University of Brasilia

Institute of Exact Sciences
Department of Computer Science

Visual and Textual Feature Fusion for Document Analysis

Patricia Medyna Lauritzen de Lucena Drumond

Document presented for qualifying examination of the Ph.D. Program in Computer Science

Supervisor
Prof. Dr. Teófilo Emidio de Campos

Joint Supervisor
Prof. Dr. Fabrício Ataídes Braz

Brasilia
2022

University of Brasilia

Institute of Exact Sciences
Department of Computer Science

Visual and Textual Feature Fusion for Document Analysis

Patricia Medyna Lauritzen de Lucena Drumond

Document presented for qualifying examination of the Ph.D. Program in Computer Science

Prof. Dr. Teófilo Emidio de Campos (Supervisor)
CIC/UnB

Prof. Dr. Li Weigang
CIC/UnB

Prof. Carolina Scarton, PhD
University of Sheffield

Prof. Dr. Ricardo Pezzuol Jacobi
Computer Science Graduate Program Coordinator

Brasilia, November 11, 2022

Abstract

The large volume of documents produced daily in all sectors, such as industry, commerce, and government agencies, has increased the number of researches aimed at automating the process of reading, understanding, and analyzing documents [15]. Business documents can be born digital, as electronic files, or can be a digitized form that comes from writing or printed on paper. In addition, these documents often come in various layouts and formats. They can be organized in different ways, from plain text, multi-column layouts, and a wide variety of tables/forms/figures. In many documents, the spatial relationship of text blocks usually contains important semantic information for downstream tasks. The relative position of text blocks plays a crucial role in document understanding. However, the task of embedding layout information in the representation of a page instance is not trivial. In the last decade, Computer Vision (CV) and Natural Language Processing (NLP) pre-training techniques have been advancing in extracting content from document images considering visual, textual, and layout features. Deep learning methods, especially the pre-training technique, represented by Transformer architecture [48], have become a new paradigm for solving various downstream tasks. However, a major drawback of such pre-trained models is that they require a high computational cost. Unlike these models, we propose a simple and traditional rule-based spatial layout encoding method, which combines textual and spatial information from text blocks. We show that this enables a standard NLP pipeline to be significantly enhanced without requiring expensive mid or high-level multimodal fusion. We evaluate our method on two datasets, Tobacco800 [56] and RVL-CDIP [19], for document image classification tasks. The document classification performed with our method obtained an accuracy of 83.6% on the large-scale RVL-CDIP [19] and 99.5% on the Tobacco800 [56] datasets. In order to validate the effectiveness of our method, we intend to carry out more experiments. First, we will use other more robust datasets. Then we will change parameters such as quadrant amounts, insertion/deletion of positional tokens, and other classifiers.

Keywords: Document Intelligence, Computer Vision, Natural Language Processing, Document Image Classification

Resumo Expandido

Diariamente é produzido um grande volume de documentos nas organizações industriais, comerciais, governamentais, entre outras. Além disso, com o mercado competitivo na internet, as transações de negócios têm crescido numa velocidade imensa. Esses fatos aumentam cada vez mais a necessidade da automação e extração de informações de documentos. Os documentos podem ter sido originados digitalmente como um arquivo eletrônico ou podem ser uma cópia digitalizada de documento impresso em papel. Além disso, esses documentos, geralmente, são ricos de informações visuais e podem estar organizados de diferentes maneiras, desde páginas simples contendo apenas texto, até páginas com layouts de várias colunas de texto e uma ampla variedade de elementos não textuais como figuras e tabelas. Para análise e classificação desses documentos a extração de informações baseadas somente em blocos de texto ou em características visuais nem sempre é eficaz. Em geral, a relação espacial desses elementos e blocos de texto contém informações semânticas cruciais para compreensão de documentos.

O processo de automação da análise e extração de informações de documentos é desafiador devido aos vários formatos e layouts dos documentos de negócios, e tem atraído a atenção em áreas de pesquisa como Visão Computacional (CV) e Processamento de Linguagem Natural (NLP). *Document Intelligence* é um termo recente utilizado para aplicações da Inteligência Artificial que envolve a automatização de leitura, compreensão e análise de documentos visualmente ricos de informação [53]. O primeiro workshop de *Document Intelligence* (DI'2019) foi realizado no dia 14 de dezembro de 2019 na Conferência sobre Sistemas de Processamento de Informações Neurais (NeurIPS) em Vancouver, Canadá. Essas aplicações, também conhecidas como *Document AI*, são geralmente desenvolvidas para resolver tarefas como análise de layout de documentos, extração de informações visuais, resposta-pergunta visuais de documento e classificação de imagem de documentos, etc.

Na última década, várias abordagens multimodais [1, 6, 51, 52] unindo técnicas de CV e NLP vêm avançando em tarefas de compreensão de documentos, como por exemplo, análise de layout, segmentação de páginas e classificação de imagens de documentos considerando a junção de pelo menos duas das modalidades de recursos: visuais, textuais

e de layout. Existem algumas abordagens que foram propostas para lidar com layouts nas imagens do documento. As abordagens tradicionais baseadas em regras (top-down, bottom-up e híbridas) e as abordagens baseadas em Machine Learning e Deep Learning. No entanto, o surgimento da abordagem Deep Learning, principalmente com as técnicas de pré-treinamento, utilizando Redes Neurais Convolucionais e Arquitetura Transformer [48] tem avançado em pesquisa reduzindo o número de pesquisas com abordagens tradicionais.

A tecnologia de Deep Learning usada em *Document Intelligence* envolve a extração de informações de diferentes tipos de documentos através de ferramentas de extração, como OCR, extração de HTML/XML e PDF. As informações de texto, layout e visuais depois de extraídas são pre-treinadas em redes neurais para realizar as tarefas downstream. O modelo de linguagem BERT (Bidirectional Encoder Representations from Transformers) [16] tem sido usado como backbone para outros modelos de pre-treinamento [21, 30, 53, 54] combinando recursos visuais e textuais para tarefas downstream. Apesar do excelente desempenho dos modelos Transformer existem vários desafios associados à sua aplicabilidade para configurações prática. Os gargalos mais importantes incluem requisitos para grandes quantidades de dados de treinamento e altos custos computacionais associados.

Ao contrário desses modelos, nós propomos um método de codificação de layout espacial simples e tradicional baseado em regras, LayoutQT, que combina informações textuais e espaciais de blocos de texto. Nós mostramos que isso permite que um pipeline de NLP padrão seja significativamente aprimorado sem exigir custos de fusão multimodal de médio ou alto nível. O LayoutQT divide a imagem de documento em quadrantes e associa a cada quadrante um token. Na extração de blocos de texto, são inseridos os tokens relativo às posições de início e fim dos blocos de texto. Além disso, foram inseridos tokens relativos às posições centrais de texto. Para avaliar nosso método, nós realizamos experimentos de classificação de documentos utilizando as redes neurais LSTM [45] e AWD-LSTM [36] em duas bases de dados, Tobacco800 [28] e RVL-CDIP [19], publicamente acessíveis. Além disso, como baseline realizamos os mesmos experimentos sem o nosso método. A classificação de documentos realizada com nosso método obteve uma precisão de 83,6% na base de dados RVL-CDIP de grande escala e 99,5% na base de dados Tobacco800. RVL-CDIP contém 400.000 imagens de documentos divididos em 16 classes e é utilizada para classificação de documentos, enquanto a Tobacco800, possui 1.290 imagens de documentos dividida em duas classes (FirstPage e NextPage), utilizada para classificar se a imagem é a primeira página de um documento ou se é uma página de continuidade. Em seguida, nós pesquisamos na literatura outras base de dados compatíveis com as já utilizadas em nossa abordagem para o problema de classificação de documentos. As bases de dados encontradas que são disponíveis publicamente foram: Tobacco-3482 [26] e VICTOR [3].

A Tobacco-3482 é composta por 3.482 imagens de documentos dividida em 10 classes sendo um subconjunto da base de dados RVL-CDIP. VICTOR é uma base de dados mais robusta contendo 692.966 documentos de processos judiciais do Supremo Tribunal Federal (STF)do Brasil compreendendo 4.603.784 páginas dividida em 6 classes. Essa base de dados faz parte de um projeto com mesmo nome, resultado da parceria entre a UnB, STF e a Finep.

Para trabalhos futuros, iremos realizar mais experimentos com nosso modelo modificando os parâmetros. Nos experimentos realizados anteriormente, nós utilizamos uma quantidade fixa de 24 quadrantes, ou seja, nós dividimos a imagem em regiões verticais por 6 regiões horizontais. Para validar nosso modelo, pretendemos variar a quantidade de quadrantes e comparar os resultados. Além disso, nós iremos utilizar as duas bases de dados já utilizadas, Tobacco800 e RVL-CDIP e acrescentar aos experimentos a base VICTOR por ser mais robusta e diferente das anteriores para tarefa de classificação.

Palavras-chave: Inteligência de Documento, Visão Computacional, Processamento de Linguagem Natural, Classificação de Imagem de Documento

Contents

List of Acronyms and Abbreviations **ix**

1 Introduction **1**

1.1 Contextualization . . . 1

1.2 Objectives . . . 3

1.3 Contributions . . . 3

1.4 Document Outline . . . 3

2 Background **4**

2.1 Document Artificial Intelligence . . . 4

2.2 Document Classification . . . 6

2.3 Processes of physical layout analysis . . . 7

2.4 Rule-based Approaches . . . 10

2.5 Feature Engineering . . . 11

2.6 Machine Learning Approaches . . . 11

2.7 Deep Learning Approaches . . . 13

2.7.1 Convolutional Neural Network . . . 14

2.7.2 Recurrent Neural Networks . . . 14

2.7.3 Long Short Term Memory networks . . . 15

2.7.4 Transformer . . . 16

2.7.5 Pretraining Objectives . . . 20

2.8 Other Recent Works . . . 22

2.9 Summary . . . 25

3 Datasets **28**

3.1 Tobacco800 . . . 28

3.2 RVL-CDIP . . . 30

3.3 Tobacco-3482 . . . 31

3.4 VICTOR . . . 32

3.5 Datasets from Websites . . . 34

3.6 Summary 35

4 LayoutQT 37

4.1 Preprocessing LayoutQT 37

4.2 Experiments 39

4.2.1 Experiment details 40

4.2.2 Experiment Setting 41

4.3 Results and Discussion 42

4.4 Summary 46

5 Concluding Remarks and Proposal 47

5.1 Conclusion 47

5.2 Schedule 47

References 50

List of Acronyms and Abbreviations

AI Artificial Intelligence

AMLM Area-Masked Language Model

ANN Artificial Neural Network

ASGD Asynchronous Stochastic Gradient Descent

AWD-LSTM ASGD Weight-Dropped Long Short-Term Memory

BERT Bidirectional Encoder Representations from Transformers

BPTT Backpropagation Through Time

BVic Big VICTOR

CCs Connected Components

CDIP Complex Document Information Processing

CNN Convolutional Neural Network

CPC Cell Position Classification

CV Computer Vision

DLA Document Layout Analysis

DVFE Deep Visual Feature Extractor

FFN FeedForward Network

GNN Graph Neural Network

HMM Hidden Markov Model

HTML HyperText Markup Language

IIT Illinois Institute of Technology

KNN K-Nearest Neighbor

LayoutQT Layout Quadrant Tags

LSTM Long Short-Term Memory

LTDL Legacy Tobacco Documents Library

LTR Learning-To-Reconstruct

MDC Multi-label Document Classification

MLM Masked Language Modeling

MLP Multilayer Perceptron

MM-MLM Multi-Modal Masked Language Modeling

MVic Medium VICTOR Dataset

MVLM Masked Visual-Language Model

NeurIPS Conference on Neural Information Processing Systems

NLP Natural Language Processing

NSP Next Sentence Prediction

OCR Optical Caracter Recognition

PSS Page Stream Segmentation

R-CNN Regions with CNN features

RNN Recurrent Neural Network

RVL-CDIP Ryerson Vision Lab Complex Document Information Processing

SAM Sequential Association Module

STF Supremo Tribunal Federal

SVFE Shallow Visual Feature Extractor

SVic Small VICTOR Dataset

SVic+ Extension of Small VICTOR Dataset

SVM Support Vector Machines

TDI Text Describes Image

TFE Text Feature Extractor

TIA Text-Image Alignment

TIM Text-Image Matching

UCSF University of California San Francisco

ULMFiT Universal Language Model Fine-Tuning

UnB Universidade de Brasília

VQA Visual Question Answering

Chapter 1

Introduction

This Chapter contains a brief contextualization of our field of study, the motivation, and the statement of the problem we intend to face. It also includes our objectives, the contributions we have achieved, and the expected contributions. To conclude the Chapter, an outline of the entire document is presented.

1.1 Contextualization

Business documents are essential for the operation carried out in their organizations. Automated processing has helped to organize and extract information from these documents. However, the massive amount of digitized documents produced in the last decades requires a significant effort in developing document image processing methods for information extraction. In addition, the information in business documents is presented in various ways, from plain text, multi-column formats, and a wide variety of tables. These documents often reflect complex legal agreements and refer explicitly or implicitly to regulations, legislation, case law, and standard business practices [53].

Documents follow some layout, including vital structural and visual information (e.g., font sizes and geographic position of the text). It is important to locate the region of the structural elements, like text, figures, and tables; it contains most of the document layout information. Figure 1.1 presents four examples of documents with different layouts. Consequently, the information is not easily accessible for extraction and recognition. Layout analysis is, therefore, an important step in machine based document understanding, and it strongly depends on the detection of structural elements contained in the documents [34].

Analyzing digitized documents is a task that has advanced over the years with the growth of methods from computer vision (CV) and natural language processing (NLP). Computer vision methods have been used for optical character recognition (OCR) sys-

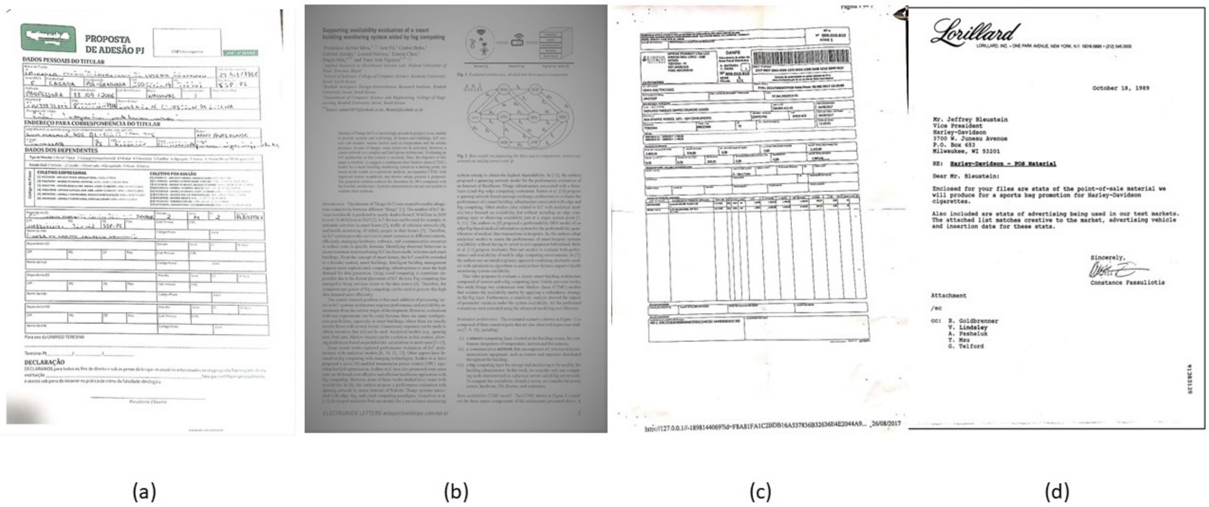


Figure 1.1: Examples of document images with different visual styles (a) a form (b) a scientific publication page (c) an invoice and (d) a memo.

tems to extract text from image documents based on their visual appearance [6]. To some extent, OCR could be a solution that can extract the text from an image of a document and convert it into computer readable form, which may further be used for editing. Nonetheless, OCR is prone to errors and is not always applicable to all documents, e.g. handwriting text is still difficult to read, and those document images must have high resolution [38]. The main issue with traditional OCR is that it does not extract and attach the positional values of the text with extracted text [18].

On the other hand, much of the relevant information is in the text, so extracting text-based information from documents has been the subject of NLP studies for some time. However, a system cannot rely on text alone but requires incorporating structure and image information. Although the text allows retrieving information about the document's content, the visual layout plays an equally important role [38]. The document layout comprises both the structure and visual information (e.g. font sizes, text centring, location of parts of the text) that are vital to the understanding of the document by readers and but often ignored by models that consider only the textual content. Thus, combining visual, textual, and document image layout resources in extracting information is of great importance [24]. Contemporary approaches to document AI are often built by combining computer vision and natural language processing perspectives.

1.2 Objectives

This research aims to propose, implement and evaluate document processing methods that combine textual information and layout by performing experiments on different downstream tasks and datasets with low computational cost. More specifically, we aim to:

1) propose a joint feature learning approach that combines positional information of text block and text embeddings for extracting information. 2) evaluate this approach for for document classification and page segmentation.

3) compare the models with baselines.

1.3 Contributions

Our main contributions are:

- A novel approach to fuse textual and layout information which exploits a by product of the text digitalization process, incurring in insignificant additional computational cost.
- The simple yet effective method of fusion textual and layout features for extracting information from documents that only requires increasing text embedding by injecting spatial tokens concerning text block positions.
- The source code of our library, which is available from `https://github.com/patriciamedyna/LayoutQT` and the package on `https://pypi.org/project/docSilhouette`. It can be used immediately in the engineering of other products.

1.4 Document Outline

This manuscript is structured in 5 chapters. Chapter 1 consists in this introduction. In Chapter 2, we present some general knowledge related to the development of Document Intelligence systems.

Chapter 3 describes benchmarks some publicly available Document AI, including the RVL-CDIP dataset for document image classification.

Chapter 4 describes the contributions achieved so far. It presents the methodology, results and conclusions.

Chapter 5 describes the plan we expect to follow to conclude this research project.

Chapter 2

Background

This Chapter introduces Document Intelligence Systems and their applications to downstream tasks such as document layout analysis, visual information extraction, document visual question answering, document image classification, etc. In addition, it reviews some traditional and Deep Learning techniques used to extract visual and textual features. Finally, it presents the most recent works developed.

2.1 Document Artificial Intelligence

Document AI, or Document Intelligence [37], is an application of Artificial Intelligence (AI) that involves automatic reading, comprehension and analysis of business documents. It is very challenging due to the diversity of layouts and formats from webpages, digital-born or scanned documents, low-quality scanned document images and the template structure's complexity. With the various structures of business document images, extracting semantic information from its textual content favours downstream tasks such as document retrieval, information extraction, and text classification [15].

The first workshop on Document Intelligence was held on December 14, 2019 at Conference on Neural Information Processing Systems (NeurIPS) in Vancouver, Canada [37]. Document Intelligence is a research topic that has been growing in recent years involving natural language processing and computer vision. With the acceleration of digitization, the structured analysis and content extraction of documents, images, and others has become a key part of digital success. Key information extraction from business document images requires understanding texts in various layouts. Many AI technologies have advanced to improve the use and handling of industrial documents, such as machine [34] and deep learning [57].

Deep learning methods have become a new paradigm for solving many machine learning problems. In addition, most recent approaches try to solve the task by developing pre-training language models [21, 30, 53, 54] focusing on combining visual features from document images with texts and their layout using a unified Transformer architecture [48]. The development of Document AI also reflects a similar trend with other applications in deep learning, especially in the pre-training technique represented by Convolutional Neural Networks (CNN), Graph Neural Networks (GNN), and Transformer architecture.

Among all these approaches, a typical pipeline for pre-training Document AI models usually starts with the vision-based understanding, such as Optical Character Recognition (OCR) or document layout analysis. In real-world application scenarios, a typical Document Intelligence System mainly includes four types of tasks, namely: Document Layout Analysis, Visual Information Extraction, Document Visual Question Answering, and Document Image Classification [15].

Document Layout Analysis (DLA) is a means to identify different functional/logical content elements (e.g. sentences, titles, captions, author names, and addresses) on a given page. It is realized by segmenting physical contents (e.g. pixels, characters, words, lines, figures, tables, and background) on the page and classifying them into predefined functional/logical categories, in other words, by assigning these classified entity labels. Document layout analysis plays a crucial role within the document digitization procedure because the correctness of layout analysis determines whether a subsequent text recognition procedure is operated on the correct text object. When implementing layout analysis, there are generally two approaches to carry out this procedure, the top-down approach and bottom-up approach [32], discussed in section 2.4.

