ALMA MATER STUDIORUM – UNIVERSITY OF BOLOGNA

Computer Science and Engineering
Master Degree in Artificial Intelligence

DEVELOPMENT OF AN ARTIFICIAL INTELLIGENCE-BASED SOLUTION FOR DOCUMENT PROCESSING AUTOMATION USING MACHINE LEARNING AND NLP TECHNIQUES

Dissertation in
Machine Learning and Computer Vision

Supervising Professor
Prof.Ing. Claudio Sartori

Co-relatori
Data Specialist Alex Magrini
Dott. Marco Aspromonte

Presented by
Biniam Abraha Masa

Third Graduation Session
Year 2022/2023

KEYWORDS

Intelligent Document Processing
Optical Character Recognition
invoices
Named Entity Recognition
Prodigy

0.1 Abstract

The proposal focuses on Intelligent Document Processing (IDP), which aims to automate various activities related to document processing using Artificial Intelligence technologies, particularly Machine Learning and Natural Language Processing techniques. The proposed solution seeks to improve the efficiency and quality of document processing in many business and organizational contexts by automating tasks such as classification, information extraction, validation, and verification of consistency between documents. It aims to deliver seamless searching of entity names in commercial invoice systems using Optical Character Recognition (OCR) and Natural Language Processing (NLP). This thesis paper includes the following phases: "Text Identification, OCR, Invoice Data Extraction and Quality Assurance". In case of document files, the data extraction is done in the first phase.

This project thesis details the IDP solution developed, analyse processing results and the quality of the extracted information, and evaluate the accuracy and efficiency of the system. Furthermore, it will compare the developed IDP solution with other solutions available in the market, evaluating their advantages and disadvantages. The thesis is focused on information extraction from key fields of invoices using two different methods based on sequence labeling. Invoices are semi-structured documents in which data can be located based on the context. Their performances are expected to be generally high on documents they have been trained for but processing new templates often requires new manual annotations like prodigy tool, which is tedious and time-consuming to produce labeled data. This showcases a set of trials utilizing neural networks methods to examine the balance between data prerequisites and efficacy in retrieving data from crucial sections of invoices (such as invoice date, invoice number, order number, amount, supplier's name...). The main contribution of this thesis is a system that achieves competitive results using a small amount of data compared to the state-of-the-art systems that need to be trained on large datasets, using a custom Named Entity Recognition (NER) model to extract that relevant information from commercial invoice formats. And Compare different Optical Recognition Character (OCR) framework to evaluate which of the candidates performs better on the type of invoice document. Optimize quality of document to improve OCR performance. In summary, the project seeks to develop and evaluate an innovative solution for the automation of document processing using Artificial Intelligence technologies. The proposed solution has the potential to improve the efficiency and quality of document processing in many business and organizational contexts.

Index

Chapter 1

Introduction

1.1 Background and motivation

The background and motivation part of this thesis provides an overview of the importance, and motivating factors for the creation of an artificial intelligence-based solution for automating document processing by using machine learning and natural language processing (NLP) techniques.
A growing volume of documents in nowadays digital age brings huge challenges for businesses in processing and extracting pertinent information from documents. Manual document processing is a prone to mistake and time-consuming and encounters less productive. For this reason, there is an increasing request for automated solution that can smooths document processing work and ameliorate overall efficiency.
The motivation for this research stems from the need to address the limitations of conventional document processing techniques and strengthens the advancements in AI, machine learning, and NLP to develop more intelligent and efficient solutions. By automating document processing tasks, companies can accomplish faster turnaround times, increased accuracy, decreased manual effort, and enhanced decision-making capabilities. Commercial invoices present a distinctive set of challenges due to their irregular formats or non-uniform formats and diverse layouts. Extracting underlying information, such as buyer name, invoice numbers, order number, supplier name, invoice date, and payment terms, from these invoices requires sophisticated methods that can understand and accurately interpret the essential data.

The thesis sights to close the gap between conventional document processing method and up to date AI-driven approaches using a custom Named Entity Recognition (NER) model. The NER model will be trained to recognize and extract the relevant labels from commercial invoices, which will enable the automation of data extraction process. Additionally, the thesis looks into the

use of different Optical Character Recognition (OCR) frameworks to utilize numerous document processing pipeline. By automating document processing and making data extraction faster and precise, businesses can extract valuable insights from the processed data. Better decisions can be made as a result of these insights, which also facilitate regulatory compliance, enable data-driven strategies, and support various business processes such as auditing, supply chain management, and accounting.

By understanding the background and motivation behind this investigation we can realize the underlying development of an ai driven resolution for automating document processing this research undertaking seeks to contribute value to the domain by controlling the constraints of conventional methods and capitalizing on ai technologies to unleash the full capabilities of document information driving operational effectiveness and informed decision making in the business sphere.

1.2 Problem statement

I had a collection of around 320,000 PDF files which are basically invoices which I obtained them from the DocILE repository created by Rossumai. From those I selected at least 10 documents per provider or Supplier totalling 15 different providers. The objective is I must extract meta-data information from these files. Meta-data or the entities inside the invoice PDF file I categorized them as Header and Row. The header entities itself located separately for example the entity name, "Supplier Name" which could be located on left-right or sometimes at the bottom; "Buyer Name, Buyer Address" - which is typically placed together toward the middle of the left; "Invoice number, Invoice Date, Order number, Expiration Date, Buyer P.IVA, Currency" usually located opposite to the Supplier name. The rows meta-data including "Description, Amount, Cost, Quantity" they are altogether found on a table structure below the header.

- Input -> PDF or OCR's text
- Output -> Supplier Name = "ABC"; Buyer Address = "DEF"; Invoice Date = "XX-XX-XX"

What I have done: I had passed these documents into Tesseract's OCR to generate machine readable text output file format. This text dataset contains character-wise coordinates.

What is not feasible, or challenges faced: There was some not well-prepared extracted text due to the structure and blurred effect occurred on the invoice

pdf. For this reason, when annotating the invoice entities, it was difficult to find or match the exact required information. In addition to that due to a large amount of data it made the process very confusing to search manually and annotate them. Occasionally, the labels were not fully extracted when using the prodigy tool to those text files for training the model. Json file format a combination of tesseract text extraction that we find into the key text and NER spaCy model that give us the spans section.

One significant issue I observed was some entities or labels such as "currency, cost, Buyer P.IVA, Row Total, Quantity" were absent or the annotated number rate is much lower compared to the other entities.

1.3 Research objectives

The goal of this thesis is to identify the challenges when implementing an Intelligent document process in business and propose a machine learning model and NLP that is easy to implement and maintain and increases efficiency and accuracy of document processing to improve business productivity to reduce manual effort. This thesis also includes a performance evaluation of the proposed custom NER model by comparing it with commonly used rule-based classifiers. Finally, this thesis will aim to address the following task:

- Data collection from GitHub DocILE repository created by Rossum
- Building the dataset with respect to various providers having equal documents
- Building custom OCR system to feed documents to enable converting to a machine readable text to make use them initiate the labeling process.
- Understanding the scope of various available metadata in the invoices
- Utilization of prodigy tool to annotate and labeling the invoice entity names.
- Training the built model using the labeled extracted prepared dataset resources from prodigy with NLP framework
- Evaluating the performance of the trained model and steps to improve the accuracy of the model.

While working with the first point of acquiring data to my local space, to broaden my understanding I used to do scouting on the company platform called

confluence exploring the NLP frameworks spaCy, OCR system and getting ready to familiarize with the company suggested or offered famous software called prodigy. The valuable contribution the thesis has been carried out as a research study conducted in Cloudif S.r.l a company based in Bologna that develops Artificial Intelligence Solutions, with the supervision of Alex Magrini and Dot. Marco Aspromonte, to whom I extend my sincere appreciation. All the experimental work was made possible throughout their invaluable support and guidance alongside Ing. Claudio Sartori's.

1.4 Scope and limitations

This part details the boundaries and constraints of the research study and explains a clear understanding of a covered areas and the possible restrictions of the results to help set clear expectations and give the context for interpreting the findings.

The scope of this research revolves around the development of an artificial intelligence-based solution for document processing automation using machine learning and natural language processing (NLP) techniques. Specifically, the focus is on addressing the challenges associated with extracting relevant information from commercial invoice formats. The primary objective is to build a custom Named Entity Recognition (NER) model that can accurately extract key information from invoices. This involves collecting a dataset from a repository and obtaining Optical Character Recognition (OCR) outputs from the Docusense server machine. The relevant entities, such as invoice number, invoice date, order number, currency, amount, buyer details, supplier details, and payment terms, are identified and compiled. To train the NER model, the dataset is labeled using tools like Prodigy for drawing bounding boxes and annotating the entities. The model is implemented using the spaCy framework, and its performance is evaluated by collecting metrics and analyzing the results. If the performance is not satisfactory, additional documents are added to the dataset, and the process is repeated.

Furthermore, the research includes a comparison of different OCR frameworks, such as Easy OCR, Paddle OCR, and Tesseract. The NER model's performance is assessed on the outputs of these OCR systems, allowing for a comparative analysis of their effectiveness in the document processing pipeline. Denoise techniques, including Microsoft Old Photo Restoration and Gaussian Noise reduction, are employed to enhance the quality of invoice images, thereby facilitating more accurate OCR results. While striving to achieve these research objectives, it is important to acknowledge certain limitations. The availability and quality of training data may impact the performance and generalizabil-

ity of the NER model. Additionally, the evaluation and applicability of the proposed solution may vary depending on the specific domain and types of documents being processed. Real-world scenarios and complexities may not be fully covered, and certain assumptions and simplifications are made during the research. Practical implementation challenges, such as system integration and resource limitations, also need to be considered.

1.5 Thesis organization

The thesis is organized and outlined into the following sections showing the logical flow and highlighting overall structure of the thesis with their respective content.

- **Chapter 1:** This chapter introduces Background and motivation behind the development of an artificial intelligence-based solution to document processing using machine learning and natural language processing, Problem statement, Research objectives in Cloudif Task, Scope and limitations and Thesis organization.

- **Chapter 2:** this chapter provides overview of the research relevant to document processing automation, NLP techniques, machine learning, Named entity Recognition and Optical Character Recognition frameworks in the context of document processing automation including EasyOCR, paddle OCR and Tesseract.

- **Chapter 3:** this chapter presents the process of data collection from a repository mainly worked with. And how the data is obtained and highlights a followed steps to extract document using given token and the pre-processed steps to apply to the raw documents for further analysis.

- **Chapter 4:** this chapter focuses on the development of the model for commercial invoices. Its highlight's with introduction of custom NER and its importance in processing documents. Tells the process of building labeled dataset followed by the implementation of custom NER model using spaCy and evaluation and performance metrics.

- **Chapter 5:** this chapter explores setting up the software and creation of the file for prodigy. Explaining the workflow and methodology used to annotate the data and mentioning some observation and challenges encountered during the process.

- **Chapter 6:** this chapter conducts the performance evaluation of the developed system and presentation analysis of experimental results including discussion of findings.

Chapter 2

Literature Review

2.1 Overview of document processing automation

Document processing automation is always a strategy for business executives to improve operational efficiency. With Optical Character Recognition (OCR) and machine learning techniques, businesses can apply Artificial Intelligence (AI) to automate the process [1].

Its significance in business operations, companies use documents to communicate ideas, transact business, and store agreements with external and internal parties. Typical business document categories are invoices, purchase orders, sales agreements, and tax forms. Processing business documents still relies heavily on manual effort to classify the documents and extract the information – a costly operation. The function of document processing becomes a key driver to improve operational efficiency and reduce costs. Current approaches for reducing the cost of document processing can be performed through business process outsourcing [2] or in-house [3]. Through outsourcing, businesses can have several additional advantages, such as focusing on core strategic areas, but face some challenges, like risks of exposing confidential data and management difficulties [3]. A typical document processing cycle within an organization includes receiving documents, sorting documents, pre-processing documents, and dispatching documents [4]. The process owner receives the document and conducts the corresponding transactions. The goal for businesses is to automate this repetitive operation. Optical Character Recognition (OCR), workflow system, and machine learning techniques are the key technologies to build automatic document processing[5]. Additionally, Natural Language Processing (NLP) techniques are widely used to understand the content of business documents [6]. Lastly, Computer vision and image 9 processing techniques

are often necessary preprocessing tools for building an AI based automatic document processing[5].

Mentioning Challenges and inefficiencies in manual document processing. Through outsourcing, businesses can have several additional advantages, such as focusing on core strategic areas, but face some challenges, like risks of exposing confidential data and management difficulties [1]. Several applications were developed in the 1990s [7], [8]; however, these applications are not scalable, and they are limited to particular organizations. Humanly document processing relies on strong effort leading to increased costs and time consumption and which could also possibly occur a potential error.

Handling efficiency of documents in automation to errors on this thesis proposes a machine learning NER model and a Word2vec embeddings as document features to classify business document from unstructured text into structured text using a trained labeled data and reaches 0.863 Macro F1-score using scanned business documents from DocILE GitHub available dataset. Document processing automation is used in real world areas mainly as data extraction, contract management, in different type of documents such as invoice, reports, surveys in saving time using a machine learning techniques to recognize a specific data point.

2.2 Machine learning and NLP techniques in document processing

Deep learning models, such as convolutional neural network (CNN) is applied to the OCR area [9].

Word embedding is a Natural Language Processing (NLP) technique using vectors to represent the semantic meanings of the words [10]. Word2vec was proposed by Google in 2013. There are two types of training models for Word2vec: continuous bag-of-words (CBOW) and continuous skip-gram. CBOW uses surrounding words to predict the current word. Skip-gram uses the current word to predict the surrounding words. There are two common ways, average and sum, to construct document embeddings from individual word embeddings [11]. The average operator is used to approximate the document embeddings. Some industry specific application of NLP and machine learning at document processing in healthcare used for medical records, patient data and clinical docs where and in finance sector invoice processing , fraud detection and loan application process.

2.3 Named Entity Recognition (NER) models and techniques

Named entity recognition serves as a bridge between unstructured text and structured data, enabling machines to sift through vast amounts of textual information and extract nuggets of valuable data in categorized forms. By pinpointing specific entities within a sea of words, NER transforms the way we process and utilize textual data. NER's primary objective is to comb through unstructured text and identify specific chunks as named entities, subsequently classifying them into predefined categories. This conversion of raw text into structured information makes data more actionable, facilitating tasks like data analysis, information retrieval, and knowledge graph construction.[12].

NER model techniques or approaches used are Deep Learning-based approach and use recurrent neural network or convolutional neural networks to learn context from text. A deep learning-based NER system example is a spaCy that used to identify and classify named entities in text. And a CNN an approach to NER used for image classification tasks which can also learn local feature form text.

