts dataset for each entity type.	49
5.5	The performance of PICK [87] (fine-tuned on SROIE training set) on the SROIE test set and Austrian Receipts dataset for each entity type.	50
5.6	Performance comparison between using ground truth transcriptions and OCR engine results (EasyOCR).	50
5.7	Extraction results on example Austrian receipt.	51

Bibliography

- [1] Kiran Adnan and Rehan Akbar. An analytical study of information extraction from unstructured and multidimensional big data. *Journal of Big Data*, 6(1):1–38, 2019. doi:10.1186/s40537-019-0254-8.
- [2] Souhail Bakkali, Zuheng Ming, Mickaël Coustaty, and Marçal Rusiñol. Visual and textual deep feature fusion for document image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 2394–2403, 2020. doi:10.1109/CVPRW50498.2020.00289.
- [3] Raphaël Barman, Maud Ehrmann, Simon Clematide, Sofia Ares Oliveira, and Frédéric Kaplan. Combining visual and textual features for semantic segmentation of historical newspapers. *Journal of Data Mining and Digital Humanities*, 2021:1–26, 2021. doi:10.46298/JDM DH.6107.
- [4] Dipali Baviskar, Swati Ahirrao, Vidyasagar Potdar, and Ketan Kotecha. Efficient automated processing of the unstructured documents using artificial intelligence: A systematic literature review and future directions. *IEEE Access*, 9:72894–72936, 2021. doi:10.1109/ACCESS.2021.3072900.
- [5] Galal M. Binmakhashen and Sabri A. Mahmoud. Document layout analysis: A comprehensive survey. *ACM Computing Surveys*, 52(6):109, 2019. doi:10.1145/3355610.
- [6] Jerome Blanchard, Yolande Belaid, and Abdel Belaid. Automatic generation of a custom corpora for invoice analysis and recognition. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, page 1, 2019. doi:10.1109/ICDARW.2019.60121.
- [7] Manuel Carbonell, Pau Riba, Mauricio Villegas, Alicia Fornes, and Josep Lladós. Named entity recognition and relation extraction with graph neural networks in semi structured documents. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 9622–9627, 2021. doi:10.1109/ICPR48806.2021.9412669.
- [8] Branden Chan, Stefan Schweter, and Timo Möller. German’s next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, 2020. doi:10.18653/V1/2020.COLING-MAIN.598.

- [9] Xiaoxue Chen, Lianwen Jin, Yuanzhi Zhu, Canjie Luo, and Tianwei Wang. Text recognition in the wild: A survey. *ACM Computing Surveys*, 54(2):1–35, 2021. doi:10.1145/3440756.
- [10] Matteo Cristani, Andrea Bertolaso, Simone Scannapieco, and Claudio Tomazzoli. Future paradigms of automated processing of business documents. *International Journal of Information Management*, 40:67–75, 2018. doi:10.1016/J.IJINFOMGT.2018.01.010.
- [11] Vincent Poulain d’Andecy, Emmanuel Hartmann, and Marcal Rusinol. Field extraction by hybrid incremental and a-priori structural templates. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 251–256, 2018. doi:10.1109/DAS.2018.29.
- [12] Timo I. Denk and Christian Reisswig. Bertgrid: Contextualized embedding for 2d document representation and understanding. *Workshop on Document Intelligence at NeurIPS 2019*, 2019.
- [13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2018. doi:10.18653/V1/N19-1423.
- [14] Daniel Esser, Daniel Schuster, Klemens Muthmann, Michael Berger, and Alexander Schill. Automatic indexing of scanned documents: a layout-based approach. In *Document Recognition and Retrieval XIX*, volume 8297, 2012. doi:10.1117/12.908542.
- [15] Daniel Esser, Daniel Schuster, Klemens Muthmann, and Alexander Schill. Few-exemplar information extraction for business documents. In *ICEIS 2014 Proceedings of the 16th International Conference on Enterprise Information Systems - Volume 1*, pages 293–298, 2014. doi:10.5220/0004946702930298.
- [16] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, pages 1263–1272, 2017.
- [17] Ralph Grishman. Information extraction. *IEEE Intelligent Systems*, 2015. doi:10.1109/MIS.2015.68.
- [18] Marcel Hanke, Klemens Muthmann, Daniel Schuster, Alexander Schill, Kamil Aliyev, and Michael Berger. Continuous user feedback learning for data capture from business documents. In *HAIIS’12 Proceedings of the 7th international conference on Hybrid Artificial Intelligent Systems - Volume Part II*, pages 538–549, 2012. doi:10.1007/978-3-642-28931-6_51.