Visual Information Extraction refers to the technology of extracting semantic entities and their relationships from many unstructured visually-rich documents. Visual information extraction differs in different document categories and the extracted entities are also different. Unlike traditional pure text information extraction, the construction of the document turns the text from a one-dimensional sequential arrangement into a two-dimensional spatial arrangement. This makes text information, visual information and layout information extremely important influencing factors in visual information extraction [53].

Document Visual Question Answering (VQA) is a high-level understanding task for document images. Specifically, given a document image and a related question, the model needs to give the correct answer to the question based on the given image [15]. A set of VQA tasks is defined based on various application scenarios, including statistical charts, daily-life photos and digital-born documents. Document VQA task aims to extract information from documents and answer natural language questions.

Document Image Classification is the process of analyzing and identifying document images, while classifying them into different categories such as scientific papers, resumes, invoices, receipts and many others. Document image classification is a special subtask of image classification, thus classification models for natural images can also address the problem of document image classification [53]. Document Image Classification task tries to predict the class which a document belongs to by means of analyzing its image representation.

For these four main Document AI tasks, there have been many open-sourced benchmark datasets in academia and industry, which has greatly promoted the development of new algorithms and models by researchers in related research areas. Several methods have been proposed to parse the layout of different documents, and they can be categorized into two major classes: traditional and deep learning-based. The next section introduces the different methods, including techniques based on heuristic rules, approaches based on machine learning, and deep learning to Document AI. However, the main focus of this work will be on approaches to document image classification tasks combining visual and textual features.

2.2 Document Classification

Document image classification consists in assigning a document image into one of a set of predefined document classes. In most research papers as well as their respective datasets, the methods focus on treating each single page, as a sample with a single class. Classification can be based on various features, such as visual, layout, or textual features. Classifiers solve various document classification problems, differ in how they use training data to construct models of document classes, and differ in their choice of document features and recognition algorithms. Choice of document features is an important step in classifier design [13].

Classification may be performed at different stages of document processing, with a diverse choice of document features, feature representations, class models and classification algorithms. These aspects are interrelated: design decisions made regarding one aspect have influence on design of other aspects. For example, if document features are represented in fixed-length feature vectors, then statistical models and classification algorithms are usually considered [13].

Some classifiers only use an image, structural, or textual features; others use a combination of resources from multiple groups. Global image features are extracted directly from the entire document image, and local features are extracted from a segmented image region. Structural features are obtained from physical or logical layout analysis. Textual features can be extracted from OCR output or directly from document images.

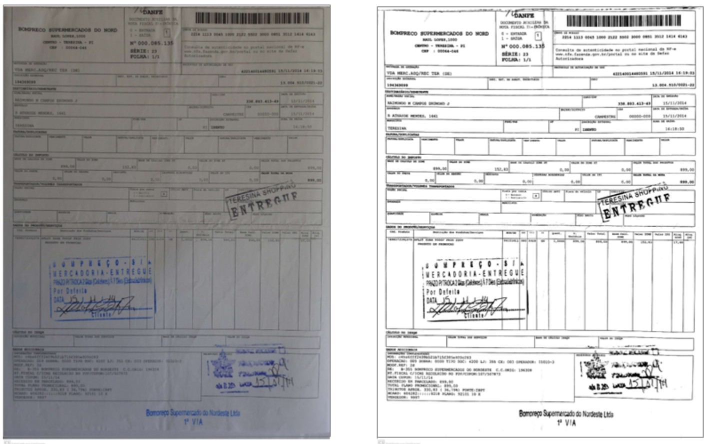
2.3 Processes of physical layout analysis

Physical layout analysis is the step that locates lines of text in the image and identifies its reading order and involves different processes. In document layout analysis step, an input document image is segmented into different regions. These regions are then classified as text or non-text. The non-text regions are further classified into different sub-classes like table, image, separator, graphic, chart, etc., whereas text regions are classified as title, paragraph, header, footer, caption, drop-capital, etc [9]. Most of the layout analysis systems use processes of binarization, noise removal, skew correction, page segmentation, zone classification and reading order determination in some form.

Binarization is an important first step in most document analysis systems. The goal of document binarization is to convert a given greyscale or color document image into a bi-level representation. When a document with black text on white background is scanned with a flatbed scanner to convert it to digital form, noise from several sources is added to its digital counterpart. This noise comes both from the imaging mechanisms like finite spatial sampling rate, noise in electronic components, pixel sensor sensitivity variations, and from the scanning process like de-focusing, non-uniform or poor illumination, and print-through from the other side of the page. Even if the original paper document was bi-level, the image obtained after scanning is greyscale. There are different types of binarization techniques like, Otsu, Adaptive, Sauvola, Global threshold based, etc. The result of running a binarization algorithm on a scanned document is shown in Figure 2.1.

Noise Removal is a process that tries to detect and remove noise pixels in a document that are introduced by scanning or binarization process.

Skew Correction is a process that detects and corrects the deviation of a document's orientation angle from the horizontal direction (see Fig. 2.2). Skew is introduced in a document image when a document is scanned or imaged at an angle concerning the reference axes. Paper positioning variations are a class of document degradations that results in skew and translation of the page contents in the scanned image. The problem of skew correction plays an important role in the effectiveness of many document analysis algorithms, such as text line estimation, region boundary detection, etc. For example, algorithms based on projection profiles assume an axis-aligned scan. The primary challenge

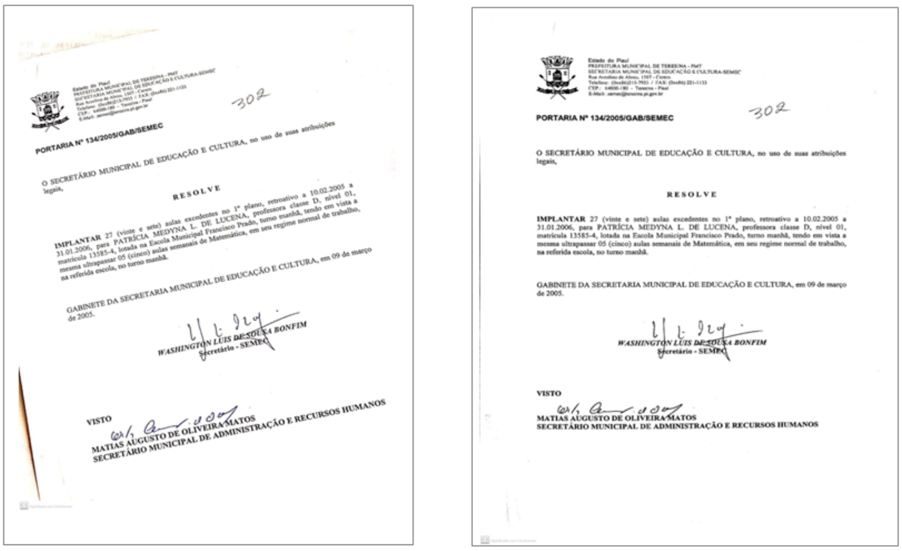


(a) Input Image

(b) Binarized Image

Figure 2.1: The result of applying binarization algorithm.(a) the input image is the scanned image of a document. (b) image of the document after the binarization process.

in skew correction is estimating the exact skew angle of a document image. A variety of techniques are used for the detection of skew. Most of them assume the presence of some text component in the document and estimate the orientation of text lines using different methods. A commonly used technique is projection profiles, in which a given image is rotated at different angles for a range. The maximum difference between the peaks of the pixel histogram of that image at each angle is calculated. The angle of rotation for skew correction will be the angle for which the maximum difference is obtained.

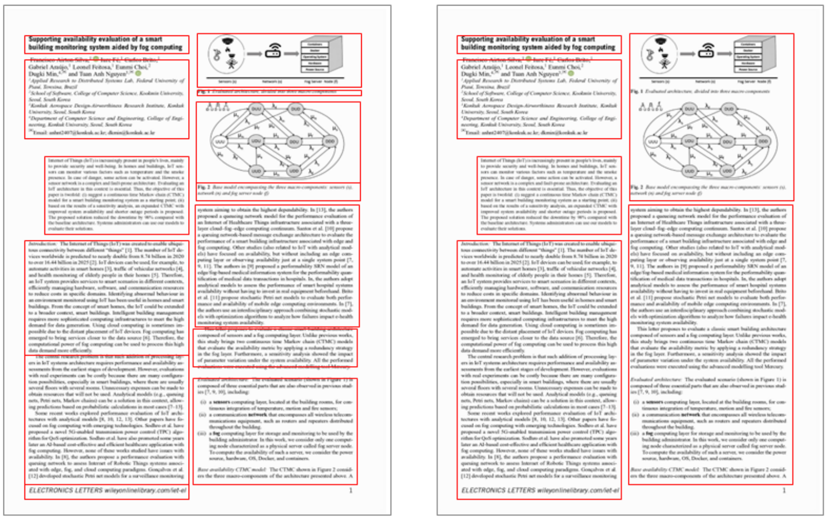


(a) Input Image

(b) Output Image

Figure 2.2: Example of a document image with a skew of 15 degrees.

Page Segmentation is a process that divides a document image into homogeneous zones, each consisting of only one physical layout structure (text, graphics, pictures, etc) while respecting the columnar structure of the document. The performance of OCR systems depends heavily on the page segmentation algorithm used. Page segmentation is a key component of geometric layout analysis. Given a document image, the goal of page segmentation is to perform a decomposition of the document image into smaller zones or segments as shown in Figure 2.3. The segments thus obtained are classified as containing text or non-text elements. The text segments or zones are then fed to a character recognition module to convert them into electronic format. If a page segmentation algorithm fails to correctly segment text from images, the character recognition module outputs a lot of garbage characters originating from the image parts. Additionally, if the document contains more than one text-column, the page segmentation algorithm should segment all text-columns separately so that the text-lines in different text-columns are not merged together.



Supporting availability evaluation of a smart building monitoring system aided by fog computing

ELECTRONICS LETTERS wileyonlinelibrary.com/iet-el

Supporting availability evaluation of a smart building monitoring system aided by fog computing

ELECTRONICS LETTERS wileyonlinelibrary.com/iet-el

(a) Segmentation A

(b) Segmentation B

Figure 2.3: Two different segmentations of the same document page.

Zone Classification aims at classifying the blocks detected by the page segmentation step of a geometric layout analysis system into one of a set of predefined classes (e.g. text, image, graphics, etc). Blocks identified as text can then be fed to a character recognition module. Similarly, other actions can be taken for zones of specific types; for instance graphics regions can be sent to a raster to vector conversion program, whereas table zones can be fed to a table understanding system.

Reading Order Determination tries to recover the order in which a human will go through different parts (segments) of the document.

Binarization, noise removal and skew correction are typically considered as pre-processing steps in layout analysis. The core part of geometric layout analysis consists of page segmentation and zone classification modules. Reading order determination is generally considered as a post-processing step in which simple ordering criterion can be used to identify the reading order of the detected page segments.

2.4 Rule-based Approaches

These approaches can be further divided into three types of analysis methods: top-down, bottom-up and hybrid. These methods rely heavily on heuristic rules and require many parameters to improve the performance. When the layout of a document is relatively complex, these methods may fail to deliver the optimal results.

Top-down: separates the original document into different regions and then use many heuristic filters to classify each region [31, 39]. The top-down approaches segment a page as a whole into one or more content blocks and recursively segment the segmented blocks into paragraphs, lines, words and character. Traditional top-down methods are only effective when the document has a Manhattan layout [1] [46]. While these methods work well in some documents, they require much human effort to discover better rules. These methods have a low generalization capability since they depend on the layout structure of the document represented in the input image. Furthermore, they depend highly on the parameters chosen based on a priori knowledge of the layout structure, which can vary greatly. In recent decades, documents have become more varied, having more complexity and not necessarily following those rules.

Bottom-up methods are more flexible as they do not require prior knowledge of the layout structure. Instead, they operate by processing an image from its lowest levels, such as its pixels or connected components, and increasingly group them into higher-level regions. The first group of connected components is produced by the black and white pixel in characters, then words, then lines, then text lines [32]. The document segmentation process combines them in blocks or paragraphs according to the different structural characteristics. Texture and geometric features, including spatial autocorrelation and Gabor filters, are the most common handcrafted features used in these approaches. However,

[1] Manhattan layouts are defined as layouts that can be decomposed into individual segments by vertical and horizontal cuts.

these methods use a lot of memory space and are time-consuming. They need higher computational costs as an exchange.

Hybrid Methods are created from the combination of the two basic approaches, and one of the most representative methods is Connected Components (CCs) analysis: CCs are detected from the entire images first, and then researchers analyze these CCs to acquire areas of interest [12, 44, 46, 47]. These algorithms mostly analyzed the connected components and the whitespaces between them. Hybrid methods can handle a variety of documents at a relatively fast speed. However, the results of these methods are still not convincing for problems, such as non-text identification.

These rule-based methods are mostly developed to perform document layout analysis. A DLA system primarily segments an input document image into various regions and classifies these as text or non-text region. The non-text regions are further classified into sub-classes like table, image, separator, graphic, and chart. In contrast, text regions are classified as title, paragraph, header, footer, caption, drop-capital, etc. [9].

2.5 Feature Engineering

A feature is a data transformation designed to make it easier to model. Feature engineering is the process of extracting features from raw data to enable the application of algorithms. It is crucial to the whole machine learning model and sometimes determines its performance's upper limit. Traditional machine learning methods (shallow learning) require features to be designed manually [50]. Therefore engineering feature-based approaches depend highly on feature identification, which largely depends on humans.

Feature engineering techniques are typically applied after gathering and cleaning the input data. In the cleaning step, one typically deals with missing values, errors, outliers and duplicates. Many feature engineering techniques exist, and it is not always clear which techniques fall under the definition of feature engineering and which do not [49]. In the feature selection step, redundant or unused features are removed, creating a subset of original features. Resource extraction reduces the dimension of the dataset creating new features, which can be linear combinations of the originals.

2.6 Machine Learning Approaches

Some researchers define Machine Learning as a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention. Algorithms and statistical frameworks help the system learn by itself and make predictions about certain functions. Image classification and text extraction are some of the applications of machine learning. Image classification is the process of feature extraction and pattern recognition from the images and classifying them.

Machine learning can be divided into supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning. In supervised learning, the corresponding outputs of the training data have been labeled. In contrast, the corresponding outputs of the training data in unsupervised learning are unlabeled. For semi-supervised learning, some training data are labeled, and the remaining data are unlabeled; the amount of unlabeled data often exceeds the number of labeled data. In reinforcement learning, reinforcement signals provided by the environment are used to evaluate the quality of the generated actions and improve the strategies for adapting to the environment.

Machine learning techniques create a predictor, such as a classifier or a regressor, through an inductive learning process. A classifier is created based on relationships between documents and associated labels in the document parsing task. Then the algorithm classifies a document not yet known in one of the categories learned in the training phase, making decisions based on experiences gained through previous successful problem-solving.

Several classic machine learning techniques, such as support vector machine (SVM) [17], K-Nearest Neighbor (KNN), Hidden Markov Model (HMM), Multilayer Perceptron (MLP) [41], Adaptive impulse decision tree (Adaboost) [27] and Artificial Neural Networks (ANN) [35], have been applied to linear classification. However, with the advent of deep learning models, every field of artificial intelligence has been affected, including text classification. These deep learning methods gained traction because they could model complex features without needing manual engineering by removing parts' domain knowledge requirements.

Artificial Neural Networks (ANNs) are inspired by brain studies and based on the operation of biological neural networks. They contain a series of mathematical equations that simulate biological systems processes such as learning and memory. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. ANNs learning process involves adjustments to the synaptic connections between the neurons. ANNs combine several artificial neurons to process information. Neural networks are trained to execute complex functions in various fields of application, including pattern recognition, identification, classification, clustering, speech, vision, and control systems. ANNs combine several artificial neurons to process information.

Artificial neurons essentially consist of 'inputs', which are multiplied by 'weights' and then computed by a mathematical function, which determines the 'activation' of the neuron, as depicted in Fig. 2.4 (a). Another function computes the 'output' of the artificial neuron, sometimes dependent on a certain 'threshold'. Weights can also be negative, so it can be said that the negative weight inhibits the signal. Depending on the weights, the computation of the neuron will be different. The weights are iteratively adjusted during the learning or training process until the output for specific inputs is close to the desired one.

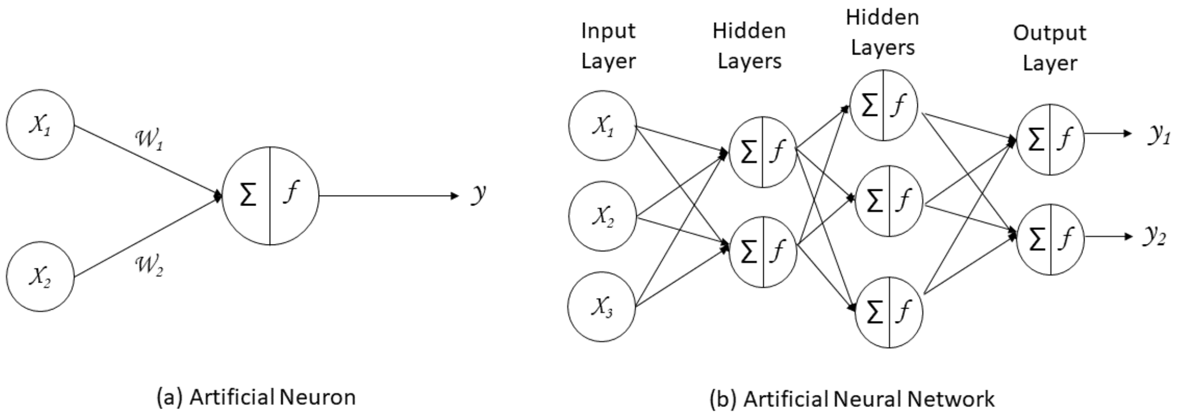


Figure 2.4: Illustration of an artificial neuron (a) and a single artificial neural network (b)

Figure 2.4 (b) shows an ANN, consisting of a layer of input and output nodes (neurons) connected by one or more layers of hidden nodes. Input layer nodes pass information to hidden layer nodes by firing activation functions, and hidden layer nodes fire or remain dormant depending on the evidence presented. The hidden layers apply weighting functions to the evidence, and when the value of a particular node or set of nodes in the hidden layer reaches some threshold, a value is passed to one or more nodes in the output layer.

2.7 Deep Learning Approaches

In recent years, deep learning methods have become a new paradigm for solving many machine learning problems. Deep Learning is a branch of machine learning that deals with deep neural networks, where each of the layers is trained to extract higher level representations of the previous ones. Deep learning methods have been confirmed to be effective in many research areas [57].

Specific layout approaches have been proposed in the literature where knowledge used to label zones in document images comes from geometric characteristics and physical appearance of the layouts that have already been seen by the model during training.

Existing approaches for document image classification and retrieval differ from each other based both on the type of extracted information (textual or visual) and/or the type of image analysis that is performed over the processed documents (global or local) [38].

2.7.1 Convolutional Neural Network

Convolutional Neural Networks (CNNs) are artificial neural networks that can be used to classify images, group them by similarity and perform object recognition within scenes and images. Convolutional Neural Networks (CNNs) are analogous to traditional ANNs in that they are comprised of neurons that self-optimise through learning. Each neuron will still receive an input and perform a operation (such as a scalar product followed by a non-linear function) - the basis of countless ANNs. From the input raw image vectors to the final output of the class score, the entire network will still express a single perceptive score function (the weight). The last layer will contain loss functions associated with the classes, and all of the regular tips and tricks developed for traditional ANNs still apply.

A Convolutional Neural Network (CNN) is a Deep Learning algorithm that can capture an input image, assign importance (learned weights and biases) to various aspects/objects of the image, and differentiate one from the other. These algorithms can identify faces, individuals, objects, characters, and many other aspects of visual data. Convolutional networks perform OCR to digitize text and make natural language processing possible in analog and handwritten documents, where images are symbols to be transcribed.

Recently, deep learning has been widely explored in document layout classification. A fast CNN based document layout analysis was introduced, where two one-dimensional projection of images were considered to train the model. To identify complex document layouts, a CNN architecture that learns a hierarchy of features from a raw image was proposed for the document image classification. A Deep CNN architecture was applied for classification, where CNNs were extensively used for both feature extraction and model training process.