2.4 Optical Character Recognition (OCR) frameworks

Understanding OCR system and framework PaddleOCR, EasyOCR, and tesseract. OCR also called Optical Character Reader is a system that provides a full alphanumeric recognition of printed or handwritten characters at electronic speed by simply scanning the form. More recently, the term Intelligent Character Recognition (ICR) has been used to describe the process of interpreting image data, in particular alphanumeric text. Forms containing characters images can be scanned through scanner and then recognition engine of the OCR system interpret the images and turn images of handwritten or printed characters into ASCII data (machine-readable characters). The technology provides a complete form processing and documents capture solution.

Usually, OCR uses a modular architecture that is open, scalable and workflow controlled. It includes forms definition, scanning, image pre-processing, and recognition capabilities. Intelligent Character Recognition (ICR) is the module of OCR that has the ability to turn images of handwritten or printed characters into ASCII data, sometimes also known as OCR. ICR and OCR are recognition engines used with imaging; while OMR is a data collection technology that

does not require a recognition engine. Therefore, basically, OMR can not recognize hand-printed or machine-printed characters. However, in the OCR technology, answer for question in "tick" or "mark" is also known as OCR. The most common scenario for OCR is the printed/pdf OCR. The structured nature of printed documents makes it much easier to parse them. Most OCR tools (e.g., Tesseract) are mostly intended to address this task and achieve good result.

OCR also called Optical Character Reader is a system that provides a full alphanumeric recognition of printed or handwritten characters at electronic speed by simply scanning the form. More recently, the term Intelligent Character Recognition (ICR) has been used to describe the process of interpreting image data, in particular alphanumeric text.

Forms containing characters images can be scanned through scanner and then recognition engine of the OCR system interpret the images and turn images of handwritten or printed characters into ASCII data (machine-readable characters). The technology provides a complete form processing and documents capture solution. Usually, OCR uses a modular architecture that is open, scalable and workflow controlled. It includes forms definition, scanning, image pre-processing, and recognition.

OCR enables conversion of scanned documents into readable text. Processing invoice using OCR scans or uploads the invoice to an OCR-based invoice processing solution. It extracts key data or metadata from the invoice such supplier name, invoice number, invoice data...etc.

Invoice can be used to extract metadata, validate the extracted invoice data. While our world largely exists in the digital realm, most businesses still use print media. This includes documents such as invoices, contracts, scanned legal documents, and other paper forms. Scanning documents into images can be time-consuming as it requires manual input. OCR saves individuals and businesses time and money by converting images into text data that is able to be read by other business software.

Types of OCR

- Simple optical character recognition software stores different text and font image patterns as templates. This software will use pattern dash-matching algorithms to find the differences between text images. It will analyze character by character in its internal database. Optical word recognition is when the system replicates the text word by word. It is not possible for every font and handwriting style to be captured as there

are unlimited amounts of both, so this solution has its limits.

- Intelligent character recognition (ICR) software is a part of modern OCR technologies. ICR reads text the same way humans read it. Using machine learning software, machines can be trained to act like humans. A machine learning system called a neural network studies text and processes images repeatedly. It searches for image aspects such as lines, curves, loops, and intersections and puts together the outcome of the different levels of data to get a final conclusion.

- Intelligent word recognition technologies work on the same rules as ICR, but those technologies study whole word images rather than pre-modifying the images into characters.

- Optical mark recognition finds watermarks, logos, and other text signs in a document.

OCR and machine learning have grown exponentially over the past couple of decades and will only continue to see improvements over the coming years. The next generation of OCR is built using machine learning and artificial intelligence that is not limited to the character-matching software of previous software. OCR software will continue to think and learn more on its own. Not only will OCR technology continue to perceive scanned text, but it will also find the text's meaning and make sense of the content. Machine learning may be an idea of the past as deep learning continues to develop and transform OCR technologies. Deep learning technologies are composed of neural networks that imitate human brain functionality to verify that algorithms don't need to depend on historical patterns to confirm accuracy. Deep learning means the technology can do this on its own and not only see the text but find the meaning behind it.

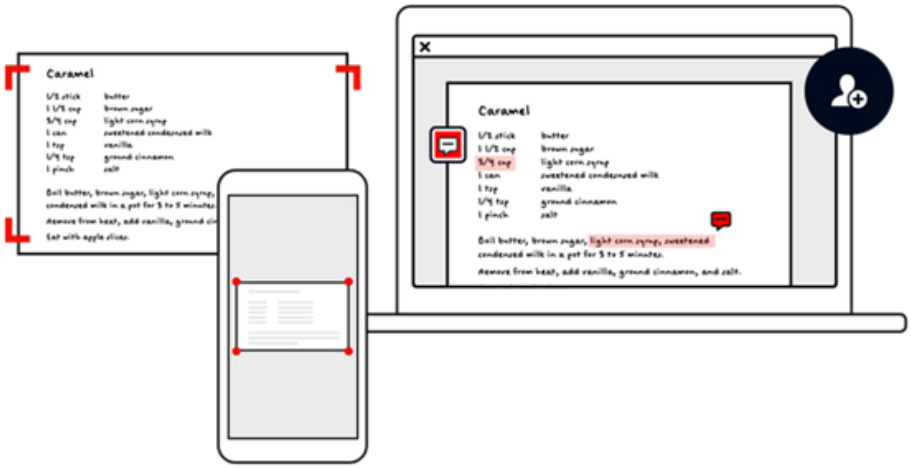


Figure 2.1: OCR System

2.5 EasyOCR, PaddleOCR, and Tesseract frameworks

Comparison of the performance of NER model on the output of these three frameworks.

1.Tesseract OCR:

Ease of use Tesseract is an open-source OCR engine actively developed by Google. Tesseract OCR is known for its high accuracy and extensive language support. Provides command line tools and APIs for various programming languages. Tesseract supports multiple languages and offers options for page layout analysis, text recognition and post-processing. Adaptability Tesseract can adapt by training on a specific dataset.

You can tune his OCR engine using the Tesseracts training tool, but this requires expertise and effort. It has more extensive customization options than EasyOCR but requires more technical knowledge. Tesseract is widely recognized as a reliable OCR engine with excellent performance. It has evolved over the years and provides reasonable accuracy, especially when combined with suitable pre-treatment techniques.

Main advantage: Extensive language support: Tesseract OCR supports over 100 languages, making it suitable for applications requiring multilingual support.

Accurate: Tesseract OCR has achieved the highest performance in various OCR benchmarks, making it a highly reliable OCR system. user friendly: Tesseract OCR has a simple interface and can be easily integrated into your application. Limitation Limited Image Pre-processing Tesseract OCR relies heavily on image pre-processing techniques to improve accuracy, which can be time and resource consuming. Comparison To compare these OCR methods, we evaluated accuracy, speed, language support, customization options, and community support. Accuracy All three OCR systems achieved high accuracy in various OCR benchmarks. Both Paddle OCR and KerasOCR achieved the best performance in various benchmarks, and EasyOCR also achieved high accuracy.

2.Paddle OCR:

Usability PaddleOCR is a deep learning-based OCR framework developed by PaddlePaddle, a Chinese AI company. Paddle OCR is based on the PaddlePaddle framework, known for its fast and efficient deep learning algorithms. It supports dozens of languages, including Chinese, English, Japanese, and Korean, and can correctly recognize different text styles and fonts. It provides an easy-to-use API and supports multiple languages. It provides pre-trained models for various OCR tasks such as text recognition, recognition, and layout analysis. PaddleOCR allows for customization by fine-tuning the pre-trained model to a specific dataset. Tools and documentation are provided to guide users through the customization process. PaddleOCR is known for its accuracy and robust performance.

It leverages deep learning models such as CRNN and Transformer to provide state-of-the-art performance on various OCR tasks.

main advantage: Highly accurate paddle OCR achieved top performance in various OCR benchmarks including ICDAR 2015 and ICDAR 2017 competitions. And it is Fast and Efficient. Paddle OCR is optimized for speed and can process large numbers of images in real time, making it suitable for applications requiring high throughput.

Limitations: Limited Language Support Paddle OCR supports multiple languages, but not as many as some of our competitors. Limited Community Support Paddle OCR is a relatively new OCR system, and its community is not as large as some of its competitors, making it difficult to find resources and support.

3.Easy OCR:

Ease of use EasyOCR is a Python library that aims to provide a simple and straightforward OCR solution.

It supports multiple languages over 70 languages and provides an easy-to-use API for text recognition and recognition.

EasyOCR does not offer extensive customization options by default. However, we can refine the underlying model or incorporate additional preprocessing steps to improve the results. However, this requires more advanced knowledge and implementation. It performs well in terms of accuracy and speed. It uses pre-trained models based on CRNN architecture and provides good results for various OCR tasks.

Main advantages: Fast and Efficient. EasyOCR is optimized for speed and can process large amounts of images in real-time. Easy to use EasyOCR has a simple user interface and can be easily integrated into Python applications.

limitation: Limited customization: EasyOCR doesn't offer as many customization options as some of its competitors, making it difficult to fine-tune your model. Limited language support: EasyOCR supports over 70 languages, but it's not as comprehensive as some of its competitors.

2.6 Quality optimization techniques for document processing

Techniques for improving the quality of documents automation through AI using OCR, machine learning and NLP can automate businesses can use them for classification and extraction of metadata from documents to reduce manual efforts and maximize accuracy. Machine learning NER using spaCy model use document embeddings such as Word2vec for representation of documents and make predictions.

To improve reliability of extracted data techniques such as spell checking and entity validation in which they involve extracted text for spelling errors and correcting them and validating labels such as invoice name , invoice date, supplier name ...etc.

Chapter 3

Data Collection and Preprocessing

3.1 Selection of dataset from the repository

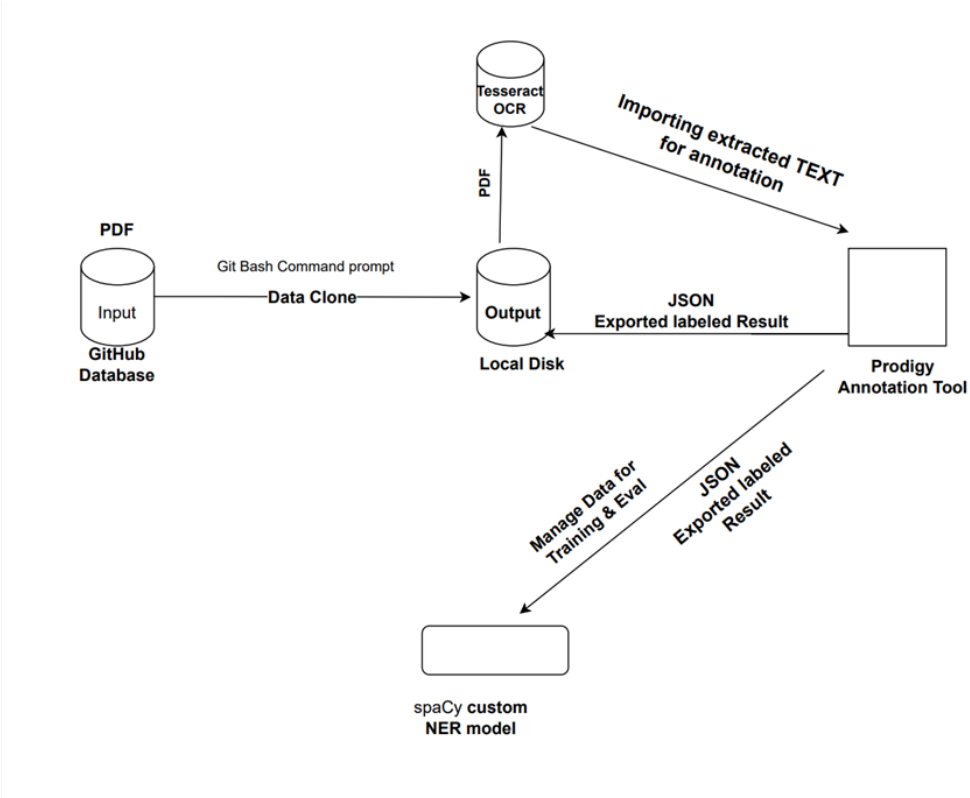


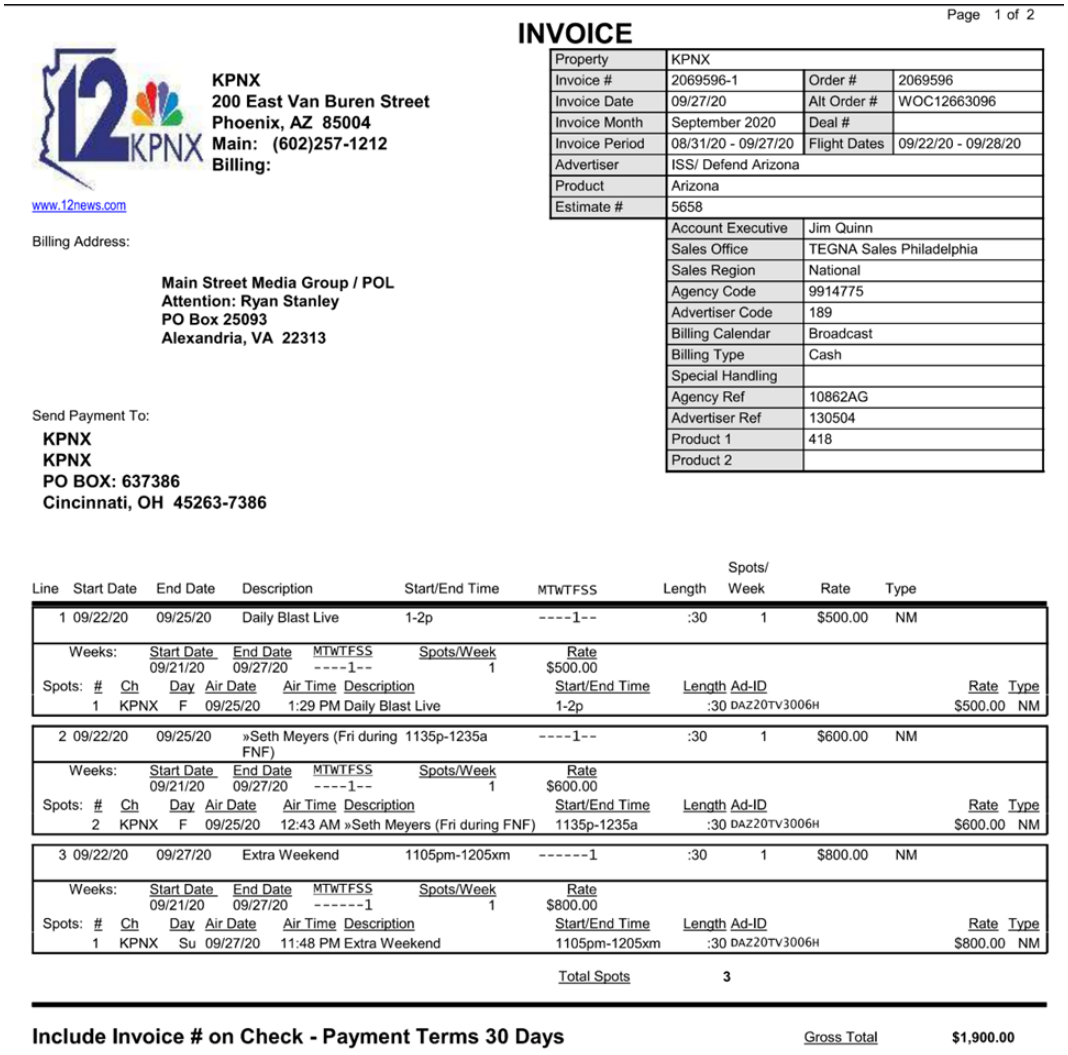
Figure 3.1: Process of gathering and labeling data for training, validating, and testing the model

I obtained the invoice dataset, the repository I used the source is publicly available in the GitHub.