- [19] Adam W. Harley, Alex Ufkes, and Konstantinos G. Derpanis. Evaluation of deep convolutional nets for document image classification and retrieval. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 991–995, 2015. doi:10.1109/ICDAR.2015.7333910.
- [20] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):386–397, 2020. doi:10.1109/TPAMI.2018.2844175.
- [21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi:10.1109/CVPR.2016.90.
- [22] Benjamin Heinzerling and Michael Strube. BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages. In Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 7-12, 2018 2018. European Language Resources Association (ELRA). ISBN 979-10-95546-00-9.
- [23] Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and C. V. Jawahar. Icdar2019 competition on scanned receipt ocr and information extraction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1516–1520, 2019. doi:10.1109/ICDAR.2019.00244.
- [24] Wonseok Hwang, Jinyeong Yim, Seunghyun Park, Sohee Yang, and Minjoon Seo. Spatial dependency parsing for semi-structured document information extraction. In *ACL 2021: 59th annual meeting of the Association for Computational Linguistics*, pages 330–343, 2021. doi:10.18653/V1/2021.FINDINGS-ACL.28.
- [25] Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. Funsd: A dataset for form understanding in noisy scanned documents. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, pages 1–6, 2019. doi:10.1109/ICDARW.2019.10029.
- [26] Bo Jiang, Ziyang Zhang, Doudou Lin, Jin Tang, and Bin Luo. Semi-supervised learning with graph learning-convolutional networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11313–11320, 2019. doi:10.1109/CVPR.2019.01157.
- [27] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 60(1):493–502, 1972. doi:10.1108/EB026526.

- [28] Anoop Raveendra Katti, Christian Reisswig, Cordula Guder, Sebastian Brarda, Steffen Bickel, Johannes Höhne, and Jean Baptiste Faddoul. Chargrid: Towards understanding 2d documents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4459–4469, 2018. doi:10.18653/V1/D18-1476.
- [29] Mohamed Kerroumi, Othmane Sayem, and Aymen Shabou. Visualwordgrid: Information extraction from scanned documents using a multimodal approach. In *International Conference on Document Analysis and Recognition*, pages 389–402, 2021. doi:10.1007/978-3-030-86159-9_28.
- [30] Shachar Klaiman and Marius Lehne. Docreader: Bounding-box free training of a document information extraction model. In *Document Analysis and Recognition – ICDAR 2021*, pages 451–465. Springer International Publishing, 2021. ISBN 978-3-030-86549-8.
- [31] Bertin Klein, Stevan Agne, and Andreas Dengel. Results of a study on invoice-reading systems in germany. In *International Workshop on Document Analysis Systems*, pages 451–462, 2004. doi:10.1007/978-3-540-28640-0_43.
- [32] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura E. Barnes, and Donald E. Brown. Text classification algorithms: A survey. *Information-an International Interdisciplinary Journal*, 10(4):150, 2019. doi:10.3390/INFO10040150.
- [33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of The ACM*, 60(6):84–90, 2017. doi:10.1145/3065386.
- [34] Zhanghui Kuang, Hongbin Sun, Zhizhong Li, Xiaoyu Yue, Tsui Hin Lin, Jianyong Chen, Huaqiang Wei, Yiqin Zhu, Tong Gao, Wenwei Zhang, Kai Chen, Wayne Zhang, and Dahua Lin. Mmocr: A comprehensive toolbox for text detection, recognition and understanding. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3791–3794, 2021. doi:10.1145/3474085.3478328.
- [35] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML ’01, page 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558607781.
- [36] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*, pages 260–270, 2016. doi:10.18653/V1/N16-1030.