2.7.2 Recurrent Neural Networks

A Recurrent Neural Network (RNN) is a special artificial neural network adapted to work for time series data or data that involves sequences of data such as text. These Neural Networks have been applied to several problems, such as language translation, NLP, speech recognition, genomes and numerical series. RNNs have the concept of 'memory' that helps them store the states or information of previous inputs to generate the next output of the sequence. The decision of a recurrent step reached in time step 1 affects the decision to reach a later time. Thus, recurrent networks have two sources of input, the present and the recent past, which combine to determine the appearance of new data, as shown in Fig. 2.5.

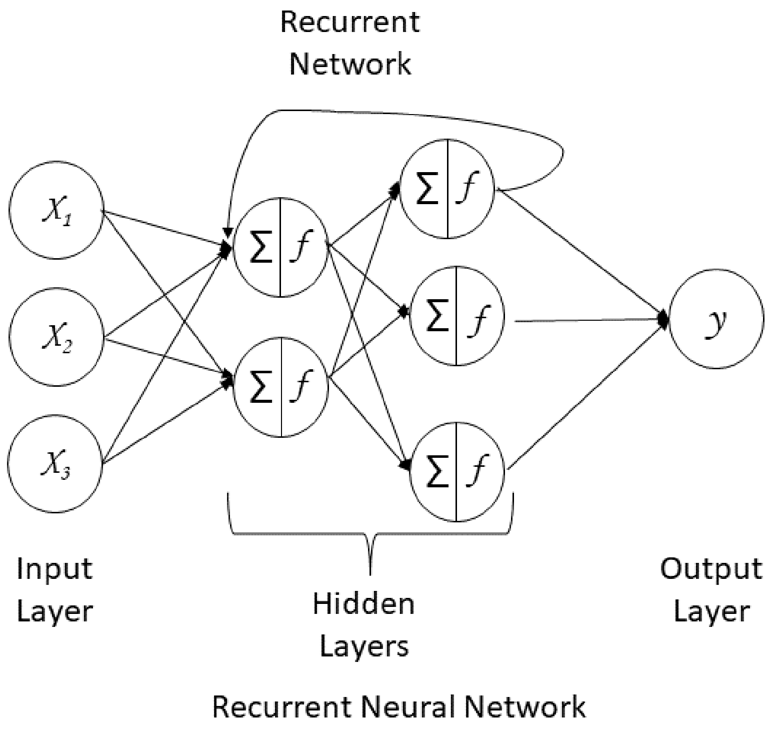


Figure 2.5: Illustration of a Recurrent Neural Network.

Recurrent Neural Networks leverage the backpropagation through time (BPTT) algorithm to determine the gradients. BPTT is slightly different from traditional backpropagation as it is specific to sequence data. It also differs from the traditional approach in that BPTT sums errors at each time step, whereas feedforward networks do not need to sum errors as they do not share parameters across each layer. The principles of BPTT are the same as traditional backpropagation, where the model trains itself by calculating errors from its output layer to its input layer. These calculations allow us to adjust and fit the model's parameters appropriately.

2.7.3 Long Short Term Memory networks

Long Short Term Memory networks (LSTMs) [45] are a special kind of RNN, capable of learning long-term dependencies. They work tremendously well on many problems and are now widely used in NLP. LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior. The basic difference between the architectures of RNNs and LSTMs is that the hidden layer of LSTM is a gated unit or gated cell. It consists of four layers that interact with one another to produce the output of that cell along with the cell state. These two things are then passed onto the next hidden layer.

The LSTM consists of three parts, as shown in Fig. 2.6, and each part performs an individual function. At a high level, LSTM works very much like an RNN cell. LSTM cells possess three gates, an input, a forget and an output gate, that allow changes on a cell state vector propagated iteratively to capture long-term dependencies. This controlled information flow within the cell enables the network to memorize multiple time dependencies with different characteristics. LSTM is mainly used for the modeling of long-term dependencies. LSTM provides a mechanism that limits the change gradient realized at each iteration. Hence, LSTM does not allow past information to be completely discarded.

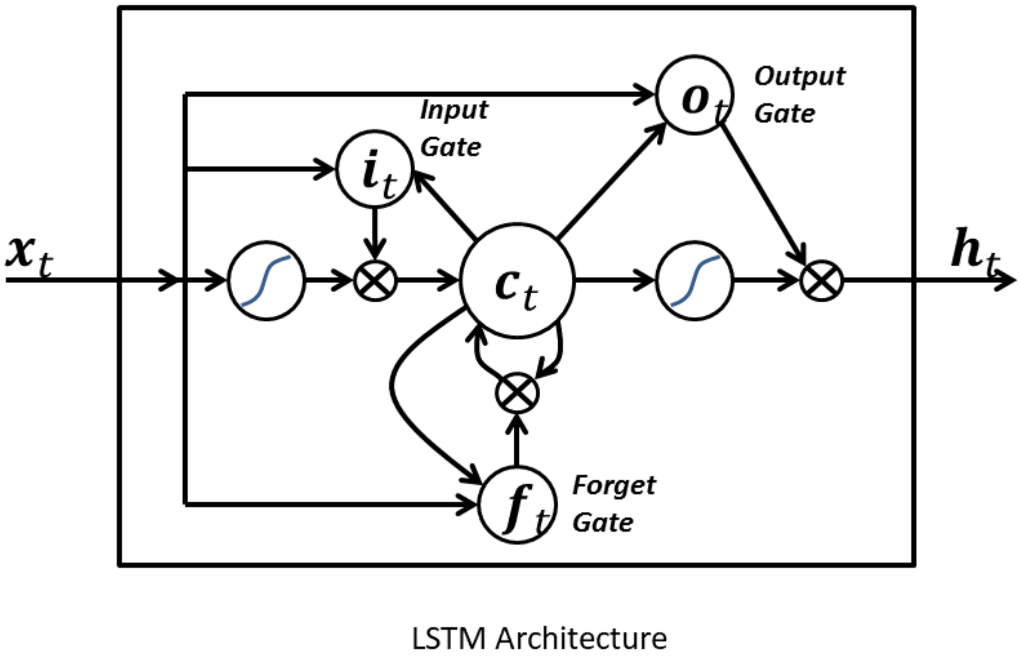


Figure 2.6: Illustration of the main elements of the architecture of the cell of a Long Short Term Memory network.

2.7.4 Transformer

An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. The vanilla transformer [48] is the first transduction model relying entirely on an attention mechanism without using sequence-aligned RNNs or convolution to draw global dependencies between input and output. The original Transformer model follows the overall architecture of Figure 2.7 using stacked self-attention and point-wise, with six layers. The output of layer l is the input of layer l+1 until the final prediction is reached.

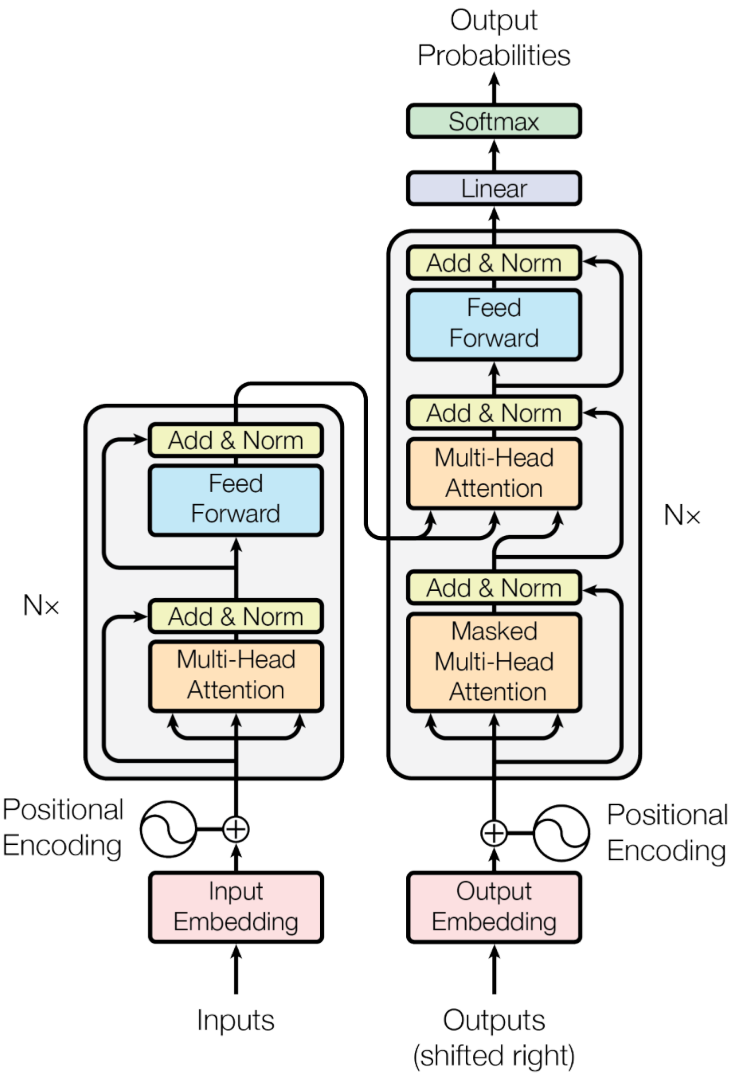


Figure 2.7: Vanilla Transformer Model Architecture [40, 48]. On the left, there is an N = 6 layers encoder stack. The inputs enter the encoder side of the Transformer through an attention sub-layer and FeedForward Network (FFN) sub-layer. On the right, there is an N = 6 layers decoder stack. The target outputs go into the decoder side of the Transformer through two attention sub-layers and an FFN sub-layer.

The encoder is composed of a stack of N = 6 identical layers. Each encoder layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position-wise fully connected feed-forward network. A residual connection surrounds each main sub-layer in the Transformer model. These connections transport the unprocessed input of a sub-layer to a layer normalization function. This way, we are certain that key information such as positional encoding is not lost on the way [40].

The decoder layer structure remains the same as the encoder layer for all N = 6 layers of the Transformer model. Each layer contains three sub-layers: a multi-headed masked attention mechanism, a multi-headed attention mechanism, and a fully connected position-wise feed-forward network. The decoder has a third main sub-layer, the masked multi-head attention mechanism. In this sublayer output, the following words are masked at a certain position, so Transformer bases its assumptions on its inferences without seeing the rest of the sequence. The Transformer only performs a small, constant number of steps (chosen empirically). Each step applies a self-attention mechanism [2] that directly models relationships between all words in a sentence, regardless of their respective position.

In recent years, self-attention-based models like Transformers and BERT [16] have achieved state-of-the-art performance on several Natural Language Processing tasks. The BERT model, Bidirectional Encoder Representations from Transformers, is an attention-based bidirectional language modeling approach. It is designed to pre-train deep bidirectional representations from the unlabeled text by jointly conditioning on both left and right contexts in all layers. The overall framework of BERT is a multi-layer bidirectional Transformer encoder as shown in Fig. 2.8. It accepts a sequence of tokens and stacks multiple layers to produce final representations.

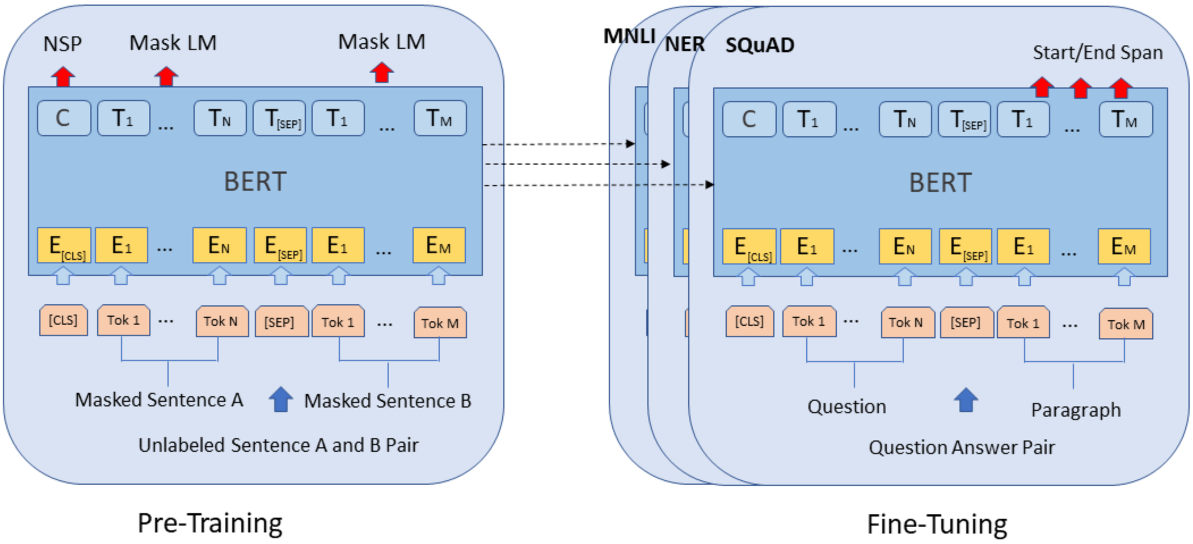


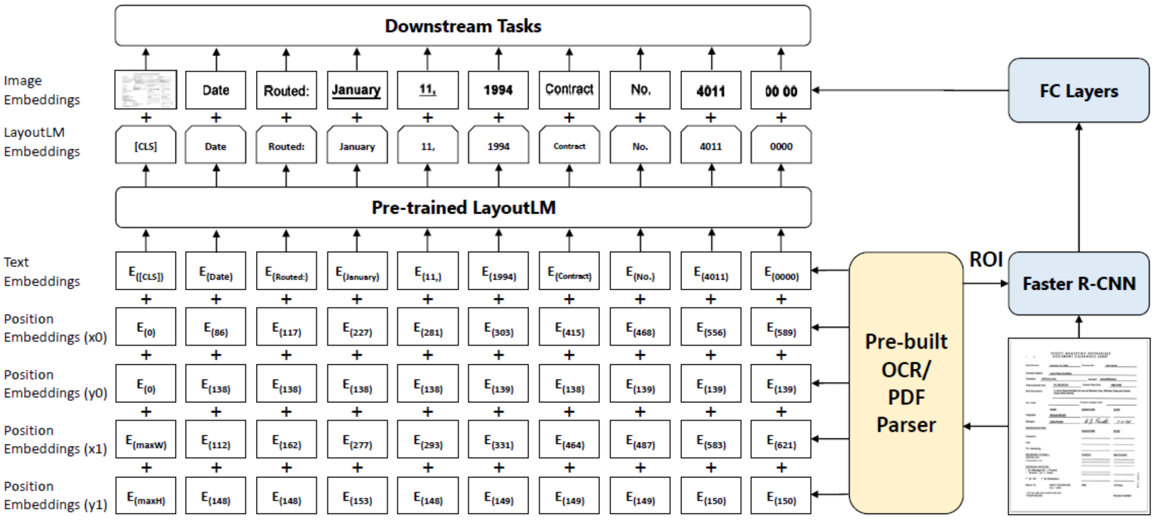
Figure 2.8: The overall framework of BERT adapted from Devlin et al. (2019) [16]. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different downstream tasks. During fine-tuning, all parameters are fine-tuned. [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token.

There are two steps in the BERT [16] framework: pre-training and fine-tuning. During the pre-training, the model uses two objectives to learn the language representation:

[2] Self-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence

Masked Language Modeling (MLM) and Next Sentence Prediction (NSP), where MLM randomly masks some input tokens, and the objective is to recover these masked tokens, and NSP is a binary classification task taking a pair of sentences as inputs and classifying whether they are two consecutive sentences, see in section 2.7.5. In fine-tuning, task-specific datasets are used to update all parameters end-to-end. The BERT [16] model has been successfully applied in a set of NLP tasks [53].

LayoutLM [53] model is proposed as the pioneer pre-training method of text and layout for document image understanding tasks, which expands 1D positional encoding of BERT to 2D to avoid the loss of layout information. It is trained over a large corpus of business documents to understand spatial dependencies between text blocks. Image embeddings are combined in the fine-tuning stage, and the image information is integrated into the pre-training stage. The overall framework of LayoutLM is shown in Fig. 2.9. In addition, it adopted a multi-task learning objective for LayoutLM, including a Masked Visual-Language Model (MVLM) loss and a Multi-label Document Classification (MDC) loss, which are discussed in subsection 2.7.5. They add the 2-D position embedding layers with four embedding representations (x0, y0, x1, y1), where (x0, y0) corresponds to the position of the upper left in the bounding box, and (x1, y1) represents the position of the lower right. They also add four position embedding layers with two embedding tables, where the embedding layers representing the same dimension share the same embedding table. This means that they look up the position embedding of x0 and x1 in the embedding table X and look up y0 and y1 in table Y.



Downstream Tasks
Image Embeddings
Date
Routed:
January
11,
1994
Contract
No.
4011
00 00
LayoutLM Embeddings
[CLS]
Date
Routed:
January
11,
1994
Contract
No.
4011
0000
Pre-trained LayoutLM
Text Embeddings
E([CLS]) E(Date) E(Routed:) E(January) E(11,) E(1994) E(Contract) E(No.) E(4011) E(0000)
Position Embeddings (x0)
E(0) E(86) E(117) E(227) E(281) E(303) E(415) E(468) E(556) E(589)
Position Embeddings (y0)
E(0) E(138) E(138) E(138) E(138) E(139) E(138) E(139) E(139) E(139)
Position Embeddings (x1)
E(maxW) E(112) E(162) E(277) E(293) E(331) E(464) E(487) E(583) E(621)
Position Embeddings (y1)
E(maxH) E(148) E(148) E(153) E(148) E(149) E(149) E(149) E(150) E(150)
+
FC Layers
ROI
Faster R-CNN
Pre-built OCR/ PDF Parser

Figure 2.9: The overall framework of LayoutLM [53], where 2-D layout and image embeddings are integrated into the original BERT architecture. The LayoutLM embeddings and image embeddings from Faster R-CNN work together for downstream tasks.

Source: reproduced from Xu et al. (2020) [53] (2020) [3]

To align the image feature of a document with the text, they add an image embedding layer to represent image features in language representation. With the bounding box of each word from OCR results, they split the image into several pieces and one-to-one correspondence with the words. They generate the image region features with these pieces of images from the Faster R-CNN model as the token image embeddings. For the [CLS] token, they also use the Faster R-CNN model to produce embeddings using the whole scanned document image as the Region of Interest (ROI) to benefit the downstream tasks which need the representation of the [CLS] token [53].

2.7.5 Pretraining Objectives

Inspired by BERT, many pre-trained language models have emerged for various tasks in understanding visually rich documents. These models use pre-training jointly different modalities such as text, layout, and visual information in a single framework. Pre-training objectives have been used in pre-training and fine-tuning language models.

Masked Language Model (MLM) was proposed firstly in BERT [16] architecture to learn bidirectional representations by predicting the original vocabulary id of a randomly masked word token based on its context. MLM objective allows the representation to fuse the left and the right context, which allows pre-training for a deep bidirectional Transformer. Some percentage of the input tokens at random are masked to train a deep bidirectional representation. In this case, the final hidden vectors corresponding to the mask tokens are fed into an output softmax over the vocabulary, as in a standard Language Model. BERT randomly masks 15% of all WordPiece tokens with a special token [MASK] in each sequence and only predicts the masked words rather than reconstructing the entire input. The training data generator randomly chooses 15% of the token positions for prediction. Masked tokens are replaced with a special [MASK] token 80% of the time, a random word 10%, and an unaltered 10%. Figure 2.10 (a) shows that MLM is a fill-in-the-blank task, words are masked from the input, and the transformer network must predict the missing words. The BERT model is then trained to reconstruct these masked tokens given the observed set.