As of my research is oriented to extract a relevant data from contexts of any publicly given and to identify the related or needed text for automatic process, I preferred this invoice dataset as it aligned with my task to provide intelligent document processing. I chose these datasets as they have different template and are sourced from Rossum.ai a known cloud-based OCR with quality and reliable datasets and is able to investigate the relevant data from them. For example, invoice desired named entities such as invoice number, order number ...etc.

The diversity of the dataset size is huge and contains many various invoice formats which is essential for analysis. Not only this it was accessible for my research work. For this case this docile invoice dataset is characterized or mirrored to different challenge that could be faced in different scenarios at document processing automation or OCR and is essential to the real-world application and is characterized with complex layouts and widely distributed used language are English 60%, German 20%, French 10% and other 10%. There are other datasets alternative to Docile, but they lack the diversity, for example some could only have required entity names and are aimed at specific template or geolocation or companies. There are some use cases or existed research that leveraged the DocILE invoice dataset as this data is used to evaluate performance of different line-item recognition (LIR) in the ICDAR 2023 and in the CLEF 2023 lab competition on document information localization. I am using this DocILE dataset to build custom NER model using spaCy library to process efficient than existing model and able me to detect errors and correct information from invoices. DocILE is widely recognized in many research communities.

The DocILE has too many numbers of invoices scaled around 100k invoices, with purchase invoices type from large companies' non-profit organization government agencies, and industries of finance, healthcare...etc. They are of a combination of both old and recent years datasets. Before making us of it required a cleaning or standardization steps.



Page 1 of 2

INVOICE

Property	KPNX		
Invoice #	2069596-1	Order #	2069596
Invoice Date	09/27/20	Alt Order #	WOC12663096
Invoice Month	September 2020	Deal #	
Invoice Period	08/31/20 - 09/27/20	Flight Dates	09/22/20 - 09/28/20
Advertiser	ISS/ Defend Arizona		
Product	Arizona		
Estimate #	5658		

Account Executive	Jim Quinn
Sales Office	TEGNA Sales Philadelphia
Sales Region	National
Agency Code	9914775
Advertiser Code	189
Billing Calendar	Broadcast
Billing Type	Cash
Special Handling	
Agency Ref	10862AG
Advertiser Ref	130504
Product 1	418
Product 2	

KPNX
200 East Van Buren Street
Phoenix, AZ 85004
Main: (602)257-1212
Billing:

www.12news.com

Billing Address:

Main Street Media Group / POL
Attention: Ryan Stanley
PO Box 25093
Alexandria, VA 22313

Send Payment To:

KPNX
KPNX
PO BOX: 637386
Cincinnati, OH 45263-7386

Line	Start Date	End Date	Description	Start/End Time	MTWTFSS	Length	Spots/ Week	Rate	Type
1	09/22/20	09/25/20	Daily Blast Live	1-2p	----1--	:30	1	$500.00	NM

Weeks:	Start Date	End Date	MTWTFSS	Spots/Week	Rate
	09/21/20	09/27/20	----1--	1	$500.00

Spots: #	Ch	Day	Air Date	Air Time	Description	Start/End Time	Length	Ad-ID	Rate	Type
1	KPNX	F	09/25/20	1:29 PM	Daily Blast Live	1-2p	:30	DAZ20TV3006H	$500.00	NM

2	09/22/20	09/25/20	»Seth Meyers (Fri during FNF)	1135p-1235a	----1--	:30	1	$600.00	NM

Weeks:	Start Date	End Date	MTWTFSS	Spots/Week	Rate
	09/21/20	09/27/20	----1--	1	$600.00

Spots: #	Ch	Day	Air Date	Air Time	Description	Start/End Time	Length	Ad-ID	Rate	Type
2	KPNX	F	09/25/20	12:43 AM	»Seth Meyers (Fri during FNF)	1135p-1235a	:30	DAZ20TV3006H	$600.00	NM

3	09/22/20	09/27/20	Extra Weekend	1105pm-1205xm	------1	:30	1	$800.00	NM

Weeks:	Start Date	End Date	MTWTFSS	Spots/Week	Rate
	09/21/20	09/27/20	------1	1	$800.00

Spots: #	Ch	Day	Air Date	Air Time	Description	Start/End Time	Length	Ad-ID	Rate	Type
1	KPNX	Su	09/27/20	11:48 PM	Extra Weekend	1105pm-1205xm	:30	DAZ20TV3006H	$800.00	NM

Total Spots 3

Include Invoice # on Check - Payment Terms 30 Days Gross Total **$1,900.00**

Figure 3.2: Sample invoice to illustrate the invoice entity information

In the DocILE invoice dataset, the visualization both invoice type and different language distribution shows as of the invoice type standard invoice is 80% and Credit note 10%, Debit note 5% and other 5%. It has large scale size compared to another invoice dataset. As DocILE has 100,000 number of invoices, InvoiceNet 30,000, and InvoiceParser 20,000, this makes it worth resources for developing machine learning methods. It contains sensitive and personal data like invoice dates and buyer names and buyer addresses ...etc, but to measure the privacy concern its anonymized before it publicly released and is only open to researcher signed agreement.

They were unstructured and provided in PDF and some were scanned. They

need to be structured to learn relationship between different attributes so that easier to be extracted the relevant data of every entity names. We used Json architecture and tesseract OCR tool extracts into text approach to extract from PDF using script *extract_ test.py* and converting to Json format. I did not use a pretrained machine learning model to extract data from PDF.

The data are publicly accessible and found in the repository using a free access token and be cloned the dataset in private local machine:

https://github.com/rossumai/docile

They are available for research aim and responsibly used to develop machine learning solutions. It is provided by Rossum and can be downloaded from their site or repository.

To verify the authenticity, I checked the Rossum is large company of cloud based which provides machine learning based solution. And is featured in publications such as wall street Journal and Forbes.to ensure integrity of data i downloaded through official repository which maintained by creator Rossum. As long as it is licensed or used only for data use agreement, I only used for my invoice data processing. I did data sampling by taking random 5000 invoices from dataset. I used some criteria to validate the sampled invoice among them is buyer name, buyer address, invoice date, payment terms, expiration date and invoice date are valid and unique. Of this reason and in its documentation which shows its origin and data format collection technique amplifies the data was authentic and reliable. Compared to other dataset as it has credit and debit notes other than standard invoice.

In order to access the data, I used a given secret token from my supervisor to fully grant to access to download and make use of it from the official repository using Git bash command prompt in my own machine, before I clone the directory to Git bash command as we see in the screenshot taken.

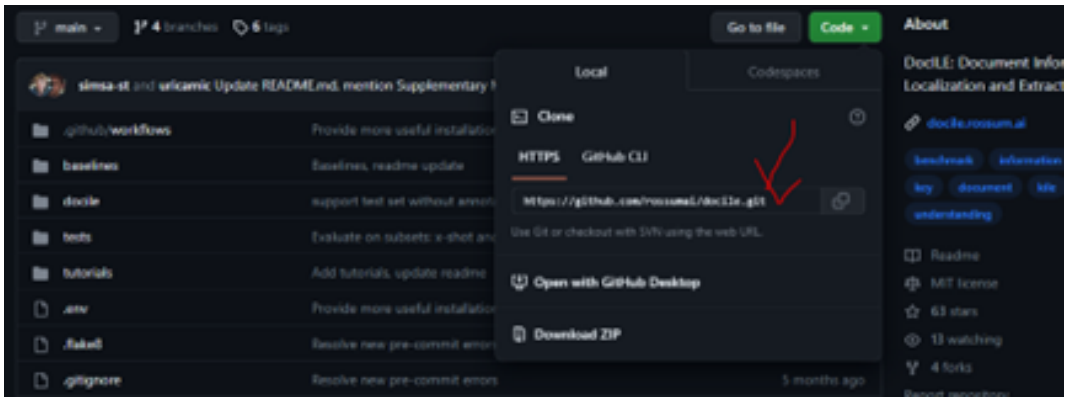


Figure 3.3: GitHub link to clone data

The license type was proprietary license type and is subjected to data use agreement owned by Rossum which means allowed people able to access it and restricted to share publicly. For such reason and ethical and legal use of it I stored in secure personal location PC never been shared publicly. So, my research is conducted using the official DocILE dataset collected by author and publisher Rossum in title DocILE invoice dataset.

Data preprocessing was needed to clean up the outliers and anomalies. Data duplicates missing values standardizing data format has processed to achieve cleaned desired data. Some entity has many duplicate values, redundant and according been removed. To achieve missing values some additional original data has been augmented from the source. I used different techniques to manage cleaning available data using such as fuzzywuzzy library to search lines having same and remove duplicate those who has similar threshold values *remove_ duplicates.py* and some other shuffling and list-based deduplication techniques. To see inconsistencies and data transformation and check normality of the data, I used visualization techniques such as bar plot using matplotlib to deal and assess with outliers so that I transform them to desired consistent data.

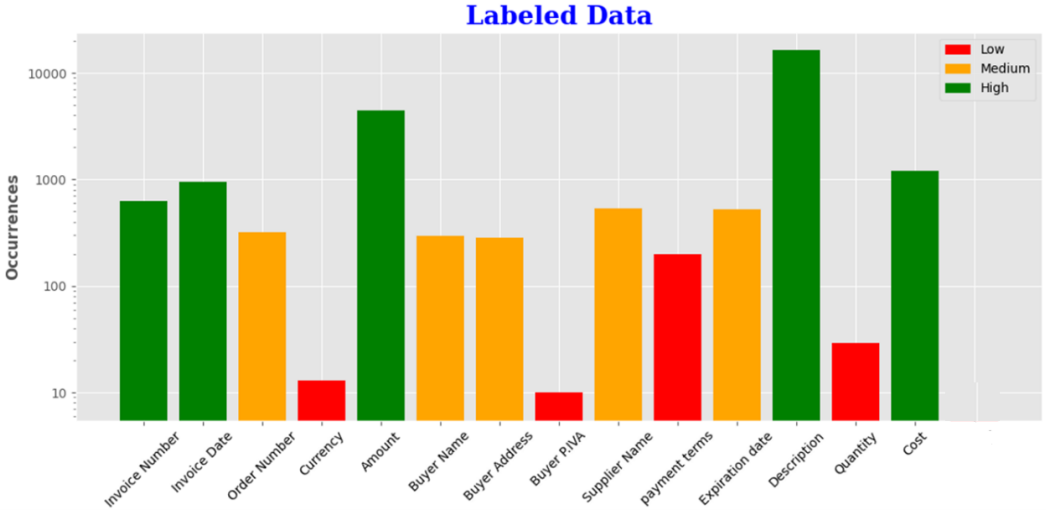


Figure 3.4: Data inconsistencies

There was a standardization taken place with some of the labels such as currency symbol I converted each worldwide symbols to consistent format with unique representation. To handle missing value which was not evenly distributed I used numerical data and categorical data by dropping incomplete records such as row total...etc during data analysis preprocessing as I observed some results changed. The whole scope of this chapter says data extraction, data preprocessing and augmentation.

To achieve and maintain the balance of the dataset I used to merge additional data for analysis using data merging or joining techniques. For instance, based on common relationship and common key such as Payment terms, buyer address, and buyer name I used to join to create new data and allows me to combine them into single data to each invoice. I separately saved the added relevant data to text file, and I used to retrieve them for the integration process for better training the model. The additional information was taken from a legit official site:

- **Buyer Address:**
  - site 1: `http://www.comuni-italiani.it/##/indirizzi.html`
  - site 2: `https://www.paginebianche.it/`
- **Buyer Name:**
  - site 1: `https://lab24.ilsole24ore.com/leader-della-crescita-2020/`

- **Currency:**
  - site 1: `https://gist.github.com/ksafranski/2973986`

I assessed the data quality completeness, accuracy, representative and consistency by analysing the impact of the missing values and errors that could occur using plot. To ensure model training data is not overfitting I used hold-out validation splitting the data to train and evaluate their performance in test set. I validated the preprocessing steps that could indicate errors using statistical and visual inspection and domain knowledge in identifying unusual pattern and getting feedback from supervisor to review them. It helped me reduce noises unnecessary information and make it easier to analyse and more reliable and helped me reduce training time and to improve model accuracy.

I preferred to work on with DocILE invoice dataset for my research objective working with subset of dataset as they are large and are computationally efficient. Regarding data quality some where with noise and so I was able to manually clean and used sampling techniques. Criteria I encountered based on time period of the dataset region or companies all were variety including USA and Europe and currency globally. I took the selected subset with high occurrence as a label as a representative to ensure for all balanced characters. The selected dataset size was in according to 15 different providers with each 10 files later augmented additional around 700 original relevant data to make sure the completeness and balance.

As the research is based on real world, but not to specific industry it was not based specific geography rather it was from different regions among them USA and Italy. However, in some aspect there was some downside that shows variation and influence the analysis of processing the data. Among them was of their type missing with some data. For example, invoices in USA region miss having entities such as VAT number buyer order while in Italy could have most information including the buyer name, buyer address, Partita. IVA (VAT number) ...etc. Another role regions could play was interaction business USA could interact with business such as customer and suppliers or providers others could be with retailer or distributors, these variations impact in invoice processing so to minimize regional variations this model includes all kinds of variations or is general developed spacy model. To mitigate I was not using train data for testing.

Impact on results, the result is generalizable to entire dataset, but there were some drawbacks as I was focused only on invoices from previous years and different industries. The selection process was fair and equity and representative.