- [37] D. Lewis, G. Agam, S. Argamon, O. Frieder, D. Grossman, and J. Heard. 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*, pages 665–666, 2006. doi:10.1145/1148170.1148307.
- [38] Chenliang Li, Bin Bi, Ming Yan, Wei Wang, Songfang Huang, Fei Huang, and Luo Si. Structurallm: Structural pre-training for form understanding. In *ACL 2021: 59th annual meeting of the Association for Computational Linguistics*, pages 6309–6318, 2021. doi:10.18653/V1/2021.ACL-LONG.493.
- [39] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, (1):1–1, 2020. doi:10.1109/TKDE.2020.2981314.
- [40] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017. doi:10.1109/CVPR.2017.106.
- [41] Weihong Lin, Qifang Gao, Lei Sun, Zhuoyao Zhong, Kai Hu, Qin Ren, and Qiang Huo. Vibertgrid: A jointly trained multi-modal 2d document representation for key information extraction from documents. *Document Analysis and Recognition - ICDAR 2021*, 2021. doi:10.1007/978-3-030-86549-8_35.
- [42] Li Liu, Zhiyu Wang, Taorong Qiu, Qiu Chen, Yue Lu, and Ching Y. Suen. Document image classification: Progress over two decades. *Neurocomputing*, 453:223–240, 2021. doi:10.1016/J.NEUCOM.2021.04.114.
- [43] Xiaojing Liu, Feiyu Gao, Qiong Zhang, and Huasha Zhao. Graph convolution for multimodal information extraction from visually rich documents. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 32–39, 2019. doi:10.18653/V1/N19-2005.
- [44] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
- [45] Devashish Lohani, Abdel Belaïd, and Yolande Belaïd. An invoice reading system using a graph convolutional network. In *International Workshop on Robust Reading*, pages 144–158, 2018. doi:10.1007/978-3-030-21074-8_12.
- [46] Shangbang Long, Xin He, and Cong Yao. Scene text detection and recognition: The deep learning era. *International Journal of Computer Vision*, 129(1):161–184, 2021. doi:10.1007/S11263-020-01369-0.

- [47] Bodhisattwa Prasad Majumder, Navneet Potti, Sandeep Tata, James Bradley Wendt, Qi Zhao, and Marc Najork. Representation learning for information extraction from form-like documents. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6495–6504, 2020. doi:10.18653/V1/2020.ACL-MAIN.580.
- [48] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, 2008. ISBN 0521865719. doi:10.1017/CBO9780511809071.
- [49] Mrunal G. Marne, Pravin R. Futane, Sakshi B. Kolekar, Aditya D. Lakhadive, and Snehwardhan K. Marathe. Identification of optimal optical character recognition (ocr) engine for proposed system. In *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 2018. doi:10.1109/ICCUBEA.2018.8697487.
- [50] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, volume 26, pages 3111–3119, 2013. URL: <https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf>.
- [51] Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. Deep learning-based text classification: A comprehensive review. *ACM Computing Surveys*, 54(3):1–40, 2021. doi:10.1145/3439726.
- [52] Hamid Motahari, Nigel Duffy, Paul Bennett, and Tania Bedrax-Weiss. A report on the first workshop on document intelligence (di) at neurips 2019. *SIGKDD Explor. Newsl.*, 22(2):8–11, jan 2021. doi:10.1145/3447556.3447563.
- [53] Rasmus Berg Palm, Ole Winther, and Florian Laws. Cloudscan - a configuration-free invoice analysis system using recurrent neural networks. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pages 406–413, 2017. doi:10.1109/ICDAR.2017.74.
- [54] Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. Cord: A consolidated receipt dataset for post-ocr parsing. *Workshop on Document Intelligence at NeurIPS 2019*, 2019.
- [55] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, pages 8026–8037, 2019. URL: <https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf>.