Next Sentence Prediction (NSP) enables the model to capture sentence-to-sentence relationships, which are crucial in many language modelling tasks such as Question Answering and Natural Language Inference. Given a pair of sentences, the model predicts a binary label, i.e., whether the pair is valid from the original document or not, see Fig. 2.10 (b). Specifically, when choosing the sentences A and B for each pre-training example,

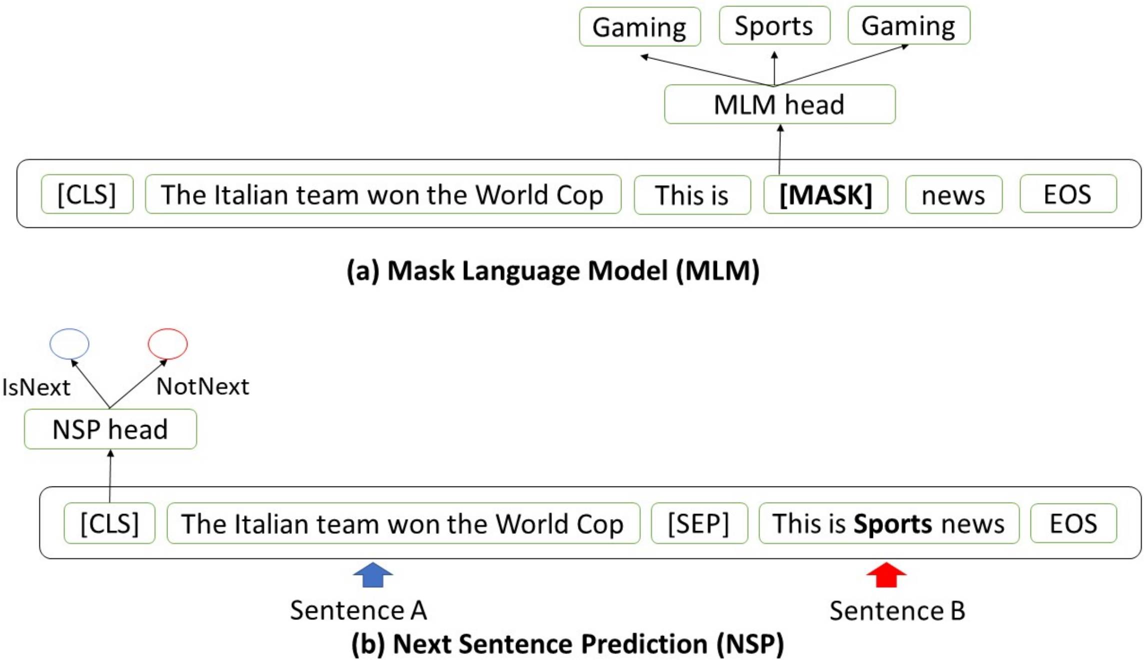

Gaming
Sports
Gaming
MLM head
[CLS]
The Italian team won the World Cop
This is
[MASK]
news
EOS
(a) Mask Language Model (MLM)
IsNext
NotNext
NSP head
[CLS]
The Italian team won the World Cop
[SEP]
This is Sports news
EOS
Sentence A
Sentence B
(b) Next Sentence Prediction (NSP)

Figure 2.10: BERT [16] (a) Masked Language Model and (b) Next Sentence Prediction objectives. BERT operates over sequences of discrete tokens comprised of vocabulary words and a small set of special tokens: [CLS], [MASK] and [SEP]. The first token of every sequence is always a special classification token ([CLS]). To mask a word that will be predicted, the special token ([MASK]) is used. ([SEP]) is a special separator token.

50% of the time B is the actual next sentence that follows A, and 50% of the time, it is a random sentence from the corpus.

Masked Visual-Language Model (MVLM) was proposed to learn language representation with the clues of 2-D position embeddings and text embeddings. The model randomly masks some input tokens during pre-training but keeps the 2-D position embeddings and other text embeddings. The model is then trained to predict the masked tokens given the context. In this way, the LayoutLM [53] model not only understands the language contexts but also utilizes the corresponding 2-D position information, thereby bridging the gap between the visual and language modalities.

Multi-label Document Classification (MDC) refers to assigning multiple relevant labels to each input document, while the entire label set might be extremely large. LayoutLM [53] uses MDC loss during the pretraining phase. Given a set of scanned documents, the model uses the document tags to supervise the pretraining process. The model can cluster the knowledge from different domains and generate better document-level representation [15].

Text-Image Alignment (TIA) was proposed in LayoutLMv2 [54] as a fine-grained cross-modality alignment task to help the model learn the spatial location correspondence between the image and coordinates of the bounding boxes. The covering operation randomly selects some tokens lines and their image regions and covers them in the document image. During pretraining, a classification layer is built above the encoder outputs. This layer predicts a label for each text token depending on whether it is covered, i.e., [Covered] or [Not Covered], and computes the binary cross-entropy loss.

Text-Image Matching (TIM) task is applied to help the model learn image-text alignment, i.e., to the model learn the correspondence between document image and textual content. LayoutLMv2 [54] feeds the output representation at tag [CLS] into a classifier to predict whether the image and text are from the same document page. Regular inputs are positive samples. Moreover, in negative samples, an image is either replaced by a page image from another document or dropped. The TIM target labels are set to tag [Covered] in negative samples.

2.8 Other Recent Works

In this section, we present a brief review focused on the problem of document classification methods that take textual and visual information as input.

Asim et al. (2019) [5] present a Naïve Deep Learning approach for the task of text document image classification, which utilizes both structural similarity and content of text document images. A filter-based feature-ranking algorithm was utilized to alleviate the dependency of textual stream on the performance of underlying OCR. This algorithm ranks the features of each class based on their ability to discriminate document images and selects a set of top 'K' features that are retained for further processing. Simultaneously, the visual stream uses deep CNN models to extract structural features of document images, and the average ensembling method concatenates textual and visual streams.

Aggarwal et al. (2020) [1] proposed a hierarchical multi-modal bottom-up approach to detect larger constructs in a form page. Specifically for the task of extracting higher-order constructs from lower-level elements. They process textual and spatial representation of candidates sequentially through a BiLSTM to obtain context-aware representations and fuse them with image patch features obtained by processing it through a CNN. Subsequently, the sequential decoder takes this fused feature vector to predict the association type between reference and candidates using a LSTM based Sequential Association Module (SAM). However, this method shows insufficient capabilities in layout modeling.

A multimodal neural network is designed by Audebert et al. (2020) [6], which is able to learn from word embeddings and images. FastText word embedding and MobileNetv2 image embedding were introduced to perform visual and textual feature extraction jointly. To perform a fine-grained classification using visual and textual features, first it was used Tesseract OCR to extract the text from the image. Then, they computed character-based word embeddings using FastText on the noisy Tesseract output and generate a document embedding which represents our text features. The visual features are learned using MobileNetv2, a standard CNN from the state of the art. Finally, they introduced an end-to-end learnable multimodal deep network that jointly learns text and image features and perform the final classification based on a fused heterogeneous representation of the document.

Bakkali et al. (2020) [8] presented a hybrid cross-modal feature learning approach that combines image features and text embedding to classify document images. First, NASNet-Large model and BERT model pretrained were used on ImageNet to extract the image and textual features, respectively, for document classification on the Tobacco-3482 dataset.

Wiedemann and Heyer (2021) [51] developed an approach based on convolutional neural networks (CNN) combining image and text features to perform page stream segmentation (PSS) as a binary classification task on single pages from a data stream. They first create two separate convolutional neural networks for binary classification of pages into either SD or ND, one based on text data and another based on image scans. In a third step, they combine the learned parameters from the two final hidden layers of both CNN to an input vector of features for a multi-layer perceptron. This MLP delivers a third and final classification result based on both feature types. The work of Braz et al. (2021) [10] is built over the proposal of Wiedemann and Heyer [51] by improving the network architecture using EfficientNet pre-trained CNN architecture, replacing the earlier proposed VGG16 Network. However, they used techniques focused only on the image of the pages. They proposed a novel approach to the PSS problem, using four training classes which can be reduced to the usual two classes of the PSS problem in the literature.

Li et al. (2021) [30] proposed the VTLayout model for document layout analysis task to locate and identify different category blocks by merging the documents deep visual, shallow visual, and text features. VTLayout consists of two stages, Category Block Localization and Category Block Classification. The Category Block Localization stage localizes the different categories from documents using the Cascade Mask R-CNN model. The Deep Visual Feature Extractor (DVFE), Shallow Visual Feature Extractor(SVFE), and Text Feature Extractor (TFE) have been built to extract different features in the Category Block Classification stage. The DVFE is built with the MobileNetV2 model to extract the deep visual feature from all the category blocks. The SVFE extracts the shallow visual feature based on the statistical pixels of different category blocks. The TFE is implemented with the TF-IDF feature extraction technique to extract the text features from the category blocks.

BROS [21] encodes relative positions of texts between text blocks in 2D space, focusing on the combinations of texts and their spatial information without relying on visual features for effective key information extraction from documents. Specifically, it is a spatial encoding method that utilizes relative positions between text blocks. In addition to the Masked Visual-Language Modeling (MVLM), BROS proposes an area-masked language model (AMLM), which masks all text blocks in a randomly selected document area and supervises the masked texts.

StructuralLM [29] is a self-supervised pretraining method designed to better model the interactions of cells and layout information in scanned document images. Unlike LayoutLM [53], StructuralLM is a structural pretraining approach that jointly exploits cell and layout information from scanned documents. It uses cell-level 2D-position embeddings to model the layout information of cells rather than word-level 2D-position embeddings. It adopts two self-supervised tasks during the pretraining stage: MVLM [53] and Cell Position Classification (CPC) task.

DocFormer [2] adopts a discrete multi-modal structure self-attention with shared spatial embeddings in an encoder-only transformer architecture. It also has a CNN backbone for visual feature extraction and encoding image information to obtain higher resolution image features and simultaneously encodes text information into text embeddings. The position information is added to the image and text information separately and passed to the Transformer layer separately. In addition, DocFormer proposes three pretraining tasks: multi-modal masked language modeling (MM-MLM), a modification of the original MLM pre-text task introduced in BERT; learning-to-reconstruct (LTR), is an image reconstruction task, and the text describes image (TDI) to teach the network if a given piece of text describes a document image.

LAMPreT was proposed by Wu et al. (2021) [52] to explore both the structure and the content of documents and consider image content to learn a multi-modal document representation. LAMPreT provides the model with more visual information to model web documents, such as font size, illustrations, etc., which helps to understand rich web data. LAMPreT framework is hierarchical, consisting of two cascaded transformers [48]. The lower-level model is trained with MLM [16] and TIM [53] objectives, while the higher-level model is trained with three block-level pretraining objectives aiming to exploit the structure of a document: block-ordering prediction, masked-block predictions, and image fitting predictions. LAMPreT was evaluated on two downstream tasks: text block filling and image suggestion.

Xu et al. (2021) [54] proposed the spatial-aware self-attention mechanism for the LayoutLMv2, which involves a 2-D relative position representation for token pairs. Different from the absolute 2-D position embeddings, the relative position embeddings explicitly provide a broader view for contextual spatial modeling. The multi-modal Transformer accepts inputs of three modalities: text, image, and layout. The input of each modality is converted to an embedding sequence and fused by the encoder. The model establishes deep interactions within and between modalities by leveraging the powerful Transformer layers. They adopted three self-supervised tasks simultaneously during the pre-training stage: Masked Visual-Language Model (MVLM), Text-Image Alignment (TIA) and Text-Image Matching (TIM).

Table 2.1 summarizes the works presented in this section proposed for AI document problems. The most recent works presented are pre-training multimodal models and used transformer architecture based on BERT as the backbone. Each model combines at least two modalities (textual, visual, and layout) for downstream tasks, except for the model proposed by Braz et al. (2021) [10] that used only visual features for PSS. The most used datasets for classification tasks are RVL-CDIP and Tobacco-3482 described in Chapter 3.

2.9 Summary

This chapter examined the Document Intelligence problem and its practical applications. *Document Intelligence* refers to the techniques for automatically reading, understanding, and analyzing documents. Understanding these documents becomes challenging due to the variety of layouts, poor quality scans and OCR, a complex structure composed of multi-columns, different tables, texts, and images. The main points presented are:

- The definition and emergence of Document Intelligence at the Conference on Neural Information Processing Systems.
- Document AI application in various downstream tasks, including document layout analysis, visual information extraction, document visual question answering, and document image classification.
- Document image classification task and its approaches based on textual, visual, and layout modalities or a combination of them.
- Recent works are based on machine and deep learning approaches.

Several studies have addressed document analysis using visual and textual resource extraction for downstream tasks. Approaches have evolved from early-stage heuristic rules to statistical machine learning. Then, deep learning methods with greater attention to the pre-trained language models based on BERT [16] have become a trend in Document AI development. Moreover, some models have designed richer pretraining objective tasks for different modalities, such as the MLM objective task introduced by LayoutLM [53]. A major drawback of such pre-trained models based on the Transformer architecture [48] is that they require a high computational cost. Unlike these previous methods, our approach aims to improve the performance of language models by combining texts and their spatial information with a low computational cost. Specifically, we propose a spatial layout encoding method, which combines textual and spatial information from text blocks.

Table 2.1: Document AI: models, modality, backbone, datasets and pre-train objective tasks. T, L and I denote textual, layout and image features, respectively. (*) denotes that pre-train objective tasks are not applied in the approach.

Models	Modality	Backbone	Datasets	Pre-train tasks
Asim et al. (2019) [5]	T + I	InceptionV3 Multi-channel CNN	Tobacco-3482 RVL-CDIP	‘k’ features
Aggarwal et al. (2020) [1]	T + I	BiLSTM	MMPAN-forms	SAM
Audebert et al. (2020) [6]	T + L	Multimodal Neural Network	Tobacco-3482 RVL-CDIP	(*)
Bakkali et al. (2020) [8]	T + I	Cross-modal BERT	Tobacco-3482	MLM NSP
LayoutLM (2020) [53]	T + L	Transformer BERT	FUNSD SROIE RVL-CDIP	MVLM MDC
Wiedemann and Heyer (2021) [51]	T + I	CNN MLP	Tobacco800 German dataset	(*)
Braz et al. (2021) [10]	I only	CNN EfficientNet	Tobacco800 AI.Lab.Splitter	(*)
BROS (2021) [21]	T + L	Tansformer BERT	FUNSD SROIE CORD SciTSR	MVLM AMLM
StructuralLM (2021) [29]	T + L	BERT	FUNSD RVL-CDIP	MVLM CPC
DocFormer (2021) [2]	T + L + I	Multimodal Transformer	FUNSD CORD RVL-CDIP Kleister-NDA	MM-MLM LTR TDI
LAMPreT (2021) [52]	T + L + I	Transformer BERT	Wikipages	MVLM TIM
VTLayout (2021) [30]	T + L + I	Cascade Mask R-CNN MobileNetV2	PubLaynet	DVFE SVFE TFE
LayoutLMv2 (2021)[54]	T + L + I	Transformer	FUNSD SROIE CORD Kleister-NDA RVL-CDIP DocVQA	MVLM TIA TIM

Chapter 3

Datasets

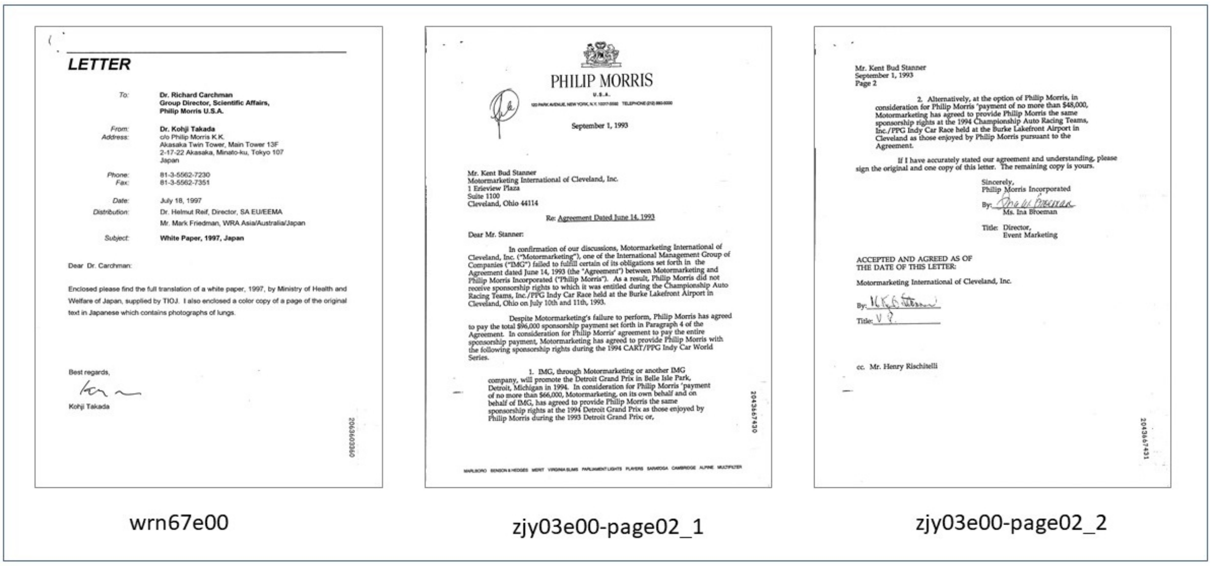
This Chapter deals with the datasets that should be used in experiments to evaluate our proposal. Publicly accessible document image collection with realistic scope and complexity is important to the document image analysis and search community.

Truth Tobacco Industry Documents, formerly known as Legacy Tobacco Documents Library (LTDL), created and hosted by the University of California San Francisco (UCSF). It was built to provide permanent access to the tobacco industry's internal corporate documents produced during litigation between the US States, the seven major tobacco industry organizations, and other sources. Complex document image processing (CDIP) test collection was constructed by the Illinois Institute of Technology (IIT), assembled from 42 million documents (in 7 million multi-page TIFF images) released by tobacco companies under the Master Settlement Agreement from the LTDL in 2006 [28]. The documents in LTDL range from the late 19th century to the present. The bulk of the collections dated 1950 through 2003.

At First, we we out of three publicly available datasets containing business documents in English namely Tobacco800 [56, 55], RVL-CDIP [19] and Tobacco-3482 [26] datasets. These datasets are subsets of the CDIP dataset found in the literature for various downstream tasks, such as document image classification, PSS and offline signature verification, among others. Next, we briefly describe VICTOR [33, 4], a dataset of court documents in Portuguese proposed for document classification. Finally, we deal with the importance and growth of data on the Web in commercial transactions and datasets of HTML pages. The properties of all datasets are described below.

3.1 Tobacco800

Tobacco800 is a pretty dataset used for several tasks, namely, offline signature verification, detection and extraction of document images, etc. Recently, it has been used for page stream segmentation. Tobacco800 is a public subset of the CDIP. The Tobacco800 dataset has only 1,290 document images of many types, such as letters, fax, memos, etc., that were collected and scanned using various equipment over time. Since the Tobacco800 dataset sample file name comes with the page, like the ones shown in Figure 3.1, when merged, it mimics a stream of pages from multiple documents ideal to split by the PSS model. In addition, Tobacco800 was manually annotated, targeting document signature and logos segmentation.



LETTER

To: **Dr. Richard Carchman**
Group Director, Scientific Affairs,
Philip Morris U.S.A.

From: **Dr. Kohji Takada**
Address: c/o Philip Morris K.K.
Akasaka Twin Tower, Main Tower 13F
2-17-22 Akasaka, Minato-ku, Tokyo 107
Japan

Phone: 81-3-5562-7230
Fax: 81-3-5562-7351

Date: July 18, 1997
Distribution: Dr. Helmut Reif, Director, SA EU/EEMA
Mr. Mark Friedman, WRA Asia/Australia/Japan

Subject: **White Paper, 1997, Japan**

Dear Dr. Carchman:

Enclosed please find the full translation of a white paper, 1997, by Ministry of Health and Welfare of Japan, supplied by TIOJ. I also enclosed a color copy of a page of the original text in Japanese which contains photographs of lungs.

Best regards,

Kohji Takada

2063603360

wrn67e00

PHILIP MORRIS
U.S.A.
120 PARK AVENUE, NEW YORK, N.Y. 10017-5592 TELEPHONE (212) 880-5000

September 1, 1993

Mr. Kent Bud Stanner
Motormarketing International of Cleveland, Inc.
1 Erieview Plaza
Suite 1100
Cleveland, Ohio 44114

Re: Agreement Dated June 14, 1993

Dear Mr. Stanner:

In confirmation of our discussions, Motormarketing International of Cleveland, Inc. ("Motormarketing"), one of the International Management Group of Companies ("IMG") failed to fulfill certain of its obligations set forth in the Agreement dated June 14, 1993 (the "Agreement") between Motormarketing and Philip Morris Incorporated ("Philip Morris"). As a result, Philip Morris did not receive sponsorship rights to which it was entitled during the Championship Auto Racing Teams, Inc./PPG Indy Car Race held at the Burke Lakefront Airport in Cleveland, Ohio on July 10th and 11th, 1993.

Despite Motormarketing's failure to perform, Philip Morris has agreed to pay the total $96,000 sponsorship payment set forth in Paragraph 4 of the Agreement. In consideration for Philip Morris' agreement to pay the entire sponsorship payment, Motormarketing has agreed to provide Philip Morris with the following sponsorship rights during the 1994 CART/PPG Indy Car World Series.