The data access method is it was publicly available in repository created by Czech company Rossum in GitHub. However, I used a secret token to get authorized access. Can be available for download in different format. I downloaded in PDF and all those data was download directly to my local machine. the data was static not changing or modifying. I used Git Bash command tool to clone the data to private storage for offline access and get complete copy free access in case removed from internet. Cloning mechanism inside Git Bash I used was:

```
c:\Users\Administrator\desktop>git clone
    https://github.com/rossumai/docile.git
```

The format of the data was PDF. After cloning I checked the data integrity of uniqueness and noise existence later, I ensured it is complete

I stored the cloned data locally in: `C:\Users\Admin\Workspace\docile`

3.2 Extraction of documents using the provided token

In this section it tries to explain the detailed process of extracting the documents from dataset with the provided token from authorized sources.

Given GitHub secrete token was access key to authenticate users or applications to a service use on behalf of it. And prevents security granular access control auditing.

The token was provided by data source as part of research agreement from supervisor so that in most cases can be accessed without password and manage of any size.

I used to download the document itself only when downloading I did not extract any relevant metadata such as invoice number order number ...etc. all entities metadata name was as

- **Invoice number:** unique key identifying the invoice data.
- **invoice date:** when invoice was issued
- **Order number:** to which the invoice is for

- **Currency:** monetary unit which entire fees is expressed.
- **Amount:** general encountered fee and price of the product
- **Buyer name:** institution or individual who purchases an item.
- **Buyer address:** where their business offices or service is located..
- **Buyer P.IVA (Partita. IVA):** tax number
- **Supplier name:** name of company who provide sales.
- **Payment terms** under which payment must be done.
- **Expiration date:** cut-off date invoice must be paid.
- **Description:** bill service being charged or product details
- **Quantity:** amount of service or product
- **Cost:** price per item

Invoice entity labels extracted to enhance understanding and use to analyse or identify top or low supplier that money has spent with and identify expensive invoice so that where you can use a reduced costs so that identifying the among insight to improve financial efficiency. Other is to notice and track invoice type where it spent most. Also used to search enhance ability for unique label or invoice entities such as order number, invoice date ...etc which save you time.

The quality of the measurement to extracted data was with NER model which has direct impact on it as it is trained on large and diverse dataset of invoice documents.

I used the extracted data in filtering by desired entity labels so able you identify group of invoices that I could have interest or in need analyse. To categorize using their entity names which helps you know how spent money on it. We can see better overview of types of metadata extracted their significance.

Validation data ensure accuracy and reliable. It helps increase efficiency easier to find data you need. I checked completeness of the invoice data by verifying header information such as invoice data buyer name and buyer address. I scanned each to verify if all remained relevant data are existed such as Quantities description, order number. In case i find incomplete or inconsistency I used to remove them do augmentation processes. I sometimes use a verification to consistency of data by comparing the original and extracted information. For

instance, I matched the original file of Buyer name with extracted JSON file. If some entity label is missing it could be a sign of incompleteness. If there are two documents that has similar data value again its a sign of duplicate invoice. If minimize such inconsistency among those data I used to correct them by double checking the original documents else, I used to remove the data that were unable to resolve.

3.3 Gathering documents from different providers

The obtained source of data was from GitHub. I selected the providers according to high availability rank for sampling that offer completeness and consistency. That also can offer various sales, regions, and time-period. This can help me represent real world data.

I considered some alternative data source for augmenting the imbalanced entities to meet some of my needs. Alternative site I used for adding was in the above section 3.1 of different sources. Each supplier provides diverse types of invoice documents such as sales invoice, purchase invoice or credit notes. Regardless their type I selected them based on regions and companies' retails.

I used multiple providers among them mentioning their name is.

Name of the provider:

- Alpha Media
- Entercom Communication Corp
- Iheart Media
- Katz Television Group
- KPNX
- KRNV
- KRXI FOX11
- KSAZ
- REMIT TO Sinclair Broadcast
- Salem
- Townsquare Media Tuscaloosa

- Salem
- WMUR
- WPXI-TV
- WSB-TV
- Mixed(alternative) Italian source

They were collected from the repository GitHub site.They are with multiple type of invoices with satisfied and wide range of real-world representation having most required relevant data. These relevant data aids me to contribute valuable to my research. As they all have desired values and correlation between each invoice. However, I have not met directly to any of those providers rather I used their licenced data to obtain access.

I considered some alternative sources such as buyer address[13][14], buyer name[15] and for currency label[16] of providers relevant data to fulfil the imbalanced data that occurred in the invoices from DocILE

All data is found in one platform DocILE official GitHub site, the data collection process for each supplier was done manually. Data was locally collected with the following steps using Git Bash:

- **step1** `c:\Users\Administrator\cd Desktop`
- **step2** `c:\Users\Administrator\desktop>git clone https://github.com/rossumai/docile.git`
- **step3** `c:\Users\Administrator\cd workspace`
- **step4** `cd docile`
- **step5** `c:\Users\Administrator\Desktop\docile`

After downloaded it was saved to specified folder and saved the labelled data in drive based:

- https://drive.google.com/drive/folders/1bqgMzP9VnvouFkq4Iv6DHrrCYrplurBT

Volume of the data varies in each provider, but with equal number of documents. Some providers in each invoice data could have different number of pages, for example in KPNX provider among the 10 acquired documents

of it some could have only 3 pages other 10 or 20 pages. The format and structure of the invoice documents could be varied in information placement each provider has diverse payment terms. Transformed data had taken place from other providers to create a unified dataset. For example, some relevant data separately save in text file in order to be augmented the less occurred entity of any providers.

Data quality checked across providers in terms of inconsistent data format, values ... etc. to standardize them process of removing duplicate filling missed values using augmentation, normalizing the data by scaling it to same threshold.

Collected meta data from each was of all those entity information to enrich the consistency of the dataset.

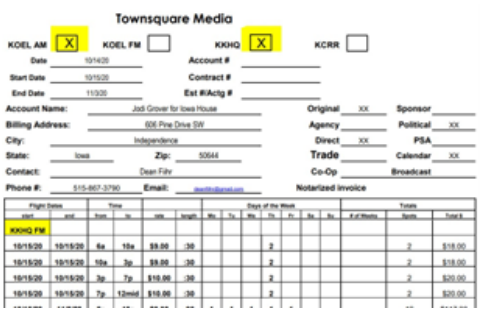
3.4 Preprocessing of raw documents for further analysis

Preprocessing strives to eliminate any undesirable characteristic or non-essential components such as interference or diminishes incongruous information and interference on the invoice data. Document processing is the daily work process in the business operations and other area which is responsible for the preparation, scanning, verification's, and communication.

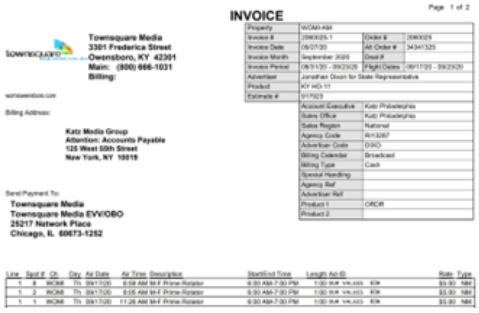
The raw invoice was in a PDF format with different elements in it having all these variety of text, image and sometimes tables which were challenges to extract information in compared to text documents only. To resolve this, I used a an open-source tesseract OCR approach to parse text information from PDF invoices and used a natural language processing tool called Prodigy to annotate the extracted information to identify and analyse the desired invoice labels which could help save time and improve accuracy as NLP model can be trained on annotated data to recognize entities with better accuracy as model depend on quality of training data.

Some PDF formats were in scanned images which makes t difficult to extract text from, I used OCR to do so, but it was not always accurate, so I used to intervene and review manually to make sure it is correct. Other PDFs were used to be in different size and fonts and line spaces this also were challenges to make them consistent and affected the preprocessing effort to handle a data cleaning pipeline such as regular expressions techniques and I did manual review. Tesseract OCR.

Structure of invoice was not same some were consistent layout, and some were different structure from one provider. For example, we can see down below two separate layout and structures from "*Townsquare Media*" supplier.



(a) highlights header and table layout



(b) highlights header and table layout

Figure 3.5: different layout for one entity label

As we see in Figure 3.5 we have a better layout which typically has Header at top of invoice page, Row with Tables at bottom of the page.

The Header includes the entity information such as Invoice name, Invoice Date, Order Number, Currency, Amount, Buyer Name, Buyer Address, Supplier Name, Payment terms, Expiration Date. In Row table we are looking for another entity information which are Description, Quantity, Cost or Price. Invoice that had different structure other than the above we have in fig.a, and fig.b, may not have proper Row or header section or could have additional section for Taxes of any or variety scenarios like shipping tax. Variation in my dataset impacts the preprocessing, so to handle I used regular expression and prodigy tools and after annotating it was important to spend more time to manual review to ensure accuracy and completeness.

The data field of those entity information was important to in different way. For example, knowing the Buyer name and Supplier name details helps recognize and study their relationship interval.

Some issues were existed that cause incompleteness to the raw invoice document. Payment terms was not informed or did not have due date, some did not include well their cost value...so I imputed missing data by considering and estimating other values, for example if cost value of item is missed, I use from same supplier by taking average cost value. If any of the invoice had no information or with many redundant at all I discard them away. So above the document format, structure, missing or incomplete and data quality were main issues or characteristics I considered to resolve the raw datasets and resolved

them with different techniques such as duplicate removal...etc.

Duplicates occurs due to multiple reason among them is system error. To identify and remove duplicates I used to normalize the data and detecting the duplicates using exact match manual searching for information in VS Visual software. Removing duplicate improve accurate insights. Some data were frequently absent due to many reasons for example the Payment terms entity name was not written or informed this could be due to agreed to be *"Cash on Delivery"*. Another entity was cost which was with many missed values and instead of excluding I did media impute to reproduce the process. Outlier detection methods I used a statistical method using plot visualization and domain knowledge when the occurrence was too high or low. I removed the redundant if they were out of desired threshold.

I did data validation check for example if data are invalid such as if any entity are miswritten or with short digit number, I fixed into correct format so that they must be unique, valid digit, supplier name must be same with local saved database documents as well to ensure its consistency. Techniques for cleaning the data I used was number of scripts to automate cleaning specific tasks within JSON file format among them was quality metrics such as outlier count, duplicate record count, missing value count...etc. And Data visualization such as bar chart to identify anomalies or patterns in the data.

Text extraction techniques was done using tesseract OCR that converts into machine readable text from PDF or scanned images as it accurate and efficient. Support many languages. Every preprocessing such as noise reduction and resolution adjustment were used to prepare the invoice document or create the file for prodigy steps. Semi automated approach was used to extract to text which means tesseract OCR and manual reviewing to ensure accuracy. Pytesseract python library and regular expression techniques to identify misread characters were used to perform OCR on scanned images and were able to extract all the required entity label.

I saved the extracted in JSON file format as its easy to parse. The extracted JSON structure looks like this:

```
{
  "text": "I Page 1 of 3 VO CC Property KDGL-FM Alpha Media Invoice #
     330926-1 Order # 330926.......",
  "spans": [
    {
      "start": 37,
      "end": 48,
      "token_start": 9,
      "token_end": 10,
      "label": "Supplier Name"
    },
    {
      "start": 59,
      "end": 72,
      "token_start": 14,
      "token_end": 15,
      "label": "Invoice Number"
    }
    // More spans here...
  ]
}
```

Listato 3.1: Sample JSON Structure

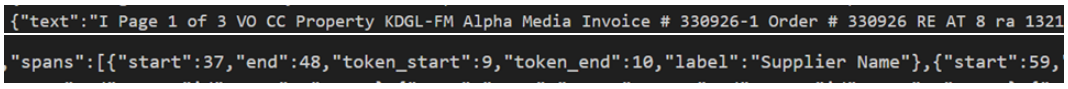


Figure 3.6: Sample JSON Structure in practice

The JSON structure holds all information of invoice in it such supplier name, invoice number ...etc. Validated and reviewed using custom script for misread characters, missing text, or improper format.

Standardization improves data analyses if all invoice formats are same, same field names and same layout. Regardless their original layout all invoices were converted to text file with the help of tesseract OCR. Then it was annotated with prodigy tool such process involve recognizing and labeling various entity information such as supplier name, order number...invoice date ...etc. These annotated JSON files were cleaned using custom python script, including removal of duplicates, and made sure labeled correctly. And it was converted to spaCy compatible format so that easily can be used by spaCy NER model. There was

variation in structure for example, some invoices used the term "vendor" while others used "supplier". There also placement of labels was not same structure some at the top and others at the bottom.

Despite their structure variation the documents were standardized into consistent format with help of prodigy tool and custom script. Due to this the spaCy NER model able to train on standardized dataset regardless their layout and structure. This is so that the NER model even if entities are presented differently, it can learn their relations between them.

Tools used for standardization were tesseract OCR, prodigy, custom script, and regular expression.

- Tesseract OCR used to convert invoice to text.
- Prodigy to annotated data in JSON file.
- Custom python to clean annotated invoice data and convert to spaCy compatible format.
- Regular expression to recognize pattern in invoice document.

Example:

```
from fuzzywuzzy import fuzz
import os

def remove_similar_duplicates(file_path, threshold=80):
    with open(file_path, 'r', encoding='utf-8') as file:
        lines = file.readlines()

    unique_lines = []
    for line in lines:
        is_duplicate = False
        for unique_line in unique_lines:
            similarity_ratio = fuzz.token_set_ratio(line, unique_line)
            if similarity_ratio >= threshold:
                is_duplicate = True
                break
        if not is_duplicate:
            unique_lines.append(line)
            .
            .
            .
```

Importance of metadata extracting can be varied in spending patterns, fraud detection or risk assessment. Such as identifying areas where prices can be decreases. Knowing payment terms assess risk to do business with various suppliers.

Data were extracted with a different technique using regular expression, natural language processing. Extracted data has impact in analysis to recognize patterns in the data. Predict future outcomes for example to predict how much money can be spent on invoices in next year by the company.

Handling language and characters encoding was done using Unicode encoding scheme UTF-8 with a python package *encoding*.

Pre-processed data were stored in a local machine and the cleaned dataset saved within prodigy were used by NER model.

Chapter 4

Custom Named Entity Recognition (NER) Model

4.1 Introduction to NER and its importance in document processing

Named entity recognition is a process where a sentence or a chunk of text is parsed through to find entities that can be put under categories like names, organizations, locations, quantities, monetary values, percentages, etc within a text. NER used in a variety of application, among them is in information retrieval to identify and extract named entities from documents.

And it plays important role in document processing especially when it comes to extracting information from unstructured texts like invoices. Because invoices data are not organized, or they are unstructured which makes it difficult to extract in traditional way. It transforms into structured format by recognizing and categorizing named entities. From invoice document it identifies information such as invoice date, invoice number and supplier name...etc so that the identified and classified data can be stored in structure database. By doing this it improves its efficiency and productivity.