- [56] Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual bert. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, 2019. doi:10.18653/v1/P19-1493.
- [57] Rafal Powalski, Lukasz Borchmann, Dawid Jurkiewicz, Tomasz Dwojak, Michal Pietruszka, and Gabriela Palka. Going full-tilt boogie on document understanding with text-image-layout transformer. In *Document Analysis and Recognition – ICDAR 2021*, pages 732–747. Springer International Publishing, 2021. ISBN 978-3-030-86549-8.
- [58] Devashish Prasad, Ayan Gadpal, Kshitij Kapadni, Manish Visave, and Kavita Sultanpure. Cascadetabnet: An approach for end to end table detection and structure recognition from image-based documents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 2439–2447, 2020. doi:10.1109/CVPRW50498.2020.00294.
- [59] Yujie Qian, Enrico Santus, Zhijing Jin, Jiang Guo, and Regina Barzilay. Graphie: A graph-based framework for information extraction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 751–761, 2018. doi:10.18653/v1/N19-1082.
- [60] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.
- [61] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017. doi:10.1109/TPAMI.2016.2577031.
- [62] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015. doi:10.1007/978-3-319-24574-4_28.
- [63] Marcal Rusinol, Tayeb Benkhelfallah, and Vincent Poulain d’Andecy. Field extraction from administrative documents by incremental structural templates. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1100–1104, 2013. doi:10.1109/ICDAR.2013.223.
- [64] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. doi:10.1007/s11263-015-0816-y.

- [65] Clement Sage, Alexandre Aussem, Haytham Elghazel, Veronique Eglin, and Jeremy Espinas. Recurrent neural network approach for table field extraction in business documents. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1308–1313, 2019. doi:10.1109/ICDAR.2019.00211.
- [66] Clément Sage, Thibault Douzon, Alex Aussem, Véronique Eglin, Haytham Elghazel, Stefan Duffner, Christophe Garcia, and Jérémy Espinas. Data-efficient information extraction from documents with pre-trained language models. In *Document Analysis and Recognition – ICDAR 2021 Workshops*, pages 455–469. Springer International Publishing, 2021. ISBN 978-3-030-86159-9.
- [67] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: language-independent named entity recognition. In *CONLL '03 Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, pages 142–147, 2003. doi:10.3115/1119176.1119195.
- [68] Erik F. Tjong Kim Sang and Jorn Veenstra. Representing text chunks. In *EACL '99 Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, pages 173–179, 1999.
- [69] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *15th International Conference on Extended Semantic Web Conference, ESWC 2018*, pages 593–607, 2018. doi:10.1007/978-3-319-93417-4_38.
- [70] Daniel Schuster, Klemens Muthmann, Daniel Esser, Alexander Schill, Michael Berger, Christoph Weidling, Kamil Aliyev, and Andreas Hofmeier. Intellix – end-user trained information extraction for document archiving. In *2013 12th International Conference on Document Analysis and Recognition*, pages 101–105, 2013. doi:10.1109/ICDAR.2013.28.
- [71] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1073–1083, 2017. doi:10.18653/V1/P17-1099.
- [72] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1715–1725, 2016. doi:10.18653/V1/P16-1162.
- [73] Fotini Simistira, Manuel Bouillon, Mathias Seuret, Marcel Wursch, Michele Alberti, Rolf Ingold, and Marcus Liwicki. Icdar2017 competition on layout analysis for challenging medieval manuscripts. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pages 1361–1370, 2017. doi:10.1109/ICDAR.2017.223.