1. IMG, through Motormarketing or another IMG company, will promote the Detroit Grand Prix in Belle Isle Park, Detroit, Michigan in 1994. In consideration for Philip Morris 'payment of no more than $66,000, Motormarketing, on its own behalf and on behalf of IMG, has agreed to provide Philip Morris the same sponsorship rights at the 1994 Detroit Grand Prix as those enjoyed by Philip Morris during the 1993 Detroit Grand Prix; or,

2043667430

MARLBORO BENSON & HEDGES MERIT VIRGINIA SLIMS PARLIAMENT LIGHTS PLAYERS SARATOGA CAMBRIDGE ALPINE MULTIFILTER

zjy03e00-page02_1

Mr. Kent Bud Stanner
September 1, 1993
Page 2

2. Alternatively, at the option of Philip Morris, in consideration for Philip Morris 'payment of no more than $48,000, Motormarketing has agreed to provide Philip Morris the same sponsorship rights at the 1994 Championship Auto Racing Teams, Inc./PPG Indy Car Race held at the Burke Lakefront Airport in Cleveland as those enjoyed by Philip Morris pursuant to the Agreement.

If I have accurately stated our agreement and understanding, please sign the original and one copy of this letter. The remaining copy is yours.

Sincerely,
Philip Morris Incorporated

By: Ina W. Broeman
Ms. Ina Broeman

Title: Director,
Event Marketing

ACCEPTED AND AGREED AS OF
THE DATE OF THIS LETTER:

Motormarketing International of Cleveland, Inc.

By: K. K. B. Stanner
Title: V. P.

cc. Mr. Henry Rischitelli

2043667431

zjy03e00-page02_2

Figure 3.1: Image documents sample of Tobacco800 dataset. In left-to-right order, the first image is a single-page document, the next two images are pages of the same document and are in ascending page order.

A significant percentage of Tobacco800 are consecutively numbered multi-page business documents, making it a valuable testbed for various content-based document image retrieval approaches. Resolutions of documents in Tobacco800 vary significantly from 150 to 300 DPI and the dimensions of images range from 1200 by 1600 to 2500 by 3200 pixels.

The classification problem here involves two classes: whether the transition between consecutive pages indicates the continuity of the same document or the beginning of a new document. Document images are classified in FirstPage or NextPage, which FirstPage represents a the first page of a document and NextPage class is formed by all pages of a document, except the first page. The Tobacco800 Dataset was used by Wiedemann and Heyer (2021) [51] to evaluate a binary classification architecture proposed by them. This work developed a hybrid approach combining image and text for page flow segmentation (PSS) task. Braz et al. (2021) [10] also used this dataset to evaluate a series of models for the PSS problem. They defined a novel approach to the PSS problem using four training classes which can be reduced to the usual two classes of the PSS problem in the literature.

3.2 RVL-CDIP

RVL-CDIP, also known as BigTobacco, stands for Ryerson Vision Lab Complex Document Information Processing. The file structure of this dataset is the same as the IIT collection, so you can query this dataset for OCR and additional metadata. RVL-CDIP is a huge dataset with 400,000 grayscale images in 16 classes, with 25,000 images per class, which was introduced by Harley et al. (2015) [19]. There are 320,000 training images, 40,000 validation images, and 40,000 test images. The images are resized, so their largest dimension does not exceed 1,000 pixels. The 16 classes include letter, form, email, handwritten, advertisement, scientific report, scientific publication, specification, file folder, news article, budget, invoice, presentation, questionnaire, resume, memo, see Figure 3.2. The evaluation metric is the overall classification accuracy.

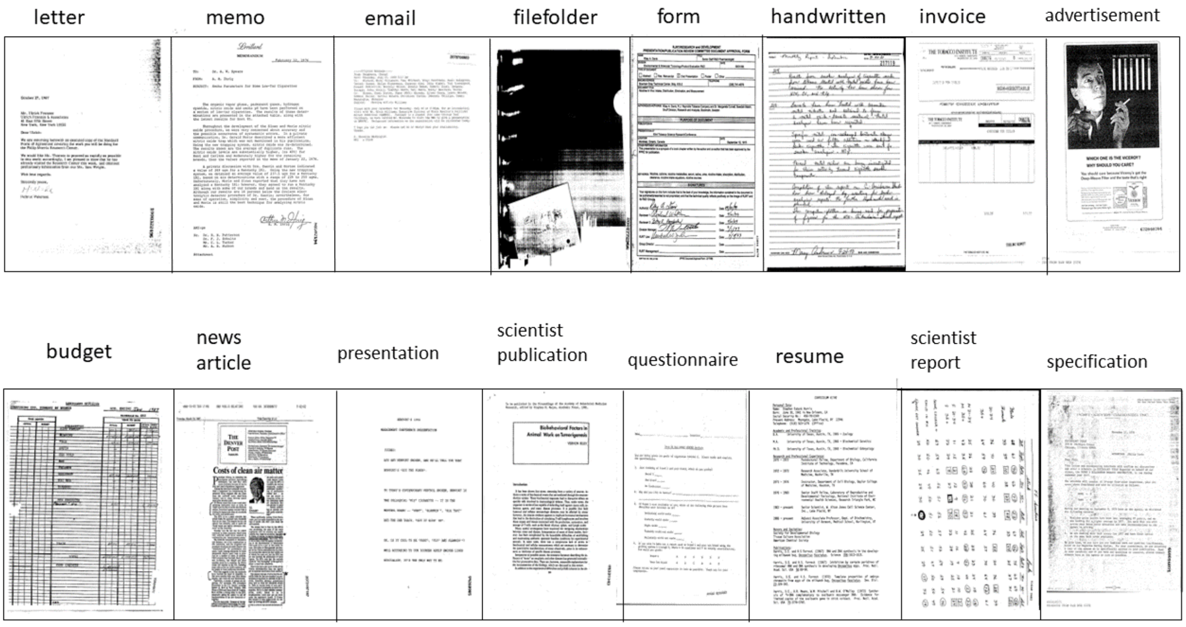


Figure 3.2: Samples of different document classes in the RVL-CDIP [19] dataset which illustrate the low inter-class discrimination and high intraclass variations of document images.

Recently, pre-training techniques have increased the development of Document AI, achieving notable progress on downstream tasks. RVL-CDIP is a representative dataset for the evaluation of document image classification tasks. It has been used in several state-of-the-art works for document AI [53, 54, 5, 6].

3.3 Tobacco-3482

Tobacco-3482, also known as SmallTobacco, is another publicly available dataset that consists of 3482 images belonging to 10 different classes extracted. It was selected and labeled by Kumar (2012) [26]. An example image from each of the ten classes (Advertisement, E-mail, Form, Letter, Memo, News, Note, Report, Resume, Scientific) in Tobacco-3482 is shown in Figure 3.3. Differently from RVL-CDIP, the Tobacco-3482 does not come with pre-built subsets for train, validation, and test. Except for the Note and Report class, all others are already included in the RVL-CDIP dataset. Unlike the RVL-CDIP dataset, the distribution of the samples across the classes is not the same.

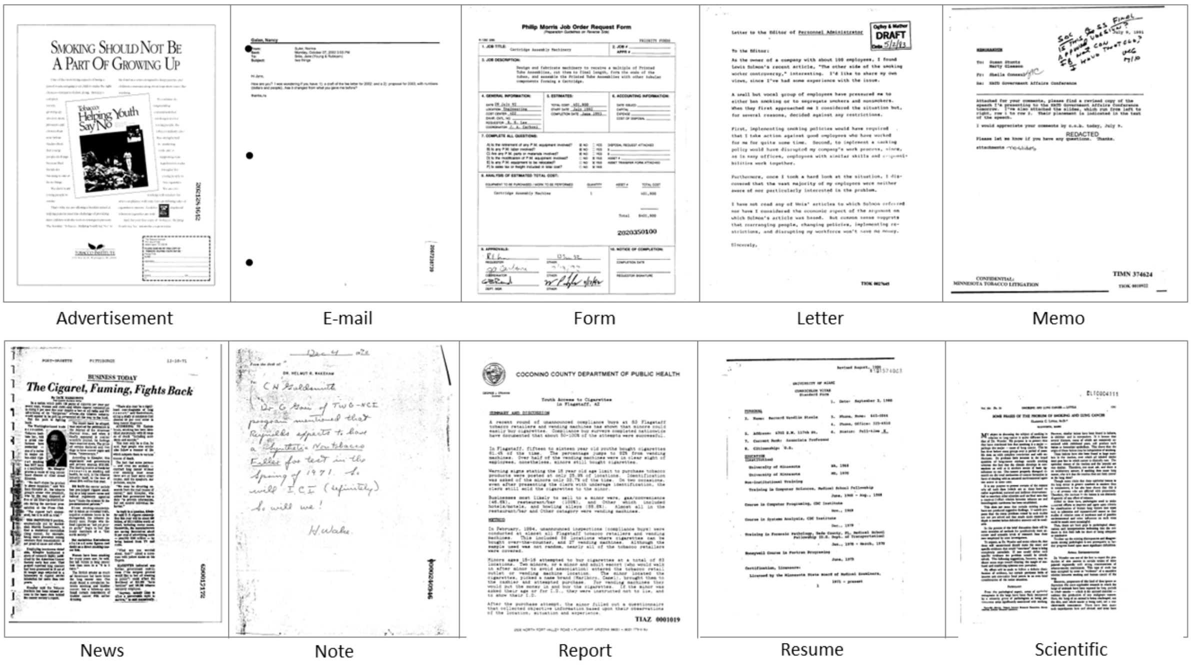


Figure 3.3: Samples of different document classes in the Tobacco-3482 [26] dataset which illustrate the low inter-class discrimination and high intraclass variations of document images.

SmallTobacco dataset was used in a number of related papers for document image classification. Tobacco-3482 was used by Noce et al. (2016) [38] to evaluate a document image classification method based on combined visual and textual information. Asim et al. (2019) [5] utilized InceptionV3 model to classify text document images using transfer learning. They have trained InceptionV3 on the RVL-CDIP dataset using ImageNet weights and utilized transfer learning to classify tobacco-3482 text document images. To evaluate the effectiveness of a cross-modal deep network that jointly learns text-image features to classify document images, Bakkalli et al. (2020) [8] utilized the benchmark tobacco-3482 dataset.

3.4 VICTOR

VICTOR [3, 4] is a dataset of legal documents belonging to Brazil's Supreme Court (Supremo Tribunal Federal or STF) suits were labeled by a team of experts. This dataset is part of the VICTOR project, a partnership between the STF, UnB, and Finatec. The project's objective was to develop an artificial intelligence tool to assist the STF in analyzing extraordinary appeals received from all over the country, especially regarding their classification in the most recurrent themes of general repercussions. Some other works that resulted from this project using the VICTOR dataset are [11, 4, 3, 42].

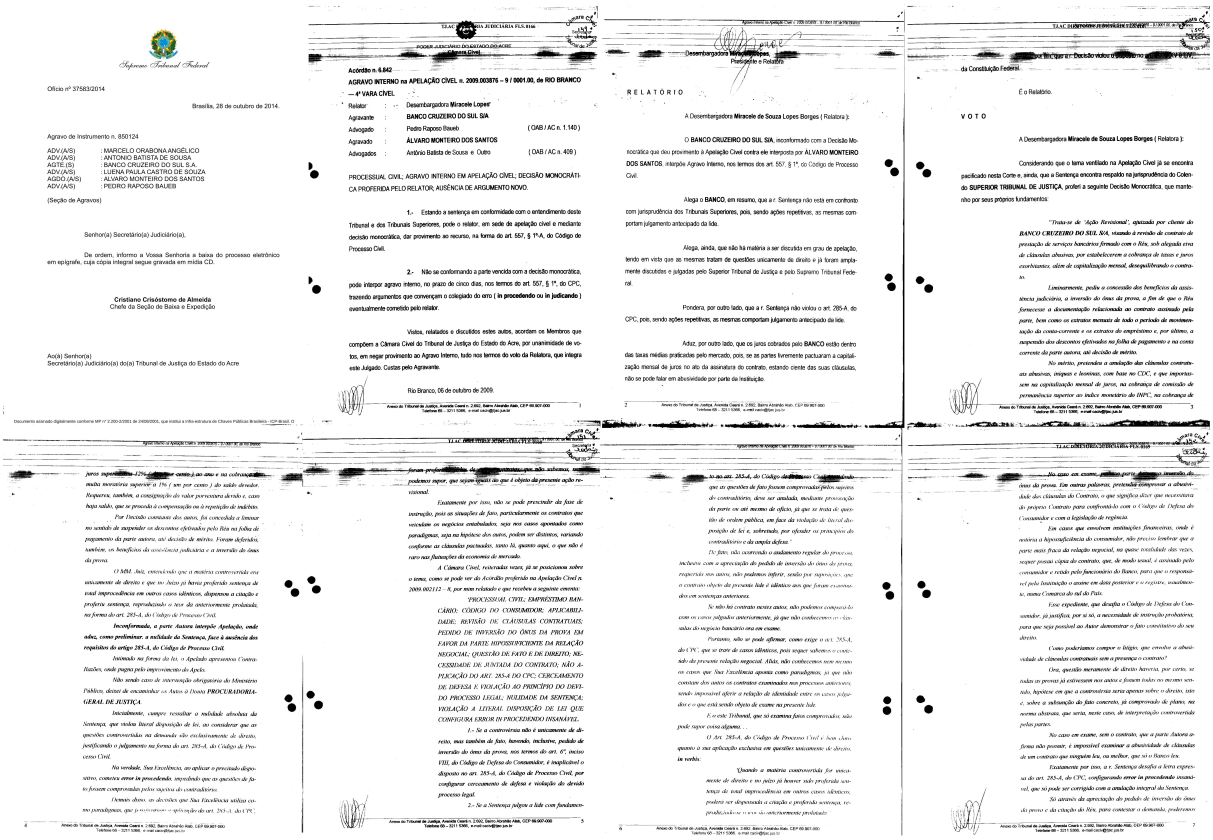
The VICTOR dataset comprises 45,532 Extraordinary Appeals (*Recursos Extraordinários*) from the STF. Each suit contains several different documents, ranging from the appeal itself to certificates and rulings, totaling 692,966 documents comprising 4,603,784 pages. Most cases reach the court as PDF files where each file represents a specific document or is an unstructured volume containing multiple documents. A significant part of the data provided is in the form of images obtained by scanning printed documents that often contain handwritten notes, stamps, stains and other sources of visual noise, like the ones shown in Figure 3.4. The dataset contains two types of annotations and supports two types of tasks: document type classification; and theme assignment, a multilabel problem.

There are six different labels for document type classification:*Acórdão*, for lower court decisions under review; *Recurso Extraordinário* (RE), for appeal petitions; *Agravo de Recurso Extraordinário* (ARE), for motions against the appeal petition; *Despacho*, for court orders; *Sentença* for judgments; and Others for documents not included in the previous classes.

Labels for lawsuit theme classification assign one or more General Repercussion (*Repercussão Geral*) themes to each Extraordinary Appeal. There are 28 theme options identified by integers (e.g., theme 810) corresponding to the most frequent ones and one class (with ID 0) for the remaining themes, summing up to 29 classes.

First, Luz et al. (2020) [3] proposed three versions of this VICTOR dataset: Big, Medium, and Small. Big VICTOR (BVic) is used only for theme classifications since it contains all data, including the unlabeled documents. Medium VICTOR (MVic), with 44,855 suits, 628,820 documents, and 2,086,899 pages, is the result of filtering out those samples and can be employed for both theme and document type classification. The number of MVic processes was limited for each theme to 100 samples in each set to create the Small VICTOR (SVic) dataset, which contains 6,510 Extraordinary Features, 94,267 documents, and 339,478 pages.

Luz et al. (2022) [3] proposed SVic+, a multimodal dataset of lawsuits composed of ordered document images and corresponding texts. This SVic+ dataset is an extension of Small VICTOR, which was expanded to include the document images in addition to



Supremo Tribunal Federal

Ofício nº 37583/2014

Brasília, 28 de outubro de 2014.

Agravo de Instrumento n. 850124

ADV.(A/S)	: MARCELO ORABONA ANGÉLICO
ADV.(A/S)	: ANTONIO BATISTA DE SOUSA
AGTE.(S)	: BANCO CRUZEIRO DO SUL S.A.
ADV.(A/S)	: LUENA PAULA CASTRO DE SOUZA
AGDO.(A/S)	: ALVARO MONTEIRO DOS SANTOS
ADV.(A/S)	: PEDRO RAPOSO BAUEB

(Seção de Agravos)

Senhor(a) Secretário(a) Judiciário(a),

De ordem, informo a Vossa Senhoria a baixa do processo eletrônico em epígrafe, cuja cópia integral segue gravada em mídia CD.

Cristiano Crisóstomo de Almeida
Chefe da Seção de Baixa e Expedição

Ao(à) Senhor(a)
Secretário(a) Judiciário(a) do(a) Tribunal de Justiça do Estado do Acre

Documento assinado digitalmente conforme MP nº 2.200-2/2001 de 24/08/2001, que institui a Infra-estrutura de Chaves Públicas Brasileira - ICP-Brasil. O

TJ.AC [illegible]RIA JUDICIÁRIA FLS. 0166

Câmara Cível 149 Secretaria Tribunal de Justiça

PODER JUDICIÁRIO DO ESTADO DO ACRE
Câmara Cível

Acórdão n. 6.842

AGRAVO INTERNO na APELAÇÃO CÍVEL n. 2009.003876 – 9 / 0001.00, de RIO BRANCO — 4ª VARA CÍVEL

Relator	:	Desembargadora **Miracele Lopes**	
Agravante	:	**BANCO CRUZEIRO DO SUL S/A**	
Advogado	:	Pedro Raposo Baueb	(OAB / AC n. 1.140)
Agravado	:	**ÁLVARO MONTEIRO DOS SANTOS**	
Advogados	:	Antônio Batista de Sousa e Outro	(OAB / AC n. 409)

PROCESSUAL CIVIL; AGRAVO INTERNO EM APELAÇÃO CÍVEL; DECISÃO MONOCRÁTICA PROFERIDA PELO RELATOR; AUSÊNCIA DE ARGUMENTO NOVO.

1.- Estando a sentença em conformidade com o entendimento deste Tribunal e dos Tribunais Superiores, pode o relator, em sede de apelação cível e mediante decisão monocrática, dar provimento ao recurso, na forma do art. 557, § 1º-A, do Código de Processo Civil.

2.- Não se conformando a parte vencida com a decisão monocrática, pode interpor agravo interno, no prazo de cinco dias, nos termos do art. 557, § 1º, do CPC, trazendo argumentos que convençam o colegiado do erro (**in procedendo ou in judicando**) eventualmente cometido pelo relator.

Vistos, relatados e discutidos estes autos, acordam os Membros que compõem a Câmara Cível do Tribunal de Justiça do Estado do Acre, por unanimidade de votos, em negar provimento ao Agravo Interno, tudo nos termos do voto da Relatora, que integra este Julgado. Custas pelo Agravante.

Rio Branco, 06 de outubro de 2009.

Anexo do Tribunal de Justiça, Avenida Ceará n. 2.692, Bairro Abrahão Alab, CEP 69.907-000
Telefone 68 – 3211 5366, e-mail caciv@tjac.jus.br 1

Agravo Interno na Apelação Cível n. 2009.003876 – 9 / 0001.00, de Rio Branco

Desembargadora [illegible], Presidente e Relatora

RELATÓRIO

A Desembargadora **Miracele de Souza Lopes Borges** (Relatora):

O **BANCO CRUZEIRO DO SUL S/A**, inconformado com a Decisão Monocrática que deu provimento à Apelação Cível contra ele interposta por **ÁLVARO MONTEIRO DOS SANTOS**, interpõe Agravo Interno, nos termos dos art. 557, § 1º, do Código de Processo Civil.

Alega o **BANCO**, em resumo, que a r. Sentença não está em confronto com jurisprudência dos Tribunais Superiores, pois, sendo ações repetitivas, as mesmas comportam julgamento antecipado da lide.

Alega, ainda, que não há matéria a ser discutida em grau de apelação, tendo em vista que as mesmas tratam de questões unicamente de direito e já foram amplamente discutidas e julgadas pelo Superior Tribunal de Justiça e pelo Supremo Tribunal Federal.

Pondera, por outro lado, que a r. Sentença não violou o art. 285-A, do CPC, pois, sendo ações repetitivas, as mesmas comportam julgamento antecipado da lide.

Aduz, por outro lado, que os juros cobrados pelo **BANCO** estão dentro das taxas médias praticadas pelo mercado, pois, se as partes livremente pactuaram a capitalização mensal de juros no ato da assinatura do contrato, estando ciente das suas cláusulas, não se pode falar em abusividade por parte da Instituição.