It has wide range of service in an invoice document processing. Extracting metadata from invoice could be invoice date, invoice number, order number, currency, buyer name, buyer address, supplier name, payment terms, expiration date...etc then this can be used for many purposes such as fraud detection or in any business area. It indexes the invoice so that easier for retrieving specific information. It helps improve efficiency in many different ways.

- Invoice date extract can be utilized for tracking payment due.

- Supplier name extraction helps track expenses done.

Due to some character like invoice having different format, structure or layout and language they could be a challenge for named entity recognition. Among those difficulty is handling OCR errors. For example. scanned invoices are converted to text using optical character recognition. But they still create some noises into the converted data making it challenge to NER model to identify named invoice entities.

Invoice could also include logos and images as well as table which could bring additional difficulties to NER model. Getting different invoice template also cause the difficulties. Invoice may also contain names of organization people products which are beyond the relevant data.Regardless these many deficiencies named entity recognition been improved to handle OCR errors, noisy data, various template. Large dataset of labelled invoice plays a big role to train it so that it would become accurate.

Some benefits of custom ner model are that they trained on domain specific task like getting information from invoices which allow them to learn type if entities and languages. Custom NER model is trained to recognize specific data that can not be detected by pretrained NER model. Custom NER model could be able to recognize the fraudulent invoice, for instance the model could be trained to recognize invoices with unusual invoice number.It can be integrated into automated document processing workflow in multiple ways such as approach we could is to use it as pre-processing step. It extracts key metadata or information from invoices before the data is used to generate a report.

Its valuable tool for many scope of areas among them are healthcare, finance, and publishing media, retail education...etc to automate process of organizing and extracting information from text data.

Custom NER model can be fine-tuned to recognize metadata or invoice information such as invoice number or date ...etc to improve the performance and accuracy because trained on labeled invoice data which help them learn difference of languages. Because invoice can vary in structure and formatting, they can be adapted to unique invoice template.

4.2 Definition of relevant entities for information extraction

Definition of Invoice Entities: Invoice has two parts which are known as Header and the Row

Header

- **Invoice Number :** A unique key identifying the invoice. Each invoice that a provider issues is given a specific identification quantity, or bill wide variety. Invoices can be tracked and referred to inside the future thanks to this, which benefits both shoppers and suppliers. They can be alphanumeric and are usually sequential.
- **Invoice Date:** The date on which the invoice was issued. The day the invoice was created. The date on which a provider troubles an bill to a consumer is referred to as the invoice date. Specifies an appropriate date the bill was made.
- **Order Number:** The number of the order that the invoice is for. The order wide variety for which the bill is intended. An order's order quantity is the unique identity code given to it by the buyer. When more than one buy orders are made to the same supplier, it can be useful to link invoices to unique buy orders on this way.
- **Currency:** The currency in which the invoice is denominated as Euro, US dollar. The Euro or US greenback used to pay the bill. The monetary unit in which the entire fee is expressed is indicated by way of foreign money. Specifies the intended charge forex for the consumer.
- **Amount:** The total amount of the invoice. The invoice's normal dollar fee. Amount is a representation of the invoice's general fee, which generally consists of all applicable taxes, surcharges, and the price of the products or offerings. This presentations the general sum that the purchaser must pay the dealer
- **Buyer Name:** The term "Purchaser Name" refers to the enterprise call or call-of-enterprise used by the person or institution making the acquisition of the goods or offerings listed on the invoice.
- **Buyer Address:** The address of the buyer with services or where their business offices are located and applies for both individuals and companies. This time period refers to each personal individuals and businesses and refers back to the cope with at which the buyer offers offerings or the place in their commercial enterprise places of work. The Buyer's Address serves as a placeholder for the Buyer's bodily or mailing deal with and indicates wherein to send invoices.
- **Buyer P.IVA:** The buyer's tax identification number. The customer's tax identification wide variety (IVA). Client P. Italian for "VAT identity

quantity" is IVA (Partita IVA). For the purposes of taxation, Italian companies are given this special identification wide variety. It may be used to pinpoint the purchaser and set up the relevant tax regulations.

- **Supplier Name:** The name of the person or company who is selling the goods or services. The name of the man or woman or business promoting the goods or offerings is called the supplier. The time period "Supplier Name" refers to the whole call or commercial enterprise name of the individual or enterprise presenting the goods or services listed on the invoice. Determine the business that is due the payment.

- **payment terms:** The terms under which the invoice must be paid. The situations beneath which the bill needs to be paid. The agreed-upon phrases for paying an invoice are mentioned within the terms of fee. It usually states the due date for payment as well as any discounts or fines related to early or overdue payments. Common phrases encompass "30 net" (fee due within 30 days) and "2/10 net 30" (2% cut price if payment is due inside 10 days, in any other case inside 30 days).

- **Expiration date:** The date by which the invoice must be paid. The cut-off date for making payment at the invoice. The bill's validity length and its fee terms are exact by using the expiration date. This indicates the very last day that payments can be made without being assessed fines or surcharges.

Row

- **Description:** The name of the bill or a brief precise of the best or service being charged. Here, in the description, are details about the products or services indexed within the particular bill row. In it, the rendered items and offerings are briefly defined or indexed.

- **Quantity:** The quantity of gadgets of the good or carrier being charged. The quantity in a row of an invoice denotes the entire amount of products or services that have been rendered, expressed because the quantity of items or units. It helps in figuring out how big or great the charges are.

- **Cost:** The price according to unit of the good or service being charged. The fee consistent with object or the charge consistent with unit for every object is distinct in this phase of the bill's row. It is increased through the sum to reach on the row total.

Recognizing invoice entities for invoice categorization, automating data entry is fundamental. As invoice categorization NER be used to categorize

invoices to better understand spending pattern.

Due to variable formats and OCR errors identifying metadata can be challenges. Variations in terms of layout, formatting and language makes difficult to develop NER model to all types of invoices. To face the issue, I used pre-trained language model ER model so that this fine-tuned model can learn language difference to more reliable to ICR errors. Using a domain specific NER model were trained on labeled invoice dataset-maintained from DocILE.

Preparing labeled training data for custom NER model followed this step gathering DocILE dataset invoices, data cleaning and preprocessing, labeling data and split the data for training and testing set.

Annotated guidelines were considered to be labeled all the named entity, but at least must be completed and it was misspelled or contain OCR errors and annotate even if different language. Instructions during annotation were taken regarding their format. For example, supplier name is a vendor name typically full of company name, some taxes were governmental-imposed taxes...etc.

Hierarchy of the invoice numbers were linked to specific invoice as it represents unique number. It relationship can also be associated with many different entities for example, invoices with customer, projects, and purchase orders.

Domain specific entities is with more variable for instance, numbers could be formatted in variety ways for this case generic model can not handle this variability. But customized NER model be able to handle this different issue occurs could lead to improved accuracy and efficiency.

Scalability fine-tuned NER model learned on different layout be able to identify entities with different layout and structure. Recognizing metadata of invoice is essential for information extraction and document processing in context of invoices.

4.3 Building a labeled dataset for training the NER model

Training and Fine-Tuning

With python source files: *hyperparameter.py*, *model_ config.cfg* , *NER-Dataset.py*, *NERTraining.py*, *run.py*, *service.py*, and *service_ config.py*

When interpreting the script, it used to give me of flexible results and I put them as first result and second result down below as error analysis, to improve

the results I first tried removing the duplicates already available in some labels which were still kept since they were exported after prodigy tool annotation process completion. The validation set is used to compute these metrics as it is used during the training and the final metrics are related to the test set.

```
1st
============================== Training pipeline ==================
 Pipeline: ['tok2vec', 'ner']
 Initial learn rate: 0.001
```

E	#	LOSS TOK2VEC	LOSS NER	ENTS_F	ENTS_P	ENTS_R	SCORE
0	0	0.00	53.92	0.00	0.00	0.00	0.00
0	759	0.00	3348.41	66.67	80.39	56.95	0.67
1	1518	0.00	2238.42	69.75	77.51	63.41	0.70
1	2277	0.00	1638.75	80.51	88.89	73.58	0.81
2	3036	0.00	1003.81	79.96	87.82	73.39	0.80
2	3795	0.00	1190.46	83.37	88.91	78.47	0.83
3	4554	0.00	1308.56	83.86	88.31	79.84	0.84
3	5313	0.00	869.80	85.45	88.98	82.19	0.85
4	6072	0.00	1661.49	85.02	88.72	81.60	0.85
4	6831	0.00	849.26	86.07	91.05	81.60	0.86
5	7590	0.00	616.07	85.33	89.66	81.41	0.85
5	8349	0.00	920.67	86.89	91.92	82.39	0.87
6	9108	0.00	648.32	87.23	91.24	83.56	0.87
7	9867	0.00	673.15	88.05	92.09	84.34	0.88
7	10626	0.00	924.89	87.93	92.08	84.15	0.88
8	11385	0.00	971.34	88.10	91.74	84.74	0.88
8	12144	0.00	554.28	88.24	91.58	85.13	0.88
9	12903	0.00	736.04	88.17	91.21	85.32	0.88
9	13662	0.00	504.39	88.60	91.46	85.91	0.89
10	14421	0.00	406.27	88.31	91.06	85.71	0.88
10	15180	0.00	494.22	88.06	91.19	85.13	0.88
11	15939	0.00	555.64	88.06	91.19	85.13	0.88
12	16698	0.00	466.47	88.62	92.18	85.32	0.89
12	17457	0.00	517.04	89.21	92.99	85.71	0.89
13	18216	0.00	641.52	89.23	92.81	85.91	0.89
14	18975	0.00	663.98	89.02	92.60	85.71	0.89
15	19734	0.00	494.33	89.25	92.63	86.11	0.89
16	20493	0.00	570.50	89.16	92.44	86.11	0.89
17	21252	0.00	512.66	88.69	91.65	85.91	0.89

```
 18   22011          0.00    600.64   89.00   91.88   86.30    0.89
 19   22770          0.00    571.59   89.05   91.53   86.69    0.89
 20   23529          0.00    487.56   89.11   91.89   86.50    0.89
 21   24288          0.00    571.23   88.96   91.34   86.69    0.89
 Saved pipeline to output directory
output/model/model-last

2023-08-03 12:03:42.622365: I tensorflow/core/platform/cpu_feature_guard.cc:182]
This TensorFlow binary is optimized to use available CPU instructions in
performance-critical operations.To enable the following instructions: AVX2
FMA, in other operations, rebuild TensorFlow with the appropriate compiler
flags.2023-08-03 12:03:44.233395: W tensorflow/compiler/tf2tensorrt/utils/py_util
 Using CPU

================================== Results ==========================

TOK     100.00
NER P   92.63
NER R   86.11
NER F   89.25
SPEED   10982
============================== NER (per type) ====================

                       P       R       F
Quantity          100.00   95.45   97.67
Currency           98.92   98.92   98.92
Buyer P.IVA       100.00   90.32   94.92
Cost              100.00   94.87   97.37
Order Number       95.56   89.58   92.47
Payment Terms      92.50   97.37   94.87
Supplier Name      87.50   73.68   80.00
Invoice Number     87.50   80.77   84.00
Invoice Date      100.00   91.30   95.45
Expiration date    85.00   85.00   85.00
Buyer Name         80.00   26.67   40.00
Buyer Address      90.74   85.96   88.29
Description        18.18   12.50   14.81
Amount             55.56   55.56   55.56
payment terms      77.78   63.64   70.00
 Saved results to output/model/result.json
#!zip -r model_best.zip /content/output/model/model-best
```

2nd

```
============================== Training pipeline ==================
 Pipeline: ['tok2vec', 'ner']
 Initial learn rate: 0.001
```

E	#	LOSS TOK2VEC	LOSS NER	ENTS_F	ENTS_P	ENTS_R	SCORE
0	0	0.00	0.00	0.00	0.00	0.00	0.00
0	1081	0.00	7139.56	57.67	77.51	45.91	0.58
1	2162	0.00	5021.62	64.53	80.77	53.73	0.65
1	3243	0.00	3228.64	75.35	84.70	67.85	0.75
2	4324	0.00	2236.97	76.54	87.63	67.94	0.77
2	5405	0.00	2411.69	78.65	87.40	71.49	0.79
3	6486	0.00	2071.23	79.16	85.71	73.53	0.79
3	7567	0.00	1947.25	80.16	89.13	72.82	0.80
4	8648	0.00	2322.08	80.17	87.25	74.16	0.80
4	9729	0.00	1725.49	80.69	87.13	75.13	0.81
5	10810	0.00	1469.65	81.53	87.32	76.47	0.82
6	11891	0.00	2130.44	81.11	84.72	77.80	0.81
6	12972	0.00	1471.18	80.55	83.93	77.44	0.81
7	14053	0.00	1569.57	81.30	84.21	78.60	0.81
7	15134	0.00	1826.35	82.46	86.61	78.69	0.82
8	16215	0.00	2050.69	83.45	87.29	79.93	0.83
9	17296	0.00	1770.79	84.10	87.67	80.82	0.84
10	18377	0.00	1685.70	83.98	87.19	80.99	0.84
11	19458	0.00	1771.21	83.64	85.87	81.53	0.84
12	20539	0.00	1604.77	83.12	84.97	81.35	0.83
13	21620	0.00	1720.25	83.36	85.38	81.44	0.83
14	22701	0.00	1563.79	83.26	85.57	81.08	0.83
15	23782	0.00	1473.60	83.73	85.75	81.79	0.84

```
 Saved pipeline to output directory

================================== Results ========================

TOK     100.00
NER P   87.67
NER R   80.82
NER F   84.10
SPEED   10549

============================== NER (per type) ====================
```

	P	R	F
Supplier Name	73.68	74.67	74.17
Invoice Number	89.02	86.90	87.95
Invoice Date	98.68	94.94	96.77
Expiration date	92.11	90.91	91.50
Quantity	100.00	95.24	97.56
Order Number	93.98	88.64	91.23
Currency	98.21	98.21	98.21
Buyer Name	86.90	81.11	83.91
Amount	89.43	59.46	71.43
Description	35.82	38.10	36.92
Buyer Address	85.53	76.47	80.75
Cost	100.00	94.87	97.37
Payment Terms	96.15	100.00	98.04
Buyer P.IVA	100.00	92.59	96.15
payment terms	82.86	85.29	84.06

Thereafter Completion processing of labeling Data using prodigy tool, I used this command to get the exported file, which is called *"exported_ data.jsonl"*.

<python -m prodigy db-out batch_0 > exported_data.jsonl >

I used the exported .Jsonl file as an input to return each available entity name or labels gathered or their occurrence in number in dictionary data type as we see down below with their visual graph in numerical distribution.