- [74] Nishant Subramani, Alexandre Matton, Malcolm Greaves, and Adrian Lam. A survey of deep learning approaches for ocr and document understanding. *arXiv: Computation and Language*, 2020.
- [75] Hongbin Sun, Zhanghui Kuang, Xiaoyu Yue, Chenhao Lin, and Wayne Zhang. Spatial dual-modality graph reasoning for key information extraction. *arXiv preprint arXiv:2103.14470*, 2021.
- [76] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, volume 30, pages 5998–6008, 2017. URL: <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>.
- [77] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. doi:10.17863/CAM.48429.
- [78] Jiapeng Wang, Chongyu Liu, Lianwen Jin, Guozhi Tang, Jiaxin Zhang, Shuaitao Zhang, Qianying Wang, Yaqiang Wu, and Mingxiang Cai. Towards robust visual information extraction in real world: New dataset and novel solution. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(4):2738–2745, May 2021. URL: <https://ojs.aaai.org/index.php/AAAI/article/view/16378>.
- [79] Jiapeng Wang, Tianwei Wang, Guozhi Tang, Lianwen Jin, Weihong Ma, Kai Ding, and Yichao Huang. Tag, copy or predict: A unified weakly-supervised learning framework for visual information extraction using sequences. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, volume 2, pages 1082–1090, 2021. doi:10.24963/IJCAI.2021/150.
- [80] Mengxi Wei, Yifan He, and Qiong Zhang. Robust layout-aware ie for visually rich documents with pre-trained language models. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2367–2376, 2020. doi:10.1145/3397271.3401442.
- [81] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, 2020. doi:10.18653/v1/2020.emnlp-demos.6.
- [82] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian,

- Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.
- [83] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks*, 32(1):4–24, 2021. doi:10.1109/TNNLS.2020.2978386.
- [84] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. In *ACL 2021: 59th annual meeting of the Association for Computational Linguistics*, pages 2579–2591, 2021. doi:10.18653/V1/2021.ACL-LONG.201.
- [85] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1192–1200, 2020. doi:10.1145/3394486.3403172.
- [86] Yiheng Xu, Tengchao Lv, Lei Cui, Guoxin Wang, Yijuan Lu, Dinei Florêncio, Cha Zhang, and Furu Wei. Layoutxlm: Multimodal pre-training for multilingual visually-rich document understanding. *arXiv preprint arXiv:2104.08836*, 2021.
- [87] Wenwen Yu, Ning Lu, Xianbiao Qi, Ping Gong, and Rong Xiao. Pick: Processing key information extraction from documents using improved graph learning-convolutional networks. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4363–4370, 2021. doi:10.1109/ICPR48806.2021.9412927.
- [88] Peng Zhang, Yunlu Xu, Zhazhan Cheng, Shiliang Pu, Jing Lu, Liang Qiao, Yi Niu, and Fei Wu. Trie: End-to-end text reading and information extraction for document understanding. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1413–1422, 2020. doi:10.1145/3394171.3413900.
- [89] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020. doi:10.1016/J.AIOOPEN.2021.01.001.
- [90] Łukasz Garncarek, Rafał Powalski, Tomasz Stanisławek, Bartosz Topolski, Piotr Halama, Michał Turski, and Filip Graliński. Lambert: Layout-aware (language) modeling for information extraction. *International Conference on Document Analysis and Recognition*, 2021. doi:10.1007/978-3-030-86549-8_34.

Weblinks

- [Web1] 16th International Conference on Document Analysis and Recognition ICDAR 2021. <https://icdar2021.org/>, 2021. [Online; accessed 28-12-2021].
- [Web2] EasyOCR. <https://github.com/JaidedAI/EasyOCR>, 2021. [Online; accessed 28-12-2021].
- [Web3] PICK-Pytorch implementation. <https://github.com/wenwenyu/PICK-pytorch>, 2021. [Online; accessed 28-12-2021].
- [Web4] PPOCRLabel. <https://github.com/Evezzerest/PPOCRLabel>, 2021. [Online; accessed 28-12-2021].
- [Web5] KDD 2021. Document Intelligence Workshop. <https://document-intelligence.github.io/DI-2021/>, 2021. [Online; accessed 28-12-2021].
- [Web6] The Apache Software Foundation. Apache Lucene. <https://lucene.apache.org/>, 2021. [Online; accessed 28-12-2021].
- [Web7] Inc. Heartex. Label Studio. <https://labelstud.io/>, 2021. [Online; accessed 28-12-2021].
- [Web8] Google LLC. Google Colaboratory. <https://colab.research.google.com/>, 2021. [Online; accessed 28-12-2021].