2 Anexo do Tribunal de Justiça, Avenida Ceará n. 2.692, Bairro Abrahão Alab, CEP 69.907-000
Telefone 68 – 3211 5366, e-mail caciv@tjac.jus.br

TJ.AC DIRETORIA JUDICIÁRIA FLS. 0167

Câmara Cível 150 Secretaria Tribunal de Justiça

[illegible] por fim, que a r. Decisão violou o disposto no [illegible] da Constituição Federal.

É o Relatório.

VOTO

A Desembargadora **Miracele de Souza Lopes Borges** (Relatora):

Considerando que o tema ventilado na Apelação Cível já se encontra pacificado nesta Corte e, ainda, que a Sentença encontra respaldo na jurisprudência do Colendo **SUPERIOR TRIBUNAL DE JUSTIÇA**, proferi a seguinte Decisão Monocrática, que mantenho por seus próprios fundamentos:

*"Trata-se de 'Ação Revisional', ajuizada por cliente do **BANCO CRUZEIRO DO SUL S/A**, visando à revisão de contrato de prestação de serviços bancários firmado com o Réu, sob alegada eiva de cláusulas abusivas, por estabelecerem a cobrança de taxas e juros exorbitantes, além de capitalização mensal, desequilibrando o contrato.*

Liminarmente, pediu a concessão dos benefícios da assistência judiciária, a inversão do ônus da prova, a fim de que o Réu fornecesse a documentação relacionada ao contrato assinado pela parte, bem como os extratos mensais de todo o período de movimentação da conta-corrente e os extratos do empréstimo e, por último, a suspensão dos descontos efetivados na folha de pagamento e na conta corrente da parte autora, até decisão de mérito.

No mérito, pretendeu a anulação das cláusulas contratuais abusivas, iníquas e leoninas, com base no CDC, e que importassem na capitalização mensal de juros, na cobrança de comissão de permanência superior ao índice monetário do INPC, na cobrança de

Anexo do Tribunal de Justiça, Avenida Ceará n. 2.692, Bairro Abrahão Alab, CEP 69.907-000
Telefone 68 – 3211 5366, e-mail caciv@tjac.jus.br 3

Agravo Interno na Apelação Cível n. 2009.003876 – 9 / 0001.00, de Rio Branco

juros super[illegible] 12% ([illegible] por cento) ao ano e na cobrança de multa moratória superior a 1% (um por cento) do saldo devedor. Requereu, também, a consignação do valor porventura devido e, caso haja saldo, que se proceda à compensação ou à repetição de indébito.

Por Decisão constante dos autos, foi concedida a liminar no sentido de suspender os descontos efetivados pelo Réu na folha de pagamento da parte autora, até decisão de mérito. Foram deferidos, também, os benefícios da assistência judiciária e a inversão do ônus da prova.

O MM. Juiz, entendendo que a matéria controvertida era unicamente de direito e que no Juízo já havia proferido sentença de total improcedência em outros casos idênticos, dispensou a citação e proferiu sentença, reproduzindo o teor da anteriormente prolatada, na forma do art. 285-A, do Código de Processo Civil.

Inconformada, a parte Autora interpõe Apelação, onde aduz, como preliminar, a nulidade da Sentença, face à ausência dos requisitos do artigo 285-A, do Código de Processo Civil.

Intimado na forma da lei, o Apelado apresentou Contra-Razões, onde pugna pelo improvimento do Apelo.

*Não sendo caso de intervenção obrigatória do Ministério Público, deixei de encaminhar os Autos à Douta **PROCURADORIA-GERAL DE JUSTIÇA**.*

Inicialmente, cumpre ressaltar a nulidade absoluta da Sentença, que violou literal disposição de lei, ao considerar que as questões controvertidas na demanda são exclusivamente de direito, justificando o julgamento na forma do art. 285-A, do Código de Processo Civil.

*Na verdade, Sua Excelência, ao aplicar o precitado dispositivo, cometeu **error in procedendo**, impedindo que as questões de fato fossem comprovadas pelos sujeitos do contraditório.*

Demais disso, as decisões que Sua Excelência utiliza como paradigmas, que justificariam a aplicação do art. 285-A, do CPC,

4 Anexo do Tribunal de Justiça, Avenida Ceará n. 2.692, Bairro Abrahão Alab, CEP 69.907-000
Telefone 68 – 3211 5366, e-mail caciv@tjac.jus.br

TJ.AC DIRETORIA JUDICIÁRIA FLS. 0168

Câmara Cível 151 Secretaria Tribunal de Justiça

foram proferi[illegible] de [illegible] contratos, que não sabemos, nem podemos supor, que sejam iguais ao que é objeto da presente ação revisional.

Exatamente por isso, não se pode prescindir da fase de instrução, pois as situações de fato, particularmente os contratos que veiculam os negócios entabulados, seja nos casos apontados como paradigmas, seja na hipótese dos autos, podem ser distintos, variando conforme as cláusulas pactuadas, tanto lá, quanto aqui, o que não é raro nas flutuações da economia de mercado.

A Câmara Cível, reiteradas vezes, já se posicionou sobre o tema, como se pode ver do Acórdão proferido na Apelação Cível n. 2009.002112 – 8, por mim relatado e que recebeu a seguinte ementa:

'PROCESSUAL CIVIL; EMPRÉSTIMO BANCÁRIO; CÓDIGO DO CONSUMIDOR; APLICABILIDADE; REVISÃO DE CLÁUSULAS CONTRATUAIS; PEDIDO DE INVERSÃO DO ÔNUS DA PROVA EM FAVOR DA PARTE HIPOSSUFICIENTE DA RELAÇÃO NEGOCIAL; QUESTÃO DE FATO E DE DIREITO; NECESSIDADE DE JUNTADA DO CONTRATO; NÃO APLICAÇÃO DO ART. 285-A DO CPC; CERCEAMENTO DE DEFESA E VIOLAÇÃO AO PRINCÍPIO DO DEVIDO PROCESSO LEGAL; NULIDADE DA SENTENÇA; VIOLAÇÃO A LITERAL DISPOSIÇÃO DE LEI QUE CONFIGURA ERROR IN PROCEDENDO INSANÁVEL.

1.- Se a controvérsia não é unicamente de direito, mas também de fato, havendo, inclusive, pedido de inversão do ônus da prova, nos termos do art. 6º, inciso VIII, do Código de Defesa do Consumidor, é inaplicável o disposto no art. 285-A, do Código de Processo Civil, por configurar cerceamento de defesa e violação do devido processo legal.

2.- Se a Sentença julgou a lide com fundamen-

Anexo do Tribunal de Justiça, Avenida Ceará n. 2.692, Bairro Abrahão Alab, CEP 69.907-000
Telefone 68 – 3211 5366, e-mail caciv@tjac.jus.br 5

Agravo Interno na Apelação Cível n. 2009.003876 – 9 / 0001.00, de Rio Branco

to no art. 285-A, do Código de Processo Civil, impedindo que as questões de fato fossem comprovadas pelos sujeitos do contraditório, deve ser anulada, mediante provocação da parte ou até mesmo de ofício, já que se trata de questão de ordem pública, em face da violação de literal disposição de lei e, sobretudo, por ofender os princípios do contraditório e da ampla defesa.'

De fato, não ocorrendo o andamento regular do processo, inclusive com a apreciação do pedido de inversão do ônus da prova, requerida nos autos, não podemos inferir, senão por suposições, que o contrato objeto da presente lide é idêntico aos que foram examinados em sentenças anteriores.

Se não há contrato nestes autos, não podemos compará-lo com os casos julgados anteriormente, já que não conhecemos as cláusulas do negócio bancário ora em exame.

Portanto, não se pode afirmar, como exige o art. 285-A, do CPC, que se trate de casos idênticos, pois sequer sabemos o conteúdo da presente relação negocial. Aliás, não conhecemos nem mesmo os casos que Sua Excelência aponta como paradigmas, já que não constam dos autos os contratos examinados nos processos anteriores, sendo impossível aferir a relação de identidade entre os casos julgados e o que está sendo objeto de exame na presente lide.

E o este Tribunal, que só examina fatos comprovados, não pode supor coisa alguma. . .

*O Art. 285-A, do Código de Processo Civil é bem claro quanto à sua aplicação exclusiva em questões unicamente de direito, **in verbis**:*

'Quando a matéria controvertida for unicamente de direito e no juízo já houver sido proferida sentença de total improcedência em outros casos idênticos, poderá ser dispensada a citação e proferida sentença, reproduzindo-se o teor da anteriormente prolatada.'

6 Anexo do Tribunal de Justiça, Avenida Ceará n. 2.692, Bairro Abrahão Alab, CEP 69.907-000
Telefone 68 – 3211 5366, e-mail caciv@tjac.jus.br

TJ.AC DIRETORIA JUDICIÁRIA FLS. 0169

Câmara Cível 152 Secretaria Tribunal de Justiça

No caso em exame, [illegible] a parte [illegible] a inversão do ônus da prova. Em outras palavras, pretendia comprovar a abusividade das cláusulas do Contrato, o que significa dizer que necessitava do próprio Contrato para confrontá-lo com o Código de Defesa do Consumidor e com a legislação de regência.

Em casos que envolvem instituições financeiras, onde é notória a hipossuficiência do consumidor, não preciso lembrar que a parte mais fraca da relação negocial, na quase totalidade das vezes, sequer possui cópia do contrato, que, de modo usual, é assinado pelo consumidor e retido pelo funcionário do Banco, para que o responsável pela Instituição o assine em data posterior e o registre, usualmente, numa Comarca do sul do País.

Esse expediente, que desafia o Código de Defesa do Consumidor, já justifica, por si só, a necessidade de instrução probatória, para que seja possível ao Autor demonstrar o fato constitutivo do seu direito.

Como poderíamos compor o litígio, que envolve a abusividade de cláusulas contratuais sem a presença o contrato?

Ora, questão meramente de direito haveria, por certo, se todas as provas já estivessem nos autos e fossem todas no mesmo sentido, hipótese em que a controvérsia seria apenas sobre o direito, isto é, sobre a subsunção do fato concreto, já comprovado de plano, na norma abstrata, que seria, neste caso, de interpretação controvertida pelas partes.

No caso em exame, sem o contrato, que a parte Autora afirma não possuir, é impossível examinar a abusividade de cláusulas de um contrato que ninguém leu, ou melhor, que só o Banco leu.

*Exatamente por isso, a r. Sentença desafia a letra expressa do art. 285-A, do CPC, configurando **error in procedendo** insanável, que só pode ser corrigido com a anulação integral da Sentença.*

Só através da apreciação do pedido de inversão do ônus da prova e da citação do Réu, para contestar a demanda, poderemos

Anexo do Tribunal de Justiça, Avenida Ceará n. 2.692, Bairro Abrahão Alab, CEP 69.907-000
Telefone 68 – 3211 5366, e-mail caciv@tjac.jus.br 7

Figure 3.4: The first eight pages of a lawsuit of the VICTOR dataset [3]. While the first page is clean, the others come from an older document and contain ink stains, stamps, handwritten signatures and other artifacts.

textual data. Every page in the expanded corpus is stored in at least one of two formats. First, as text extracted through optical character recognition, with the following additional preprocessing steps: lower-casing, removal of stop words and alphanumeric tokens, e-mail and URL tokenisation (e-mails and URLs are replaced by the tokens "email" and "link"), and special tokenisation of legislation references (e.g., Lei (law) 11.419 to LEI_11419). Second, as JPEG images converted from the original PDF files, with mean width and height of 1664 and 2322 pixels, respectively.

3.5 Datasets from Websites

Dataset of HTML pages cited in most articles for the classification problem is WebKB [14], but it is very old and current web pages are very dynamic with different layouts. This dataset contains web pages collected from computer science departments of various universities in January 1997 by the 'World Wide Web Knowledge Base (Web->Kb)' project [1] of the 'CMU text learning group'. WebKB is a dataset comprising 8,282 web pages categorized into seven classes (Student, Faculty, Staff, Department, Course, Project, Other) collected from computer science departments of various universities. The class other is a collection of pages that were not deemed the 'main page' representing an instance of the previous six classes. For each class the dataset contains pages from the four universities (Cornell, Texas, Washington, Wisconsin) and and 4,120 miscellaneous pages collected from other universities.

The exponential growth in the amount of information on the Internet has made the classification of web pages essential for managing, retrieving and integrating information from the Web. In addition to this growth of the Internet in size, new technologies and areas of use are developed daily. The emergence of e-business as a business model has influenced organizations to review their processes and automate them. Furthermore, the Web has leveraged the business world by bringing the need to track specific topics, recognize important documents, and remove unwanted content [20].

A web page is a text file combining content and design using HTML codes. It is usually written in HTML with tags to structure the file, text, and hypertext that will navigate to other web pages. There are many attributes on a web page, such as URL address, text content, hyperlinks, image content, domain and server information, HTML tags, and semantic web tags. Nevertheless, automatic Web page classification is challenging due to its complexity, diversity of Web page contents (images of different sizes, text, hyperlinks), and computational cost. Furthermore, as HTML documents grow, the data extraction process has been plagued with lengthy processing time and noisy information. Other

[1] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb/index.html

important challenges in classifying web pages are the continuous addition of new content to the Internet, the amount of attributes that make it difficult to obtain a standard for classification and requires the use of complex techniques, and the difficulty of finding adequate and descriptive data sets [7].

Hashemi (2020) [20] presented a survey of the proposed methodologies in the literature for classifying web pages. Initially, the article investigates the classification models of web pages into three main categories: text-based, image-based and combining the two methods. Furthermore, it provides collective trends and insight into existing Web page classification models and points out research gaps. Aydos et al. (2020) [7] introduced a method of deep learning algorithms that combines multiple neural networks for web page classification. In this model, each element is represented by multiple descriptive images. After the training process of the neural network model, each element is classified by calculating its descriptive image results, and the model was evaluated using Google Image Search results as descriptive images. They also introduced the WebScreenshots dataset, suitable for content- or screenshot-based website classifications.

WebScreenshots [7] contains 20000 Web pages (URLs, text contents, screenshots in 1440×900 and 224×224) separated into 4 classes upon their visual appearance (screenshots). This dataset was created in the second quarter of 2019. Nonetheless, after analyzing the dataset images, we found that it does not meet our document classification objective.

3.6 Summary

This chapter presented four datasets composed of images of literature documents for page stream segmentation task and document classification and two datasets of web pages, namely: Tobacco800, RVL-CDIP, Tobacco-3482, VICTOR, WebScreenshots and WebKB. The first four datasets are visually rich documents containing both text and non-text. Tobacco800, RVL-CDIP and Tobacco-3482 contain images of publicly available English business documents extracted from the Legacy Tobacco Documents Library (LTDL) and are very similar. The VICTOR dataset was obtained from STF court documents in Portuguese.

Finally, section 3.5 addresses the importance of classifying Web documents and the difficulty of finding representative datasets.We present the publicly available WebKB dataset, but it is very old, and the documents are very similar. WebScreenshots is another dataset of web pages that we present, being more recent. In addition, two recent works on extracting information from HTML pages.

After analyzing all the datasets, it was found that the datasets of web pages are not suitable to work with our approach. In addition, Tobacco-3482 is practically a subset of the RVL-CDIP. Given the above, only three datasets (Tobacco800, RVL-CDIP and VICTOR) were chosen to evaluate our proposed approach to the document classification task.

Chapter 4

LayoutQT

In Chapter we present LayoutQT - Layout Quadrant Tags, a lightweight preprocessing method focusing on combinations of texts and their spatial information without relying on visual features or activations from the visual modalities. Specifically, we propose a new set of tokens that encode spatial regions in language models and show that they improve results in downstream tasks with low computational cost. We evaluated our method with page stream segmentation and document classification task with Tobacco-800 and RVL-CDIP datasets, respectively.

4.1 Preprocessing LayoutQT

Our algorithm is based on a Bottom-up approach, which defines primitive components to start the clustering process. It starts with the bounding box of words as a primitive component of the page. The word grouping process identifies a group of nearest neighbors of each bounding box to form lines and blocks of text until the page end. Furthermore, each document page is divided into rectangular regions with the same *height* and *width* dimensions. Each quadrant has layout location information that is represented by spatial tokens.

Spatial tokens are added at the beginning and end of each line when indicating the quantized coordinates of the bounding box that the line belongs to. The text group beginning tag considers the distances from the top left corner of the bounding box to the image's left edge and top edge. Likewise, the end tag considers the distance between the bottom right corner of the bounding box and the image's bottom edge and right edge. Table 4.1 presents spatial tokens and their descriptions used in our LayoutQT model. For example, the beginning of a text block is marked with $xxQw_i\_h_j$ *xxbob*, to indicate the position (quadrant) of the beginning of the text block. The centralized parts of the text are also marked with spatial tokens *xxeob* and *xxbcet*.

Table 4.1: **Descriptions** of the spatial tokens

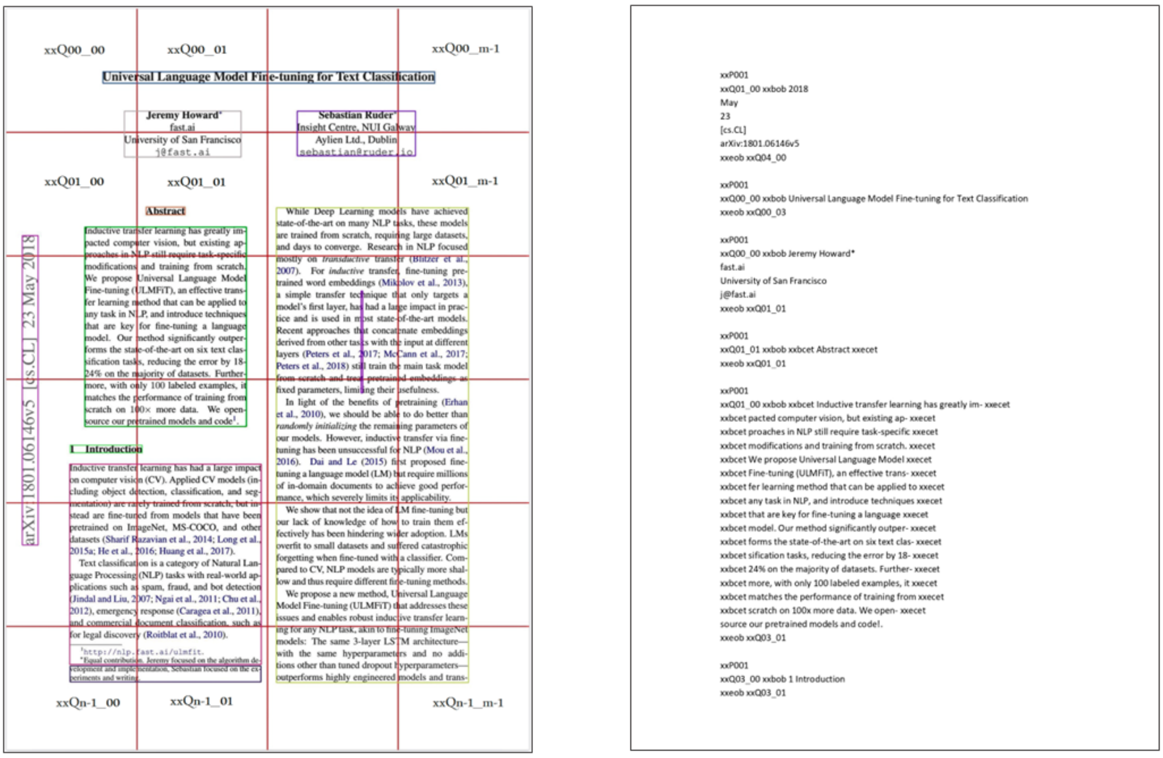
Special Token	**Descriptions**
$xxPn_k$	Page Number
$xxbob$	Begin Of Block
$xxeob$	End Of Block
$xxbcet$	Begin Of Centered Text
$xxecet$	End Of Centered_Text
$xxQw_i\_h_j$	Quadrant w_i Row h_jColumn

LayoutQT's algorithm (4.1) takes single-page or multi-page documents as input and generates tokenized text t with layout information. Initially, it adds a spatial token to the text to indicate the page. It then starts using an OCR engine [43] to generate word bounding boxes. The algorithm scans the page from top to bottom and left to right to find the boundaries of text groups and identifying the top left corner of the group. After this, it injects the location information through the spatial token. It sorts the groups that belong to the same column on the page to check which groups are centralized and adds the tokens. Moreover, it ends by adding the end-of-group spatial token. The text extraction with spatial tags is saved in a text file.