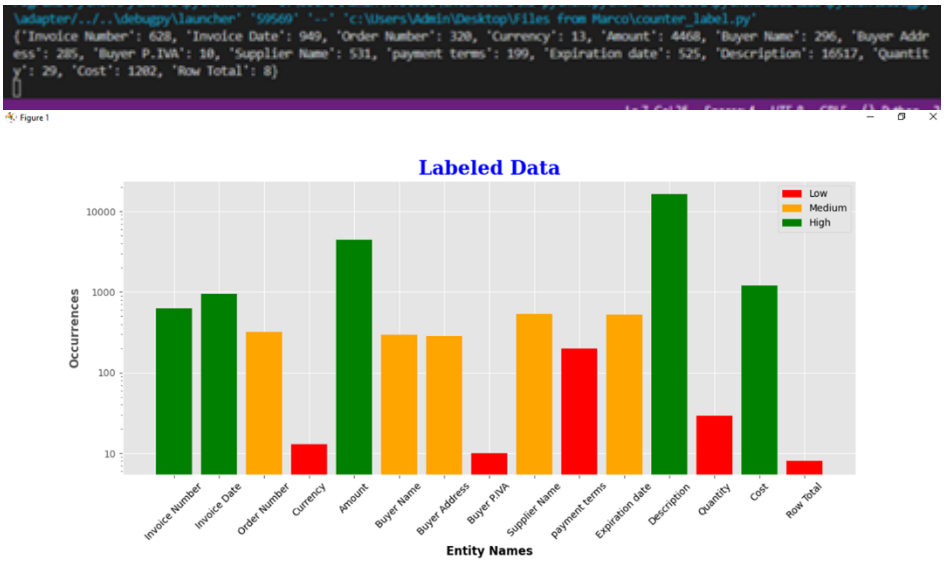


Figure 4.1: Entity occurrence and Visual Distribution

After finding the total number of annotated labels, then I saw there were high number of differences between those entities. This was caused due many reasons as some part I tried to point out the struggles when doing annotation process.

- One is the less numbered label was not present in the original PDF invoice.
- Second, they were not extracted properly using the tesseract OCR framework.

For this reason, we decided they need to have the same threshold number of entities for all and at beginning I made to have a 1000 threshold to each label to be a balanced dataset, later I used to use a threshold of 500 each. For those who are high in number to be reduced (cut-off) and those who are low to be increased. To reduce such noise data augmentation needed to be taken.

Data augmentation:

Is a technique of artificially increasing the training set by creating modified copies of a dataset using existing data. In our dataset we only used an original data, but not synthesized. It includes making minor changes to the dataset or using deep learning to generate new data points. Techniques used for audio, Text, Image, and advanced data augmentation techniques. And We use it:

- To prevent models from over-fitting.

- The initial training set is too small.
- To improve the model accuracy.
- To Reduce the operational cost of labeling and cleaning the raw dataset.

Text Data Augmentation:

- Word or sentence shuffling: randomly changing the position of a word or sentence.
- Word replacement: replace words with synonyms.
- Syntax-tree manipulation: paraphrase the sentence using the same word.
- Random word insertion: inserts words at random.
- Random word deletion: deletes words at random.

Task

I used vs code app for both to compile code and to demonstrate by launching how the JSON file structure looks like such as the number of lines in existence with the spans list with or without the labels. . . etc. Occasionally, I find similar entity names in other invoice pdf, for example, payment terms, quantity, buyer address, buyer name. But there is a different order number, or invoice numbers as this was needed to be expected beforehand.

In order to achieve the augmented data

To achieve to have the same threshold or level of balance I used to add data to scale up those less in occurrence and do cut-off to those high in occurrence regardless those labels were repetitive values or not and varied providers in European (Italian), or mostly in USA docs later. The additional data were added from docile repository and the plot visualization used to look like this after cleaning and preprocessing the data:

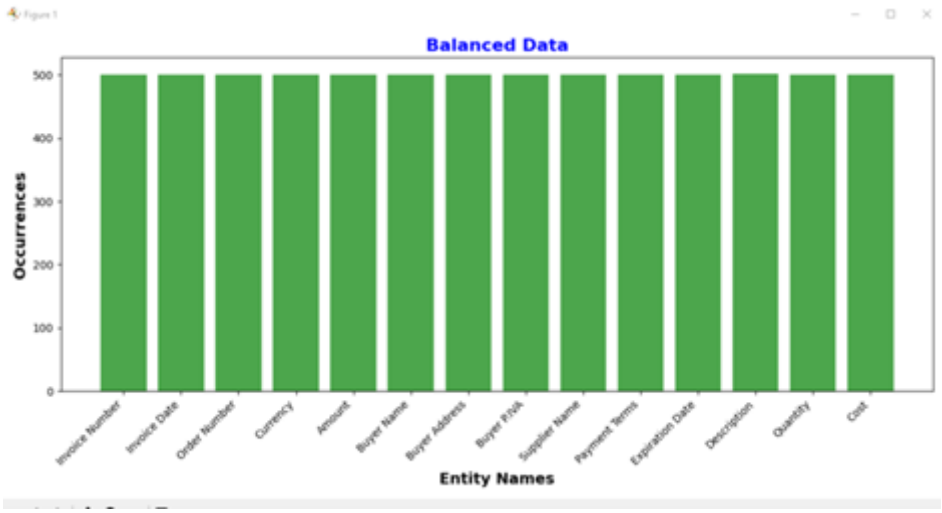


Figure 4.2: Balanced data

later I decided to make sure the metadata or the entity label should have a unique with no repetitive relevant data regardless level of balance, but they are almost all the labels have augmented data due to careful addition of unique data to each label.

The labeled data counted numbers were not same. Some were below the threshold, and some were above the threshold. I added to below threshold. This screenshot was before balance.

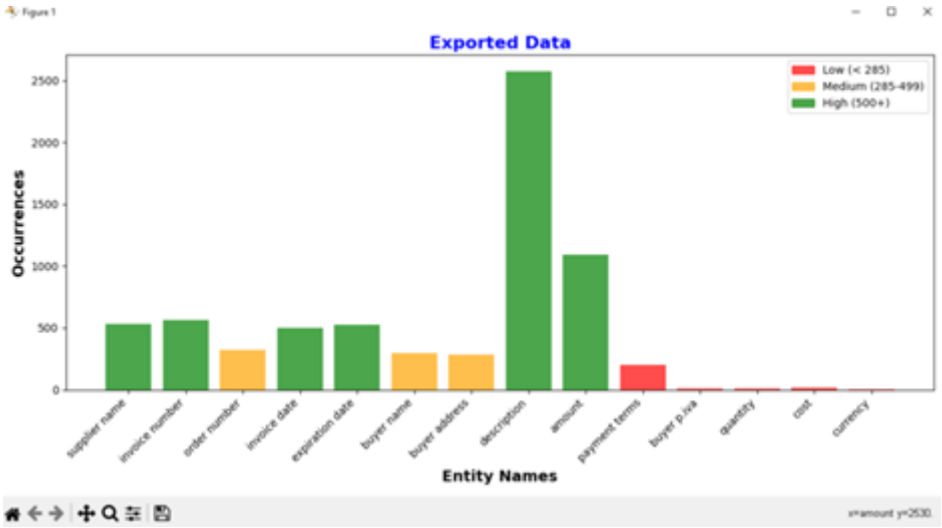


Figure 4.3: unbalanced entity labels

These are the entity name with new added relevant data values. Each these entity name aims to have the same threshold.

This down below file was after the threshold was measured and considered. To achieve the balance of the data regardless the existence of duplicated data kept from prodigy, as of now till this step I concentrated to assume an original validated data must be collected.

For this reason, to meet the required threshold the new data collected was taken from a specific website such as:

- **Buyer Address:**
  - site 1: `http://www.comuni-italiani.it/##/indirizzi.html`
  - site 2: `https://www.paginebianche.it/`
- **Buyer Name:**
  - site 1: `https://lab24.ilsole24ore.com/leader-della-crescita-2020/`
- **Currency:**
  - site 1: `https://gist.github.com/ksafranski/2973986`

So that the occurrence of the label and their plot visualization looked like this: Number of occurrences of each entity labels:

- payment terms: 363,
- supplier name: 499,
- invoice number:499,
- order number:499,
- invoice date:499,
- expiration date:499,
- buyer name: 499,
- buyer address:499,
- amount:500,
- buyer p.iva:500,

- currency:491,
- description:502,
- quantity:255,
- cost:219.

Bar Chart Plot Data Visualization of entity labels: Diversity in formats, I

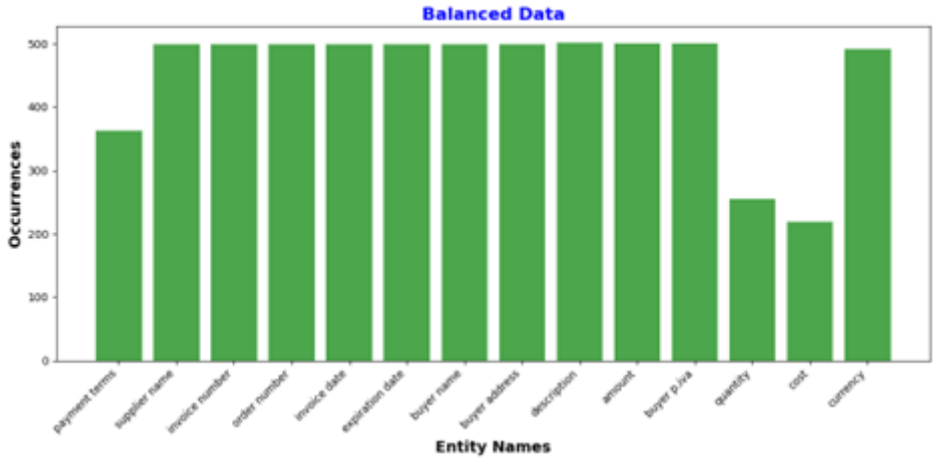


Figure 4.4: Entity labels with augmented data from different source

selected the documents from DocILE for labeling and considered the following.Diversity in formats of templates, languages and diversity in data sources invoice should include different industries and different countries. Diversity in layouts...etc.

Prodigy tool is popular for NER annotation with many features, labeling entity. To use it I needed to create recipe as they define various entities and learn according to your labeling entities:

```
C:\Users\Admin\Desktop\Files from Marco>python -m prodigy ner.manual
    labeled_data spacy/spacy batch_0.jsonl --label labeled_text.txt
```

When labeling the information, the instruction I used for each entity were generally looked like this:

- **invoice #:** 2044697-1

- **Invoice date:** 04/0621
- **Order #:** 544259C

Achieved pre-processed steps such as text extraction using tesseract OCR tool and formatting the standardization invoice and normalization.

4.4 Implementation of the NER model using SpaCy framework

Training the spaCy model with already the OCR system that I used for labeling the prodigy. The text in the JSON file I have is already extracted using the tesseract. I fine tuned the pretrained spaCy model which used aids me to recognize some inputs using my collected dataset in my own environment.

I used spaCy framework for implementing my custom NER model for the following reason. Beyond widely used by communities it provides simple API for training and to use with NER model and implements many optimizations efficient for large-scale NLP task. And is suitable for my invoice data. also provide many pre-trained models for NER tasks aims form invoice data as it is a starting point for training a custom NER model for invoice data.

It supports nested NER labels spacy in recognizing complex labeled entities such as Description which may contain multiple names or dates. As sometimes invoice documents are multilingual spaCy gives pre-trained NER model for a variety of languages which can be used to train custom NER models.

I utilized SpaCy libraries as the core framework for NER model development. TensorFlow was integrated into spacy to enable machine learning operations. I also considered TensorRT as optimization framework for deep learning models was evaluated to increase model performance, especially on GPU architecture. Entire environment was configured to run on windows command prompt to ensure compatibility with spaCy and other libraries. Loss values and model performance metrics produce training statistics.

Model architecture of the NER model developed using SpaCy framework as a foundation as it is efficient NLP library proving primary framework for building NER model. I leveraged pre-trained word vectors used from pretrained word vectors often used embeddings like Word2Vec or Glove as they provide valuable and semantic information about text and words. The input text was tokenized into subword units ensuring model processes text at granular level.

Convolutional neural network was used to perform token classification as spaCy Ner model use it. NER model scans the tokenized input text and assigns entity labels to each token based on the context provided by neighbouring tokens.

The model is trained to identify labels such as invoice number, order number...etc. Annotated trained data being used for training. During training the model reduces a loss function which measures dissimilarity among predicted entity labels. The model employed optimization techniques such as optimizers like Adam to improve performance and update its parameters. After predicting names of entity to refine results such merging adjacent tokens with same entity type. A post-processing steps were applied. Model evaluation using NER metrics including precision, recall and F1-score were used to assesses the ability to correctly recognize entities in unseen text.

Created labeled data using prodigy annotation tool containing for Named Entity Recognition (NER). Data were converted to compatible spaCy format typically includes tokenized text and entity annotations. Each entity name along with its entity type is labeled with a start and end position. Data was divided into training and evaluation sets. 90% for training and 10% for testing were used. So, the evaluation and training with their respective annotation were integrated into spaCy. And were saved in spacy binary format optimized for training model. A spaCy training configuration file *config.cfg* was generated and it contains hyperparameters and settings for training the NER model like the architecture of the neural network, batch size and dropout. NER model was trained by integrated training dataset and configuration file. the model then can be loaded for NER tasks on new unlabeled text data.

Training process or steps followed were from data preparation till model exports. Created labeled dataset for NER, including annotations for different entity type like invoice date, order number and others. This was split to training and evaluation sets. A generated spaCy configuration file specifies the architecture and hyperparameters for the NER model. Initiated training process using spaCy module providing the path to the configuration file and training data.

Training step used this spacy framework predefined architecture for named entity recognition (NER) which involves neural network based such as transformer or convolutional neural network (CNN) based model.

It includes multiple epochs. In NER model training to tune hyperparameter like dropout rate, regularization strength and learning rate. Model performance was evaluated on the evaluation dataset and models result save in JSON file for reference.

Evaluation metrics such as precision, F1-score, recall, and accuracy were used to assess the model performance. Precision indicates how many of named entities correctly identified. Recall ensures and identifies all relevant named invoice entities in dataset. High recall indicates model minimizes false negative. F1-score mean of precision and recall balances their metrics when there is imbalance between negative and positive. Accuracy measures overall models' predictions and general model performance in NER model.

$$\textbf{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\textbf{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\textbf{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\textbf{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Fine tuning and optimization techniques were applied to improve performance using spaCy framework. Transfer learning to initiate model training I leveraged pretrained word vectors like *GloVe* or *Word2Vec* which allow model to benefit from rich semantic information. And hyperparameter tuning including dropout rate learning rate and batch size to find right balance that maximize performance and prevent overfitting.