Source-Code

```
def split_bbox(line: dict, idx: int):
    """Create two entries while splitting the bbox assuming equal length
        characters retaining original label

    Args:
        line (dict): line to split
        idx (int): split-index in txt

    Returns:
        [dict] containing two lines
    """

    txt = line['txt']
    bbox = line['bbox']
    bbox_width = bbox[2] - bbox[0]
    bbox_width_per_char = bbox_width / len(txt)

    split_lines = []

    center = int(bbox[0] + bbox_width_per_char * idx)
    split_lines.append({'bbox': [bbox[0], bbox[1], center, bbox[3]], 'txt':
        txt[:idx].strip(), 'label': line['label']})
    split_lines.append({'bbox': [center, bbox[1], bbox[2], bbox[3]], 'txt':
        txt[idx:].strip(), 'label': line['label']})

    return split_lines

def cut_entity_with_bbox(line: dict, substring: str, label: str, alt_label:
    str):
    """Split the line in 1 to 3 pieces with the substring line having the
        label 'label' and the others 'alt_label'

    Args:
        line (dict): line containing substring
        substring (str): to cut from the original
        label (str): label for the line containing only the substring
        alt_label (str): label for the lines not containing the substring

    Returns:
        [dict] array of lines
    """
```

```

txt = line['txt']
idx = txt.find(substring)

entity_lines = []
if idx < 0:
    line['label'] = alt_label
    return [line]

if idx > 0:
    splitted_lines = split_bbox(line, idx)
    splitted_lines[0]['label'] = alt_label
    entity_lines.append(splitted_lines[0])
    line = splitted_lines[1]

if idx + len(substring) < len(txt):
    splitted_lines = split_bbox(line, len(substring))
    splitted_lines[0]['label'] = label
    splitted_lines[1]['label'] = alt_label
    entity_lines.append(splitted_lines[0])
    entity_lines.append(splitted_lines[1])

else:
    line['label'] = label
    entity_lines.append(line)

return entity_lines

def remove_empty_words(labeled_words):
    """remove lines containing only an empty string"""
    return [word for word in labeled_words if word['txt'] != '']

def assign_labels(lines: list, entities: dict):
    """Assing labels to lines by searching if the line is in an entity or the
    entity is in the line.
    If one of those is the case, the correspoinding words get assigned with
    the label.
    If a word does not belong to an entity, its assigned the alt label 'O'.
    It futhermore splits the txt int the lines into words (splitting the
    boundingboxes assuming equal width characters).

    Args:
        lines (list): the lines containing bbox, txt, and label
        entities (dict): the ground-truth labels

    Returns:
        [dict] lines (txt, bbox, label) containing each a word and label
    """
    labeled_lines = []
    alt_label = 'O'
    for line in lines:
        txt = line['txt']

```

```

#remove white space (avoid mismatch between ground truth and OCR)
stripped_txt = txt.replace(' ', '')

#single or two character words are labeled 'O'
if len(stripped_txt) <= 2:
    line['label'] = alt_label
    labeled_lines.append(line)
    continue

line_labeled = False
for entity in sorted(list(entities.keys()), reverse = True) :
    gt_value = entities[entity]
    stripped_gt_value = gt_value.replace(' ', '')

    if stripped_gt_value == '':
        continue

    if entity == 'ADDRESS':
        def remove_punctuation(text):
            return text.translate(text.maketrans('', '', string.punctuation))

        compare_txt = remove_punctuation(stripped_txt)
        compare_gt_value = remove_punctuation(stripped_gt_value)
        if len(compare_txt) <= 3 or compare_txt.isnumeric():
            continue
        else:
            compare_txt = stripped_txt
            compare_gt_value = stripped_gt_value

    if compare_gt_value in compare_txt:
        r_lines = cut_entity_with_bbox(line, gt_value, 'B-' + entity,
            alt_label)
        labeled_lines.extend(r_lines)
        line_labeled = True
        break

    if entity in ['ADDRESS', 'COMPANY']:
        if compare_txt in compare_gt_value:
            if compare_gt_value.startswith(compare_txt):
                line['label'] = 'B-' + entity
            else:
                line['label'] = 'I-' + entity
            labeled_lines.append(line)
            line_labeled = True
            break

if not line_labeled:
    line['label'] = alt_label
    labeled_lines.append(line)

return remove_empty_words(labeled_lines)

```