Algorithm 1 LayoutQT Algorithm
Input: multi page document
Output: tokenized text t

```
for $page = 0, \ldots, N-1$ do
    $t+ =$ add page token (where $+ =$ means insert symbol in string $t$)
    triage each word bounding boxes into line and group
    triage groups into coherent page columns
    for each group do
        $t+ =$ quadrant coordinate of group top left corner
        for each text line in this group do
            check line centralization w.r.t. its page column center position
            if the line is centralized then
                $t+ =$ centre tag
            end if
            $t+ =$ textual contents of the line
            if the line is centralized then
                $t+ =$ centre tag
            end if
        end for
        $t+ =$ quadrant coordinate of group bottom right corner
    end for
end for
```

Figure 4.1 presents a visual illustration from LayoutQT to the document page. The document input image is divided into quadrants and text groups on the left side. Each row is numbered from left to right and each column is numbered from top to bottom, so the tags of the first and last quadrants are, respectively, $xxQ00\_00$ and $xxQn-1\_m-1$. Inspired by tokenization of Fastai [22], which adds spatial tokens at the beginning and end of the sentence, LayoutQT adds spatial tokens with information about the bounding box position. All spatial tokens start with the characters xx, which is not a common English word prefix. They are added using rules for the model to recognize the important parts of a text. The image of the text file tokenized by our model is on the right side.



```
xxP001
xxQ01_00 xxbob 2018
May
23
[cs.CL]
arXiv:1801.06146v5
xxeob xxQ04_00

xxP001
xxQ00_00 xxbob Universal Language Model Fine-tuning for Text Classification
xxeob xxQ00_03

xxP001
xxQ00_00 xxbob Jeremy Howard*
fast.ai
University of San Francisco
j@fast.ai
xxeob xxQ01_01

xxP001
xxQ01_01 xxbob xxbcet Abstract xxecet
xxeob xxQ01_01

xxP001
xxQ01_00 xxbob xxbcet Inductive transfer learning has greatly im- xxecet
xxbcet pacted computer vision, but existing ap- xxecet
xxbcet proaches in NLP still require task-specific xxecet
xxbcet modifications and training from scratch. xxecet
xxbcet We propose Universal Language Model xxecet
xxbcet Fine-tuning (ULMFiT), an effective trans- xxecet
xxbcet fer learning method that can be applied to xxecet
xxbcet any task in NLP, and introduce techniques xxecet
xxbcet that are key for fine-tuning a language xxecet
xxbcet model. Our method significantly outper- xxecet
xxbcet forms the state-of-the-art on six text clas- xxecet
xxbcet sification tasks, reducing the error by 18- xxecet
xxbcet 24% on the majority of datasets. Further- xxecet
xxbcet more, with only 100 labeled examples, it xxecet
xxbcet matches the performance of training from xxecet
xxbcet scratch on 100x more data. We open- xxecet
source our pretrained models and code!.
xxeob xxQ03_01

xxP001
xxQ03_00 xxbob 1 Introduction
xxeob xxQ03_01
```

Figure 4.1: Diagram of our LayoutQT method. On the left side, an input document is divided into quadrants, receives a spatial token $xxQw_i\_h_j$ according to row i and column j positions. The green rectangles represent the bounding box of text. On the right side, the texts extracted by the OCR system with the tags indicating the position (quadrant) of beginning and end of the text of each text block.

4.2 Experiments

This section exposes the experiments in detail. We apply our model to two downstream tasks, one for page segmentation and the other for classifying document types. We performed four experiments with the Tobacco800 dataset for the page stream segmentation task and two experiments with the RVL-CDIP dataset for the document type classification. We followed the train, validation and test split defined by [10] for Tobacco800, whilst for the RVL-CDIP dataset we use the split defined by [19]. We performed classification experiments with and without using our model to compare the results. Thus, it identified the location (quadrants) of each bounding box's beginning, middle, and end and added spatial tokens (tags) to the text.

4.2.1 Experiment details

First, following the blue flow of Figure 4.2, we provided document images as input to our LayoutQT, which virtually maps page space into equally spaced quadrants. After that, we map each text block start and end position into the related quadrant and inject spatial tokens to mark the start and end position of each text box. Then the text of each bounding box is extracted along with the spatial tokens taking into account their position on the document page. The extracted texts were saved in text files. As a baseline, in the red flow, the document images fed an OCR engine to extract the text without the spatial tokens. Subsequently, the extracted texts were tokenized, trained, tested, and evaluated using the same language model for the document classification task. Finally, we compare the results obtained with and without tags.

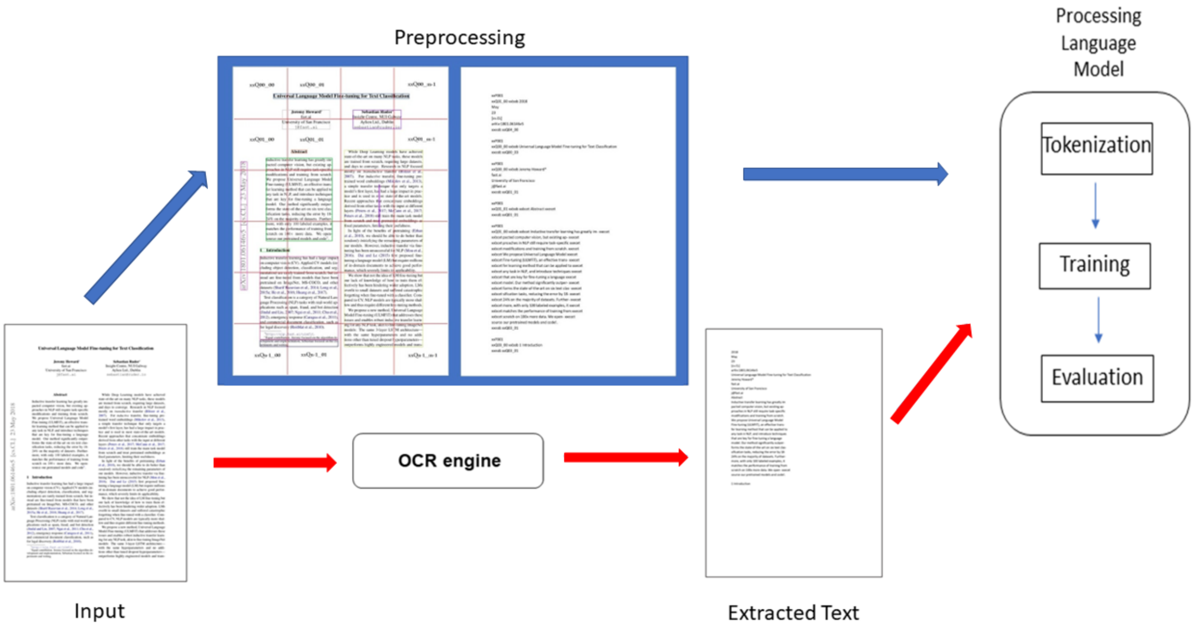


Figure 4.2: The experiment flow diagram with our LayoutQT preprocessing (blue arrows) and the baseline (red arrows) without using the method.

In processing steps, We first lowercase the text and use SentencePiece [25] to tokenize it. We use the same tokenization for the baseline method to establish a fair comparison of the approaches. All of the arguments that can be passed to Tokenize and Numericalize can also be passed to TextBlock. To train and evaluate the document page stream segmentation (PSS), we used the Tobacco-800 dataset in two network architectures, a Long Short-Term Memory (LSTM) [45] for text classification. Secondly, we used Universal Language Model Fine-Tuning (ULMFiT) [23] with ASGD Weight-Dropped LSTM (AWD-LSTM) [36] for ranking the pages as *first_page* or *next_page* class on the same dataset. AWD-LSTM language model which uses DropConnect and the average random gradient descent method, and several other regularization strategies. The weight-dropped LSTM strategy uses a DropConnect mask on the hidden-to-hidden weight matrices, as a means to prevent overfitting across the recurrent connections.

For document classification with the RVL-CDIP dataset, inspired by Howard and Ruder (2018) [23] we used ULMFiT with AWD-LSTM for training, testing and evaluation. In the training phase, each evaluation dataset was split into training, validation and test subsets. We minimized the loss function using the training set, and we assessed the model from each epoch on the validation set. We saved the model's weights of the lowest loss in the validation set iteration and evaluated the model with these weights in the test set after the whole training.

To evaluate, we compared the execution of the classifier using LayoutQT method generating the quadrant tags and without the preprocessing with both Tobacco-800 and RVL-CDIP datasets. The loss function used by default is the cross entropy loss as we essentially have a classification problem (the different categories are the words in our vocabulary).

4.2.2 Experiment Setting

This subsection describes the implementation details used to the proposed approach. We used our method of preprocessing, which starts with using an OCR engine to generate blocks of text (bounding boxes) and delimit textual elements for each image in the document. Then, It drew the horizontal and vertical lines dividing each page of the document into 24 equivalent quadrants: 4 horizontal x 6 vertical.

Initially, we performed two experiments with the tobacco-800 dataset for binary classification of document pages, one with LayoutQT and one with the baseline. We used an LSTM backbone (composed of 256 nodes fully connected with activation “ReLU" and a dropout of 0.3). Furthermore, we use binary cross-entropy as a loss function with softmax activation and Adam as an optimizer. The model was trained for 10 epochs with a batch size of 128.

Table 4.2: Result of binary classification on Tobacco 800 dataset.

Model	**Modality**	**Backbone**	**Accuraccy**
Braz_etal(2021) [10]	image only	VGG16*	92.0%
Braz_etal(2021) [10]	only image	EfficientNet-B0*	83.7%
Baseline	only text	LSTM	79.1%
LayoutQT	text + layout	LSTM	**84.7%**
Baseline	only text	ULMFiT (AWD-LSTM)	97.5%
LayoutQT	text + layout	ULMFiT (AWD-LSTM)	**99.5%**

(*) reported by Braz et al.(2021)[10]

Next, we performed the experiments with an AWD-LSTM backbone, using Tobacco800 and RVL-CDIP datasets. The model was trained for one cycle of 30 epochs with a batch size of 128 documents and a sequence length of 72 using NVIDIA Tesla V100 32GB GPU.

4.3 Results and Discussion

The document page binary classification task, which identifies whether the document is a first page (FirstPage) or a continuation (NextPage), was performed with the Tobacco-800 dataset using our LayoutQT method by adding quadrant tags and as a baseline processing without placing tags using only text. Such experiments were processed using the LSTM and ULMFiT with AWD-LSTM models. The validation split results in Table 4.2 brought a 2% gain applying LayoutQT when compared with the baseline by only using text sequence architecture. The validation split results in Table 4.2 brought out that it had a large room for improvement the baseline by only using text sequence architecture, since we've surpassed Braz et al. (2021) [10] baseline by 5.5 points. After applying LayoutQT we got more 2 points out of the 2.5 possible, which turns out a 80% of possible gain.

Figure 4.3 shows the confusion matrix of binary classification to tobacco800 dataset without tags (baseline) and with tags of quadrants (LayoutQT) using ULMFiT (AWD-LSTM) model. It's clear that in the ranking of the first page images, both the baseline and our model missed only one image, but in the ranking of the continuity pages as the next page, the model without the tags was missing four images, while with the tags there was a single error.

The confusion matrices of the document image classification result (see Figure 4.4) shows an improvement using our preprocessing. The classification results on the RVL-CDIP dataset without adding location tokens was 80.4% while using our preprocessing method with the addition of location tokens it achieved 83.6% accuracy. Our proposed

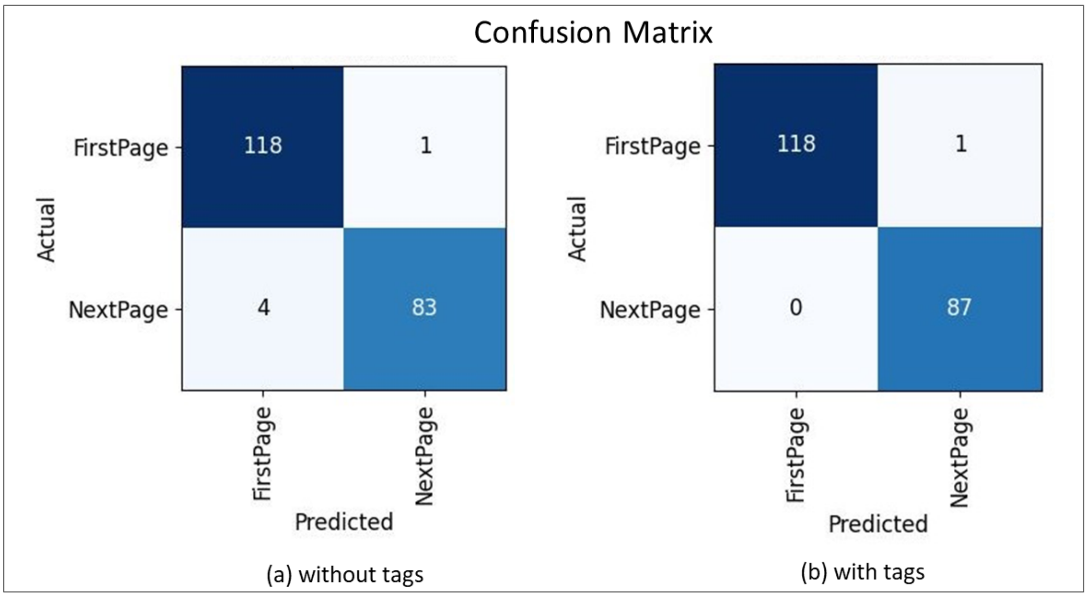
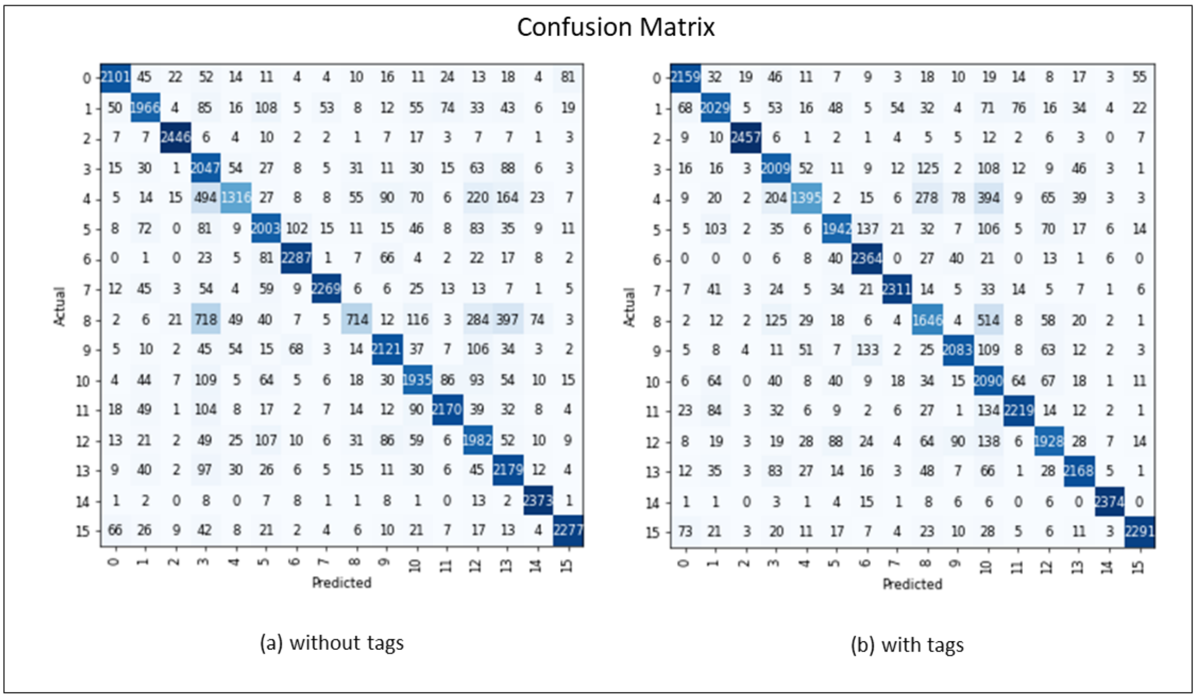


Figure 4.3: Confusion matrix of Tobacco-800 binary classification using AWD-LSTM.(a) results found from the experiment without the tags, that is, with the baseline. (b) results obtained with the tags (LayoutQT).

approach demonstrated a superior performance in the tested settings document classification tasks.

Table 4.3 compares the performance of the two document classification proposals, baseline, and LayoutQT, from the RVL-CDIP dataset for each document class. The results show that our approach with the addition of positional tags performed better. Of the 16 classes of documents, the accuracy of our approach was inferior in only five classes (handwritten, scientific report, news article, presentation, and questionnaire). However, the overall ranking result with LayoutQT showed an advantage of 3.2% in accuracy compared to baseline. Furthermore, our approach to email documents obtained the highest accuracy value with 97.6%.

We can observe that LayoutQT obtained worse results in five document classes, such as a scientific report class. There is no standardization of the layout of this type of document. In this case, the classification depended more on the textual characteristics than the layout, which may have interfered with the results obtained. However, in the classification of file folder documents, LayoutQT obtained a significantly better result than the baseline. That is, the percentage result of our approach was more than twice the baseline.



Confusion Matrix

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	2101	45	22	52	14	11	4	4	10	16	11	24	13	18	4	81
1	50	1966	4	85	16	108	5	53	8	12	55	74	33	43	6	19
2	7	7	2446	6	4	10	2	2	1	7	17	3	7	7	1	3
3	15	30	1	2047	54	27	8	5	31	11	30	15	63	88	6	3
4	5	14	15	494	1316	27	8	8	55	90	70	6	220	164	23	7
5	8	72	0	81	9	2003	102	15	11	15	46	8	83	35	9	11
6	0	1	0	23	5	81	2287	1	7	66	4	2	22	17	8	2
7	12	45	3	54	4	59	9	2269	6	6	25	13	13	7	1	5
8	2	6	21	718	49	40	7	5	714	12	116	3	284	397	74	3
9	5	10	2	45	54	15	68	3	14	2121	37	7	106	34	3	2
10	4	44	7	109	5	64	5	6	18	30	1935	86	93	54	10	15
11	18	49	1	104	8	17	2	7	14	12	90	2170	39	32	8	4
12	13	21	2	49	25	107	10	6	31	86	59	6	1982	52	10	9
13	9	40	2	97	30	26	6	5	15	11	30	6	45	2179	12	4
14	1	2	0	8	0	7	8	1	1	8	1	0	13	2	2373	1
15	66	26	9	42	8	21	2	4	6	10	21	7	17	13	4	2277

(a) without tags

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	2159	32	19	46	11	7	9	3	18	10	19	14	8	17	3	55
1	68	2029	5	53	16	48	5	54	32	4	71	76	16	34	4	22
2	9	10	2457	6	1	2	1	4	5	5	12	2	6	3	0	7
3	16	16	3	2009	52	11	9	12	125	2	108	12	9	46	3	1
4	9	20	2	204	1395	2	15	6	278	78	394	9	65	39	3	3
5	5	103	2	35	6	1942	137	21	32	7	106	5	70	17	6	14
6	0	0	0	6	8	40	2364	0	27	40	21	0	13	1	6	0
7	7	41	3	24	5	34	21	2311	14	5	33	14	5	7	1	6
8	2	12	2	125	29	18	6	4	1646	4	514	8	58	20	2	1
9	5	8	4	11	51	7	133	2	25	2083	109	8	63	12	2	3
10	6	64	0	40	8	40	9	18	34	15	2090	64	67	18	1	11
11	23	84	3	32	6	9	2	6	27	1	134	2219	14	12	2	1
12	8	19	3	19	28	88	24	4	64	90	138	6	1928	28	7	14
13	12	35	3	83	27	14	16	3	48	7	66	1	28	2168	5	1
14	1	1	0	3	1	4	15	1	8	6	6	0	6	0	2374	0
15	73	21	3	20	11	17	7	4	23	10	28	5	6	11	3	2291

(b) with tags

Figure 4.4: Confusion matrix of RVL-CDIP composed of 16 document classes: 0-letter, 1-form, 2-email, 3-handwritten, 4-advertisement, 5-scientific report, 6-scientific publication, 7-specification, 8-file folder, 9-news article, 10-budget, 11-invoice, 12-presentation, 13-questionnaire, 14-resume and 15-memo. Confusion matrix (a) shows results of processing without tags, while confusion matrix (b) shows results of our model using tags.