During implementation and training of NER model using spaCy some errors were used to be encountered and error analysis has been done to handle it. Among them data annotation quality some difficulties encountered inconsistent annotations and inaccurate lead to poor results and reviewed and corrected using custom script and regular expression methods.

Imbalanced data to the required threshold has applied for some entity that used to be seen underrepresented in the training data and to address this I used to augment and adjust the overrepresented entities. Selecting right hyperparameter for the training also was bit challenging. Evaluation metrics selecting appropriate evaluation metrics is essential for assessing model performance to address this metrics like recall and precision were defined to measure entity type relevant to invoice processing. Error occurred when the model could not find evaluation dataset then creation of evaluation dataset and added it to training data fixed it so that the model would be able to learn from both training and

evaluation data. The model was not able to generalize new data, so increasing the size and diversity of training dataset and added data from different sources and domains this changes robustness of the model.

Metrics used to assess performance of NER model on test dataset were precision, recall and f1 score.

Precision was determined approximately 86.91% and **recall** of the model was around 85.86% this metrics illustrate the model can successfully capture; higher recall shows model can recognize portion of actual named entities. **F1-score** as we see computed 86.38%, a balance between precision and recall.

Result:

```
TOK     100.00
NER P   86.91
NER R   85.86
NER F   86.38
SPEED   11515
```

4.5 Evaluation and performance metrics of the NER model

I used the data evaluation on both validation set and test set:

Precision measures the accuracy of named entity model prediction and calculated at the number of true positive or correctly identify entities divide by total number of predicted entities. It helps model prediction, and it does not falsely recognize non- entity words as entities which crucial for application such as data analysis and information extraction.

Recall a.k.a a true positive rate, measures the models, ability to recognize all relevant named entities. And calculated as true positive prediction and divided by total number of actual entities. Avoids missed entities and, high recall ensures the model does not overlook any relevant entities in text.

F1-score is a measure combining both precision and recall. It is generally described as the harmonic mean of the two. Harmonic meaning is just another way to calculate an "average" of values, generally described as more suitable for ratios (such as precision and recall) than the traditional arithmetic mean. Useful when there is imbalance between number of entities and non-entities.

Accuracy the base metric used for model evaluation is often Accuracy, describing the number of correct predictions over all predictions. While precision, recall and f1-score focuses on entity level performance, accuracy measures proportion of correctly labeled tokens both entities and non entities.

Quantitative measure results of the model performance extracted from the results and the test set:

	P	R	F
currency	100.00	97.96	98.97
supplier name	69.77	78.95	74.07
invoice number	95.12	100.00	97.50
invoice date	100.00	100.00	100.00
expiration date	90.24	97.37	93.67
buyer name	85.71	82.35	84.00
quantity	100.00	100.00	100.00
order number	95.24	97.56	96.39
buyer p.iva	100.00	100.00	100.00
buyer address	78.72	84.09	81.32
payment terms	90.00	96.43	93.10
amount	41.67	16.13	23.26
cost	100.00	95.24	97.56
description	54.29	57.58	55.88

Qualitative insight of this NER model correctly recognized entities, false positive and missed entities. Correctly recognized entities, *invoices number*, *currency*, *invoice date*, *supplier name*, *expiration date*, *buyer name*, *buyer address*, *order number*, *cost*, *quantity*. Missed entities such as *row total*.

In comparing the performance of custom NER model with existing NER tools these configuration of chosen hyper parameter and train the model used.

```
BATCH\_SIZE\_FACTOR = 4
MAX\_EPOCHS = 0
MAX\_STEPS\_RATIO = 100
PATIENCE\_RATIO = 3
EVAL\_FREQUENCY\_FACTOR = 2
LEARN\_RATE = 0.001
DROPOUT = 0.25
OPTIMIZER: Adam
```

Learning rate effects: determine step size at which the model updates its weight while training and adjusting to it too high may lead to model failure

leading to instability on the other hand too low may slow or stuck in local minima making 0.001 learning rate improve model performance and control how quickly the model updates its parameters. 4 data samples or number of training examples were used in each iteration during training. Model trained for maximum of 100 times the number of steps in training dataset also indicates the stopping criterion based on ratio of training steps. Adjusting to 3 patience ratio ensures to stop early if validation loss does not improve for 3 consecutive epochs to prevent overfitting. Configuring 2 eval frequency factor means that the model will be evaluated on validation dataset every 2 epochs to enable track performance of the model. Configuring 25% dropout rate ensures used techniques to prevent overfitting. 0 max epochs set maximum number of complete passes or epochs through training dataset or with no limit on the number of epochs till stopping criterion is met. Adam optimizer used to update model performance. I used threshold 500 values scores.

Real world application NER model performance improves efficiency in handling invoices extracting key labels or information automation such as supplier name, invoice number currency ...etc.

Chapter 5

Annotation with Prodigy Tool

Prodigy is a modern annotation tool for creating training and evaluation data for machine learning models. We can also use Prodigy to help us inspect and clean our data, it has a user-friendly interface to easily assign labels to specific region and do error analysis and develop rule-based systems to use in combination with our statistical.

Creation of file before labeling

At this stage before we start the labeling data using Prodigy tool, we converted all the cloned data into readable text using tesseract OCR technique.

Then we prepared the created input file to be uploaded to the prodigy tool to build labeling for dataset. How to create such input file. The steps are:

- We create the files using "create_prodigy_input.py" python script.
- We install the prodigy tool on local machine.
- Then you upload the files to prodigy tool then
- start the labelling.

Organizing the supplier data:

Inside the directory *"labeling_ner"* we have *"ddt_ biniam"*. This subfolder contains different provider name such as Alpha media, Enterconn, KPNX ...etc with their selected 10 documents each. The output is saved in "prodigy_dataset" directory. We have subfolder *"labeling_ biniam"* and is parameter we mention when running the script while *"ddt_ biniam"* is its parent directory. When generating text format for prodigy we use this script *"create_ prodigy_ input.py"*

with specific parameter such as *"—ddt-dir"* pointing to *"ddt_ biniam"* and *"—dataset-name"* set to *"labeling_ biniam"*.

Executing these script transforms the document into text format extracting important information for annotation or labeling on prodigy.

These steps show us creation of file, dataset creation and organization of invoice supplier data. The output data is as follows:

- "labeling_ner"
- "prodigy dataset"
- "labeling_biniam"
- "Txt"

Inside the "Txt" directory we do have separate subdirectory for each 15 suppliers, and all are in the form of text format proper for NER labeling using prodigy tool.

When extracted, the whole documents splitted into a total of 253 ddt (number of documents) and 884 pages.

```
c:\\Users\\Administrator\\Desktop\\ddt-extractor>python
   .\\labeling\_ner\\create\_prodigy\_input.py --ddt-dir ddt\_biniam
   --dataset-name labeling\_biniam
```

This above script generates the text format of those suppliers PDF documents.

5.1 Installation process and setup on your local machine

I used python 3.8.10 version and I set up prodigy tool for labeling using a recipe decorator instead of using web server to work my python function on my local machine windows command line utility.

pip install prodigy -f https://213B-A7B3-30CE-1D03@download.prodi.gy/

5.2 Annotating with Prodigy workflow and methodology

For the annotation process the context and documents were obtained from the Rossum repository sources. Those documents were invoices subjected from business sights or domain and they were a collection of multiple suppliers. The importance annotating relevant data from this chosen domain was to automate the document processing for specific template.

To insight the nature of documents, they were a PDF format, and some were mixed of scanned blurred images. They also had different length page numbers some were with one page number other with more than 15-page numbers. To improve the robustness of the annotation process, for each different providers I selected sample representative documents based on increased number of required available entity names for the annotation process considering minimization of potential biases. The process of annotation was with a pre-processed text which was extracted using the Tesseract OCR system framework.

The annotation process was manually done using prodigy tool for labeling the specific entities found in the invoice documents. The need to annotate the label was to collect the desired relevant output data for the next step machine learning training model to align document processing automation.

The series of task was, after outlining the sampled representative document of various providers and extracted into text with tesseract tool, the file then uploaded into the prodigy. Then after launching the annotator using command prompt the extracted context would preview in its dashboard using any web browser. The quality of the extraction was dependent on the type of the file as some had blurriness noises. The required entity label needed to be searched manually for the annotation process was 15, such as Invoice Number, Invoice Date, Order Number, Currency, Amount, Buyer Name, Buyer Address, Buyer P.IVA, Supplier Name, payment terms, Expiration date, Description, Quantity, Cost, and some additional were also separately saved in text editor file. During annotation the preview to annotate was in sequence of page numbers and till completing the extracted context from the same invoice provider for each document.

To ensure the accuracy and consistency of annotation methodology I have been guided by my mentor to follow labeling manually to the defined entities and what decisions to take to encounter or handle to the ambiguity words found unavailable entity. It was really hard and time-consuming task to annotate the desired entity names as it asked you to carefully annotate to help get good

performance result for model training.

To facilitate all the annotation process, I configured and customized prodigy tool in my own local machine. I am preparing the annotation invoice entity in the prodigy tool to include them in the final output for the training step or process.The following section shows illustrations of challenges occurred after completing labeling Data.

5.3 Highlighting entity information

To illustrate the work during annotating the data.

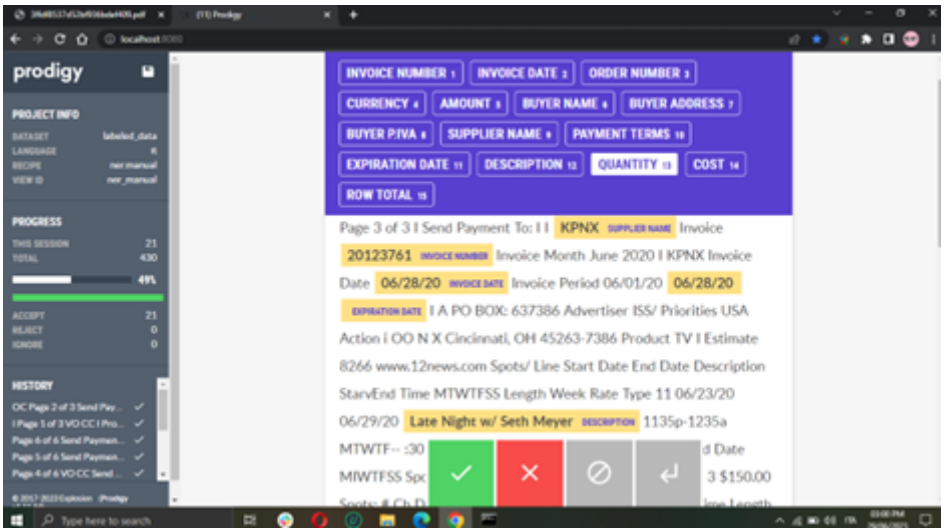
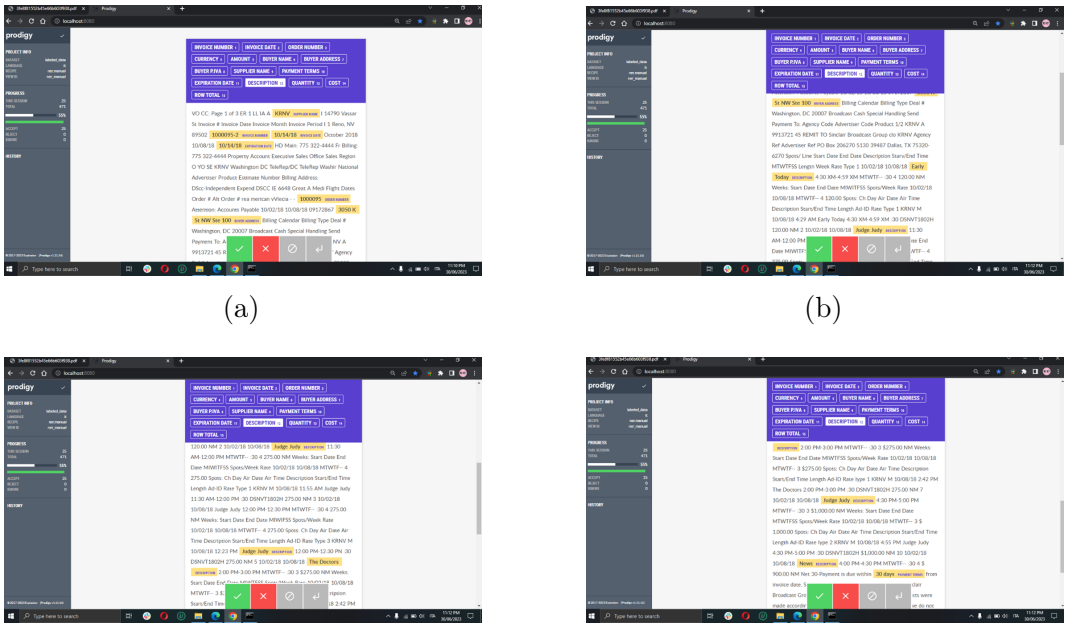


Figure 5.1: annotation process within prodigy

For better illustration whole screenshot of the file during the process of annotating we can see down saved four figures.



(a) (b)

(c) (d)

Figure 5.2: Overall annotation process within prodigy

5.4 Challenges and observations

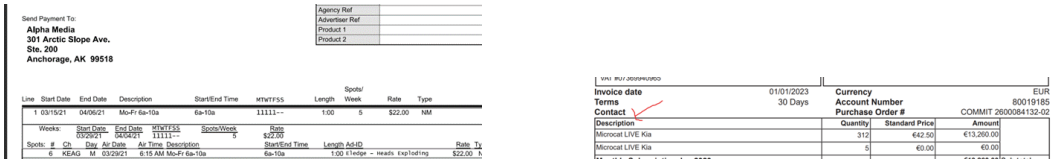
While working with prodigy highlighting the entity names, I used to experience multiple observations and challenges which makes them worth mentioning.

- When classifying them I have faced ambiguity while using the prodigy tool in assigning each invoice entity names representative within the invoice PDF as it was hard to understand which term represents which.

- There was some extracted text not very well-prepared due to the structure and blurred effect occurred on the invoice document. For this reason, when annotating the invoice entity, I could not find or match the exact referring point and due to large amount of data it was very confusing to search manually to confirm and proceed with the annotation. So, opening the document separately using PDF reader and double checking the term similarity was time consuming to approve its right instruction.

- Entity name called *Description* I found in the USA invoice information was written in terms of days and hours. For example, written as *mo-fr 6a-10a* represent days of the week. But what I found in the European is used to represent list of items or products detail. For example, both

those down below files explain how the invoice entity name "Description" represented in different way. Look in this attached down below PDF files.