Table 4.3: Accuracy of the document types classification on RVL-CDIP dataset obtained with the baseline and LayoutQT. The results in absolute numbers of hits and misses by classes are shown in Figure 4.4

Class	**Document Type**	**Baseline**	**LayoutQT**
0	letter	85.5%	**87.8%**
1	form	78.8%	**81.3%**
2	email	97.2%	**97.6%**
3	handwritten	**84.9%**	83.3%
4	advertisement	55.2%	**58.5%**
5	scientific report	**80.6%**	78.2%
6	scientific publication	89.1%	**92.1%**
7	specification	91.9%	**93.6%**
8	file folder	31.9%	**73.5%**
9	news article	**86.4%**	84.9%
10	budget	77.8%	**84.0%**
11	invoice	87.9%	**89.9%**
12	presentation	**79.9%**	77.8%
13	questionnaire	**90.0%**	89.5%
14	resume	93.6%	**93.7%**
15	memo	91.9%	**92.5%**
Average		80.4%	**83.6%**

4.4 Summary

This chapter introduced our LayoutQT - Layout Quadrant Tags model, which divides a document into 24 quadrants. Each quadrant is identified by a positional token that is later inserted into the embedding of text blocks. In addition, document classification experiments were performed using an LSTM and AWD-LSTM architecture on two state-of-the-art datasets: Tobacco800 and RVL-CDIP. The LayoutQT method experiments combining text and layout features showed an improvement over the baseline of at least two percentage points in accuracy. Ultimately, this chapter yielded an article submitted for publication in the journal Engineering Applications of Artificial Intelligence in June, and we are awaiting feedback from reviewers.

Chapter 5

Concluding Remarks and Proposal

This chapter concludes with a summary of what has been accomplished so far. Then, it presents the plan of the next activities to be developed monthly to validate our proposal.

5.1 Conclusion

We proposed a simple and effective method combining layout and textual features with a low computational cost for text processing. We use a rules-based and feature engineering approach. Specifically, it takes information from the bounding boxes issued by an OCR engine and extracts some coherent information from the text layout, like page and document position for each text block. Our method, introduced in Chapter 4, divides the document into quadrants and uses the quadrant location to add spatial tokens to mark each text box's start and end position. In addition, we also applied a greedy algorithm to organize the words in blocks, firstly processing lines, then processing the groups of words.

We performed experiments with a fixed amount of 24 rectangular regions (quadrants) in just two databases composed of document images in .tiff format. Our method has shown good results in the initial experiments, as presented in Chapter 4. This work has been described in a paper that we have submitted to the journal Engineering Applications of Artificial Intelligence.

In the next section, we introduce future work and present a timeline of our work plan for completing the thesis.

5.2 Schedule

For future work, we propose to evaluate the model with other parameters, varying the number of quadrants, inserting more tokens and excluding tokens. We also plan to evaluate our method for classifying document images in other datasets. For this, we searched

Table 5.1: Summary of research activities planning.

Activity	Sep	Out	Nov	Dec	Jan/23
Experiments	✓	✓	✓		
Baseline on VICTOR dataset	✓				
Current model on VICTOR dataset	✓				
Model parameter adjustment		✓	✓		
Training and validation on chosen datasets		✓	✓		
Thesis Writing	✓	✓	✓		
Background Update + Related Works	✓				
Methodology		✓			
Results and Discussion		✓	✓		
Submit to board				✓	
Wrap Up				✓	✓
Preparing the presentation of the thesis				✓	✓
Thesis defense					✓

for datasets of document images publicly available in the literature for the classification problem. After analyzing the datasets presented in Chapter 3, we chose to work with the VICTOR [4] dataset because it is more robust and different from the previous ones (Tobacco800 [28] and RVL-CDIP [19]) already used in our approach. VICTOR [4] is a dataset composed of 692,966 legal documents in Portuguese with 4,603,784 pages.

We planned activities to be developed in five months. According to the schedule presented in Table 5.1, we divided the activities into three groups: Experiments, Thesis Writing and Wrap Up. Experiments and thesis writing will be carried out in parallel for the first three months. That should not be a problem as the scheduled tasks are independent. Then we will dedicate the last two months to the wrap-up stage. In the first month, the experiments will be dedicated to preparing and executing the VICTOR dataset for training and validation in the baseline and LayoutQT with the current parameters. Simultaneously, we will update the background writing and related works (Thesis Chapter 2). The results obtained in the experiments will be recorded for further analysis

We allocate the second and third months to improve our LayoutQT model. We will start by adjusting the parameters (number of quadrants, inclusion and exclusion of positional tokens) to run the model with adjusted parameters. Next, we trained the model on the three chosen databases (Tobacco800, RVL-CDIP and VICTOR), evaluated the trained model in the document classification task and compared the results with the previous model. The development of these experiments will be carried out iteratively, following a cycle of: i) training, ii) evaluation, iii) improvement of the model and iv) starting a new cycle. In addition, we will update the writing of the methodology, results and conclusion simultaneously with the execution of the experiments. This thesis writing stage ends with submitting the work to the defence board.

Finally, the last two are allocated for the wrap-up. This stage comprises from the delivery of the thesis to the defence board until the date of thesis presentation. The thesis reading period by the defence board will be dedicated to preparing the presentation of the work and concluding with a thesis defence.

References

[1] Aggarwal, M., Sarkar, M., Gupta, H., and Krishnamurthy, B.: *Multi-Modal Association based Grouping for Form Structure Extraction.* 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 2064–2073, 2020. iv, 22, 27

[2] Appalaraju, S., Jasani, B., Kota, B.U., Xie, Y., and Manmatha, R.: *DocFormer: End-to-End Transformer for Document Understanding.* 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 973–983, 2021. 24, 27

[3] Araujo, P.H.L. de, Almeida, A.P.G.S. de, Braz, F.A., Silva, N.C. da, Barros Vidal, F. de, and Campos, T.E. de: *Sequence-aware multimodal page classification of Brazilian legal documents.* International Journal on Document Analysis and Recognition (IJDAR), jul 2022. `https://doi.org/10.10072Fs10032-022-00406-7`. v, 32, 33

[4] Araujo, P.H. Luz de, Campos, T.E. de, Ataides Braz, F., and Silva, N. Correia da: *VICTOR: a Dataset for Brazilian Legal Documents Classification.* In *Proceedings of the 12th Language Resources and Evaluation Conference*, pp. 1449–1458, Marseille, France, May 2020. European Language Resources Association, ISBN 979-10-95546-34-4. `https://aclanthology.org/2020.lrec-1.181`. 28, 32, 48

[5] Asim, M.N., Khan, M.U.G., Malik, M.I., Razzaque, K., Dengel, A., and Ahmed, S.: *Two Stream Deep Network for Document Image Classification.* In *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1410–1416, 2019. 22, 27, 30, 31

[6] Audebert, N., Herold, C., Slimani, K., and Vidal, C.: *Multimodal Deep Networks for Text and Image-Based Document Classification.* In Cellier, P. and Driessens, K. (eds.): *Machine Learning and Knowledge Discovery in Databases*, pp. 427–443, Cham, 2020. Springer International Publishing, ISBN 978-3-030-43823-4. iv, 2, 23, 27, 30

[7] Aydos, F., Ozbayoglu, A.M., Sirin, Y., and Demirci, M.F.: *Web page classification with Google Image Search results.* ArXiv, abs/2006.00226, 2020. 35

[8] Bakkali, S., Ming, Z., Coustaty, M., and Rusiñol, M.: *Visual and Textual Deep Feature Fusion for Document Image Classification.* In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2394–2403, 2020. 23, 27, 31

[9] Bhowmik, S. and Sarkar, R.: *Classification of Text regions in a Document Image by Analyzing the properties of Connected Components.* In *2020 IEEE Applied Signal Processing Conference (ASPCON)*, pp. 36–40, 2020. 7, 11

[10] Braz, F.A., da Silva, N.C., and Lima, J.A.S.: *Leveraging effectiveness and efficiency in Page Stream Deep Segmentation.* Engineering Applications of Artificial Intelligence, 105:104394, 2021, ISSN 0952-1976. 23, 25, 27, 29, 40, 42

[11] Braz, F.A., Silva, N.C. da, Campos, T.E. de, Chaves, F.B.S., Ferreira, M.H.S., Inazawa, P.H., Coelho, V.H.D., Sukiennik, B.P., Almeida, A.P.G.S. de, Barros Vidal, F. de, Bezerra, D.A., Gusmao, D.B., Ziegler, G.G., Fernandes, R.V.C., Zumblick, R., and Peixoto, F.: *Document classification using a Bi-LSTM to unclog Brazil's supreme court.* ArXiv, abs/1811.11569, 2018. 32

[12] Chen, K., Yin, F., and Liu, C.: *Hybrid Page Segmentation with Efficient Whitespace Rectangles Extraction and Grouping.* In *12th International Conference on Document Analysis and Recognition*, pp. 958–962, 2013. 11

[13] Chen, N. and Blostein, D.: *A survey of document image classification: problem statement, classifier architecture and performance evaluation.* International Journal of Document Analysis and Recognition (IJDAR), 10(1):1–16, 2007. 6

[14] Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., and Slattery, S.: *Learning to Extract Symbolic Knowledge from the World Wide Web.* In *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, AAAI '98/IAAI '98, p. 509–516, USA, 1998. American Association for Artificial Intelligence, ISBN 0262510987. 34

[15] Cui, L., Xu, Y., Lv, T., and Wei, F.: *Document AI: Benchmarks, Models and Applications.* ArXiv, abs/2111.08609, 2021. iii, 4, 5, 21

[16] Devlin, J., Chang, M.W., Lee, K., and Toutanova, K.: *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, vol. 1, p. 4171–4186, 2019. v, 18, 19, 20, 21, 24, 26

[17] Diem, M., Kleber, F., and Sablatnig, R.: *Text Classification and Document Layout Analysis of Paper Fragments.* In *2011 International Conference on Document Analysis and Recognition*, pp. 854–858, 2011. 12

[18] Gorai, M. and Nene, M.J.: *Layout and Text Extraction from Document Images using Neural Networks.* In *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, pp. 1107–1112, 2020. 2

[19] Harley, A.W., Ufkes, A., and Derpanis, K.G.: *Evaluation of Deep Convolutional Nets for Document Image Classification and Retrieval.* In *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 991–995, Aug. 2015. iii, v, 28, 30, 40, 48

[20] Hashemi, M.: *Web Page Classification: A Survey of Perspectives, Gaps, and Future Directions.* Multimedia Tools Appl., 79(17–18):11921–11945, may 2020, ISSN 1380-7501. `https://doi.org/10.1007/s11042-019-08373-8`. 34, 35

[21] Hong, T., Kim, D., Ji, M., Hwang, W., Nam, D., and Park, S.: *BROS: A Pre-trained Language Model Focusing on Text and Layout for Better Key Information Extraction from Documents.* arXiv preprint arXiv:2108.04539, 2021. v, 5, 24, 27

[22] Howard, J. and Gugger, S.: *Deep Learning for Coders with fastai and PyTorch.* O'Reilly Media, 2020. 39

[23] Howard, J. and Ruder, S.: *Universal Language Model Fine-tuning for Text Classification.* In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 328–339, Melbourne, Australia, July 2018. Association for Computational Linguistics. `https://aclanthology.org/P18-1031`. 41

[24] Kosaraju, S.C., Masum, M., Tsaku, N.Z., Patel, P., Bayramoglu, T., Modgil, G., and Kang, M.: *DoT-Net: Document Layout Classification Using Texture-Based CNN.* In *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1029–1034, 2019. 2

[25] Kudo, T. and Richardson, J.: *SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing.* In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71, Brussels, Belgium, Nov. 2018. Association for Computational Linguistics. `https://aclanthology.org/D18-2012`. 40

[26] Kumar, J., Ye, P., and Doermann, D.: *Learning document structure for retrieval and classification.* In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pp. 1558–1561, 2012. v, 28, 31

[27] Le, V.P., Nayef, N., Visani, M., Ogier, J., and Tran, C.D.: *Text and non-text segmentation based on connected component features.* In *13th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1096–1100, 2015. 12

[28] Lewis, D., Agam, G., Argamon, S., Frieder, O., Grossman, D., and Heard, J.: *Building a Test Collection for Complex Document Information Processing.* In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, p. 665–666, New York, NY, USA, 2006. Association for Computing Machinery, ISBN 1595933697. `https://doi.org/10.1145/1148170.1148307`. v, 28, 48

[29] Li, C., Bi, B., Yan, M., Wang, W., Huang, S., Huang, F., and Si, L.: *StructuralLM: Structural Pre-training for Form Understanding.* In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6309–6318, Online, Aug. 2021. Association for Computational Linguistics. `https://aclanthology.org/2021.acl-long.493`. 24, 27

[30] Li, S., Ma, X., Pan, S., Hu, J., Shi, L., and Wang, Q.: *VTLayout: Fusion of Visual and Text Features for Document Layout Analysis*. In *Pacific Rim International Conference on Artificial Intelligence*, pp. 308–322. Springer, 2021. v, 5, 23, 27

[31] Liang, J., Ha, J., Haralick, R.M., and Phillips, I.T.: *Document layout structure extraction using bounding boxes of different entitles*. In *Proceedings Third IEEE Workshop on Applications of Computer Vision. WACV'96*, pp. 278–283, 1996. 10

[32] Liang, X., Cheddad, A., and Hall, J.: *Comparative Study of Layout Analysis of Tabulated Historical Documents*. Big Data Research, 24, May 2021. 5, 10

[33] Luz de Araujo, P.H., de Campos, T.E., and Magalhaes Silva de Sousa, M.: *Inferring the source official texts: can SVM beat ULMFiT?* In *International Conference on the Computational Processing of Portuguese (PROPOR)*, Lecture Notes on Computer Science (LNCS), Evora, Portugal, March 2-4 2020. Springer. `https://propor.di.uevora.pt/`, Code and data available from `https://cic.unb.br/~teodecampos/KnEDLe/`. 28

[34] Maia, A.L.L.M., Julca-Aguilar, F.D., and Hirata, N.S.T.: *A Machine Learning Approach for Graph-Based Page Segmentation*. In *31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pp. 424–431, 2018. 1, 4

[35] Marinai, S.: *Introduction to Document Analysis and Recognition*. In *Machine Learning in Document Analysis and Recognition*, 2008. 12

[36] Merity, S., Keskar, N., and Socher, R.: *Regularizing and Optimizing LSTM Language Models*. In *International Conference on Learning Representations*, pp. 1–13, 2018. `https://openreview.net/forum?id=SyyGPP0TZ`. v, 41

[37] Motahari, H., Duffy, N., Bennett, P., and Bedrax-Weiss, T.: *A Report on the First Workshop on Document Intelligence (DI) at NeurIPS 2019*. SIGKDD Explor. Newsl., 22(2):8–11, jan 2021, ISSN 1931-0145. `https://doi.org/10.1145/3447556.3447563`. 4

[38] Noce, L., Gallo, I., Zamberletti, A., and Calefati, A.: *Embedded Textual Content for Document Image Classification with Convolutional Neural Networks*. In *Proceedings of the 2016 ACM Symposium on Document Engineering*, DocEng '16, p. 165–173, New York, NY, USA, 2016. ISBN 9781450344388. `https://doi.org/10.1145/2960811.2960814`. 2, 14, 31

[39] Pan, Y., Zhao, Q., and Kamata, S.: *Document layout analysis and reading order determination for a reading robot*. In *TENCON 2010 - 2010 IEEE Region 10 Conference*, pp. 1607–1612, 2010. 10

[40] Rothman, D.: *Transformers for Natural Language Processing: Build Innovative Deep Neural Network Architectures for NLP with Python, PyTorch, TensorFlow, BERT, RoBERTa, and More*. Packt Publishing, 2021, ISBN 9781800565791. `https://books.google.com.br/books?id=Ua03zgEACAAJ`. 17

[41] Sah, A.K., Bhowmik, S., Malakar, S., Sarkar, R., Kavallieratou, E., and Vasilopoulos, N.: *Text and non-text recognition using modified HOG descriptor.* In *IEEE Calcutta Conference (CALCON)*, pp. 64–68, 2017. 12

[42] Silva, N.C. da, Braz, F.A., Campos, T.E. de, Guedes, A.L.P., Mendes, D.B., Bezerra, D.A., Gusmao, D.B., Chaves, F.B.S., Ziegler, G.G., Horinouchi, L.H., Ferreira, M.U., Inazawa, P.H., Coelho, V.H.D., Fernandes, R.V.C., Peixoto, F., Filho, M.S.M., Sukiennik, B.P., Rosa, L., Silva, R., Junquilho, T.A., and Carvalho, G.H.T.: *Document type classification for Brazil's supreme court using a Convolutional Neural Network.* Proceedings of The Tenth International Conference on Forensic Computer Science and Cyber Law, 2018. 32

[43] Smith, R. *et al.*: *Tesseract ocr engine.* Lecture. Google Code. Google Inc, 2007. 38

[44] Smith, R.W.: *Hybrid Page Layout Analysis via Tab-Stop Detection.* In *10th International Conference on Document Analysis and Recognition*, pp. 241–245, 2009. 11

[45] Sundermeyer, M., Schlüter, R., and Ney, H.: *LSTM neural networks for language modeling.* In *Thirteenth annual conference of the international speech communication association*, pp. –, 2012. v, 15, 41

[46] Tran, T.A., Na, I.S., and Kim, S.H.: *Page Segmentation Using Minimum Homogeneity Algorithm and Adaptive Mathematical Morphology.* Int. J. Doc. Anal. Recognit., 19(3):191–209, Sept. 2016, ISSN 1433-2833. `https://doi.org/10.1007/s10032-016-0265-3`. 10, 11

[47] Tran, T.A., Nguyen-An, K., and Quang Vo, N.: *Document Layout Analysis: A Maximum Homogeneous Region Approach.* In *1st International Conference on Multimedia Analysis and Pattern Recognition (MAPR)*, pp. 1–5, 2018. 11

[48] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., and Polosukhin, I.: *Attention is All you Need.* In Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.): *Advances in Neural Information Processing Systems*, vol. 30, pp. –. Curran Associates, Inc., 2017. `https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`. iii, v, 5, 16, 17, 24, 26

[49] Verdonck, T., Baesens, B., Óskarsdóttir, M., and Broucke, S. vanden: *Special issue on feature engineering editorial.* Machine Learning, pp. 1–12, Aug. 2021. 11

[50] Wei, J., Chu, X., Sun, X.Y., Xu, K., Deng, H.X., Chen, J., Wei, Z., and Lei, M.: *Machine learning in materials science.* InfoMat, 1(3):338–358, 2019. 11

[51] Wiedemann, G. and Heyer, G.: *Multi-Modal Page Stream Segmentation with Convolutional Neural Networks.* Lang. Resour. Eval., 55(1):127–150, mar 2021, ISSN 1574-020X. `https://doi.org/10.1007/s10579-019-09476-2`. iv, 23, 27, 29

[52] Wu, T.L., Li, C., Zhang, M., Chen, T., Hombaiah, S.A., and Bendersky, M.: *LAMPreT: Layout-Aware Multimodal PreTraining for Document Understanding.* ArXiv, abs/2104.08405, 2021. iv, 24, 27

[53] Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., and Zhou, M.: *LayoutLM: Pre-training of Text and Layout for Document Image Understanding.* Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Aug 2020. `http://dx.doi.org/10.1145/3394486.3403172`. iv, v, 1, 5, 6, 19, 20, 21, 24, 26, 27, 30

[54] Xu, Y., Xu, Y., Lv, T., Cui, L., Wei, F., Wang, G., Lu, Y., Florencio, D., Zhang, C., Che, W., Zhang, M., and Zhou, L.: *LayoutLMv2: Multi-modal Pre-training for Visually-rich Document Understanding.* In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 2579–2591, Online, Aug. 2021. Association for Computational Linguistics. `https://aclanthology.org/2021.acl-long.201`. v, 5, 22, 25, 27, 30

[55] Zhu, G. and Doermann, D.: *Automatic Document Logo Detection.* In *In Proc. 9th International Conf. Document Analysis and Recognition (ICDAR 2007)*, pp. 864–868, 2007. 28

[56] Zhu, G., Zheng, Y., Doermann, D., and Jaeger, S.: *Multi-scale Structural Saliency for Signature Detection.* In *In Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR 2007)*, pp. 1–8, 2007. iii, 28

[57] Zulfiqar, A., Ul-Hasan, A., and Shafait, F.: *Logical Layout Analysis using Deep Learning.* In *2019 Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–5, 2019. 4, 13