(a) (b)

Figure 5.3: Illustrating differences for the entity name Description

- Some extracted files have few entities name while other not, I used to select what was available regardless the existence of all entity name. Another issue is if the e-invoice has many pages, then when you do the annotating in prodigy the pages does not go in sequence number when you click the green "accept" button. The pages shown were fluctuating, for example from page1 it goes to page 8 instead of page 2, for this reason, you need to careful when selecting the "description" part of the entity name more because mostly the other entities inside one invoice should be the same and no need to worry. the page I needed to careful was here in picture below...

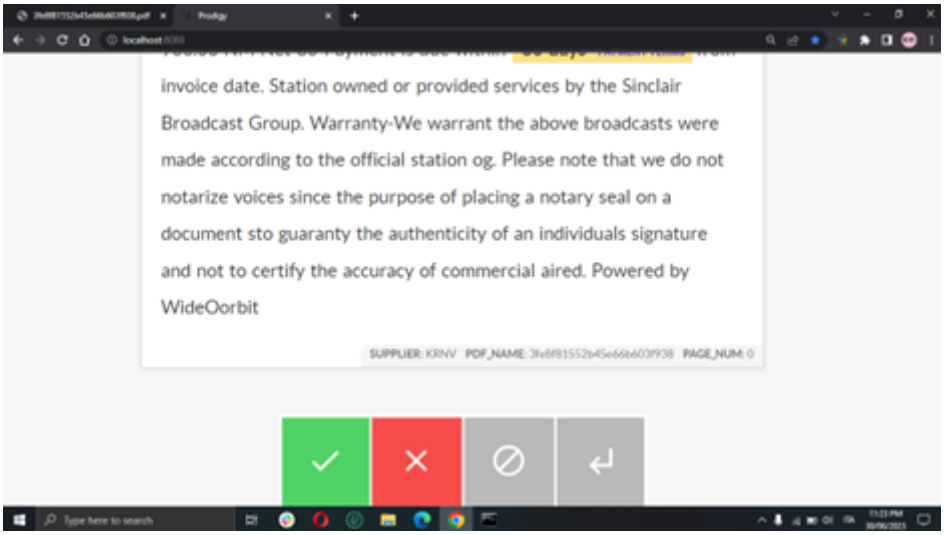
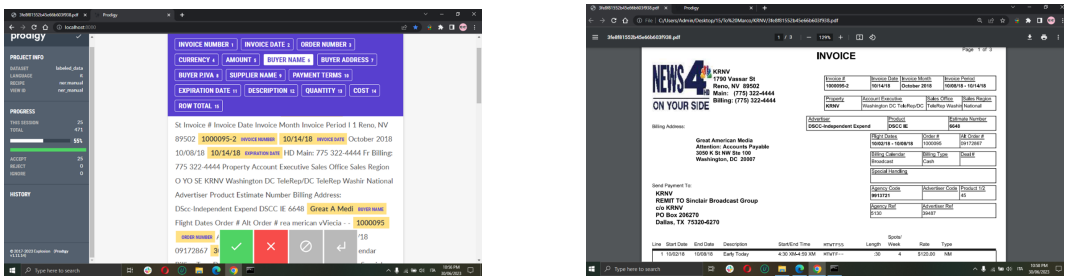


Figure 5.4: where to require attention for the page number

- During annotating inside the prodigy tool, I was observing that I could not literally find which word I need to select for a more and better training as of the model. I assumed what if I could select or leave it how much would be the short coming or side effect, I could experience during training step. For example, when I search the invoice entity name which is "buyer name" I could see the correct name in the e-invoice document itself, but when it was extracted into text for the annotating step some were extracted with missed words. For instance, in the e-invoice the buyer's name is written with "great American media" within the document having a supplier's name KRNV, but when I used to annotate to its extracted context, I see the buyer's name is extracted with a missed word, like "Great A Medi". I used to leave it because feeding to the model with a missed word may lead to an inaccurate training result. We can see in down below example picture. I tried to high light just for sample but did not label/annotate it for training, for more you can see from the e-invoice pdf taken picture itself and during the process of annotating.



(a) during prodigy (b) in the document

Figure 5.5: shows occurrence of missed word

- I see the correctly extracted entity name is placed in another position or location during annotation within prodigy. But the correct place is extracted incorrectly with a missed-text, so as long as they represent the same name, I chose it, because sometimes the entity name is repeated many times within invoice document. You can look at the picture below.

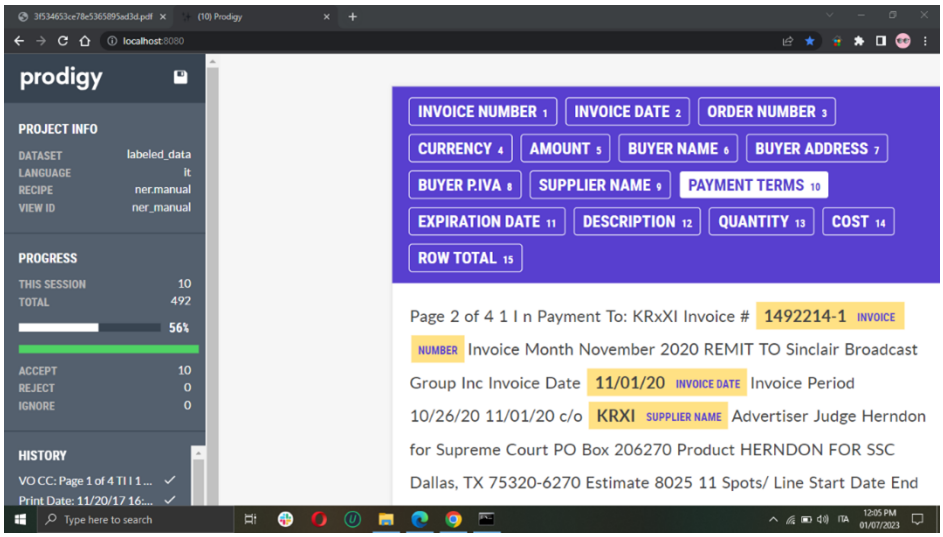
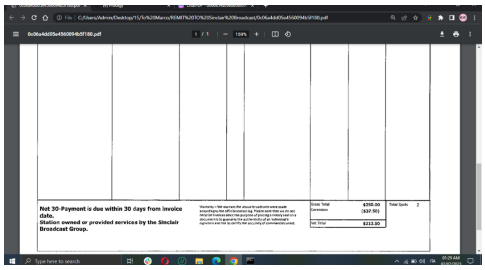
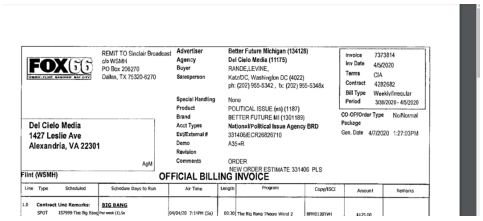


Figure 5.6: illustrates extracted label placed in different location

- You can experience much confusion to decide in representing entity name as long as the provided invoice have different structure and way of representing the entity names. For example, in the invoice document or the supplier's name *"REMIT TO Sinclair Broadcast"*, the "payment term" entity name is represented in two methods. First as *"Term= CIA"* and in second you can find mostly down to the row or table in invoice PDF document represented as *"30 days"*. We can illustrate down two taken pictures. I selected the *"CIA"*, because it's clearly also stated in other invoice providers and minimizes the confusion.



(a) payment term position in row or lower page



(b) payment terms position in header or upper page

Figure 5.7: occurrence of payment terms label in different position

- Often, I see the "quotation mark [']" were missed when extracted in the context. for example, as we see in "buyer name" the quotation mark is missed when annotating process takes place.

Figure 5.8: illustrates missing character within prodigy of the same file

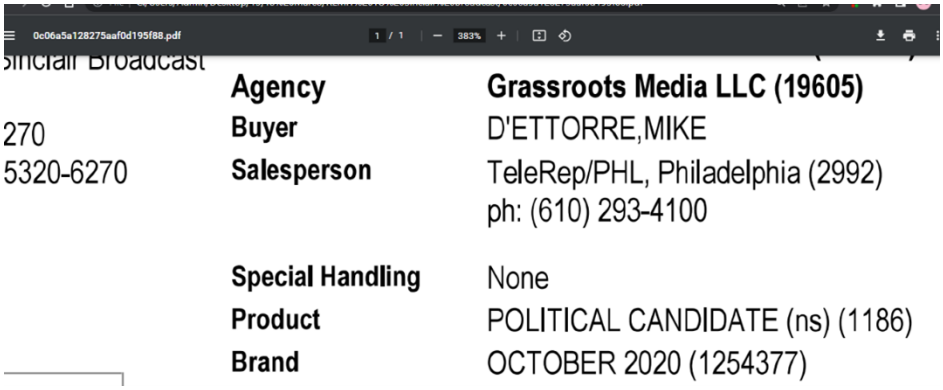


Figure 5.9: illustrates character within the document of the same file

- You find it correctly extracted, but you observe there it has space among them which are empty characters. Additionally, the dollar sign was used to be missed often when using the amount entity name in the extracted.

- Often they highlight the same invoice supplier or provider name, but they have written in a slightly different name. For instance, if we take the supplier name "Townsquare". We see their name slightly different, in figure 5.10, Townsquare Media Tuscaloosa. in figure 5.11, Townsquare Media-Trenton, in figure 5.12, Townsquare Media Battle Creek-Kalamazoo.

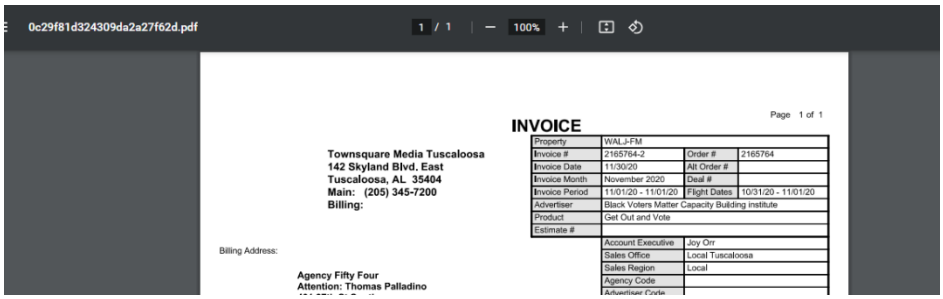


Figure 5.10: writing name representation of label in another document

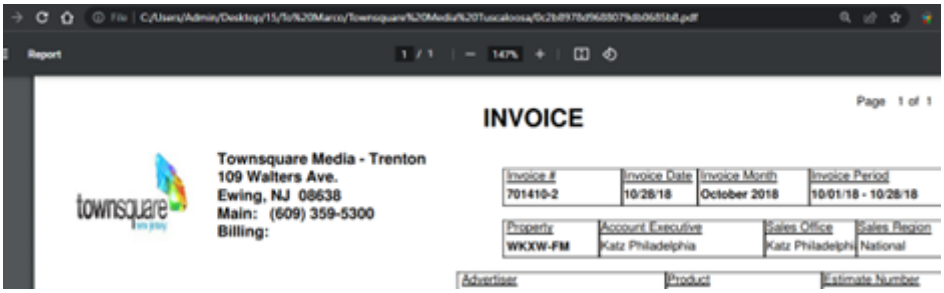


Figure 5.11: writing name representation of label in another document

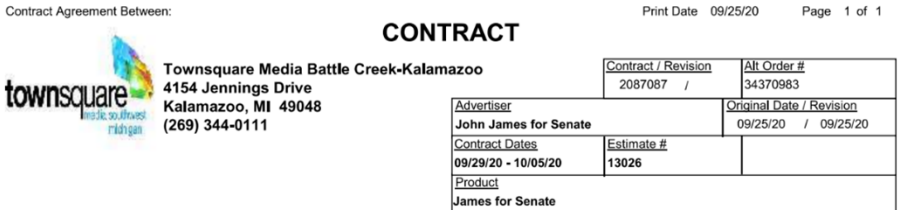


Figure 5.12: writing name representation of label in another document

- I find an invoice PDF that has two different supplier names merged together within one document and makes little ambiguity to annotate, as both even have few available invoice entity names.

Chapter 6

Results, Analysis, and Discussion

6.1 Presentation and analysis of experimental results

The proposed model system evaluated through experiments and evaluations. The machine learning NER model and Glove or Word2vec embeddings was implemented. The results shows that 0.863 F1-score which tells us the model has potential to be deployed in business sector document classification. Metrics such as recall, precision and f1 -score. A true positive rate known as recall also measures each entity label with value of 85.86% and precision with value of 86.91%. Overall, the model achieved good results compared to state-of-the-art system required training on large dataset, its performance was evaluated using custom model to extract metadata from commercial invoice data and to assess its effectiveness and performance different OCR system frameworks were used in identifying and getting the required relevant data.

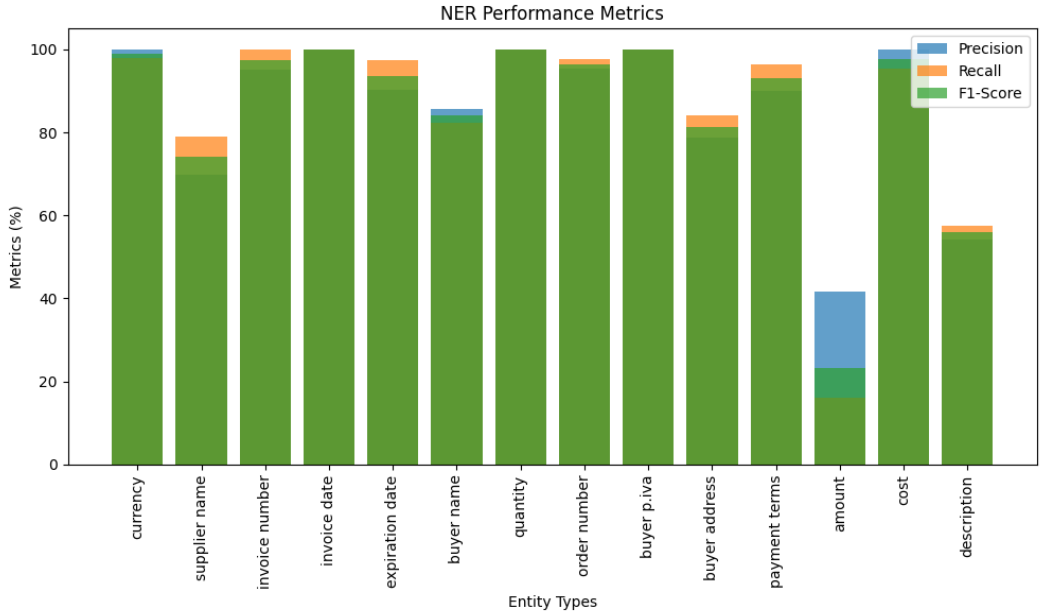


Figure 6.1: NER performance Metrics

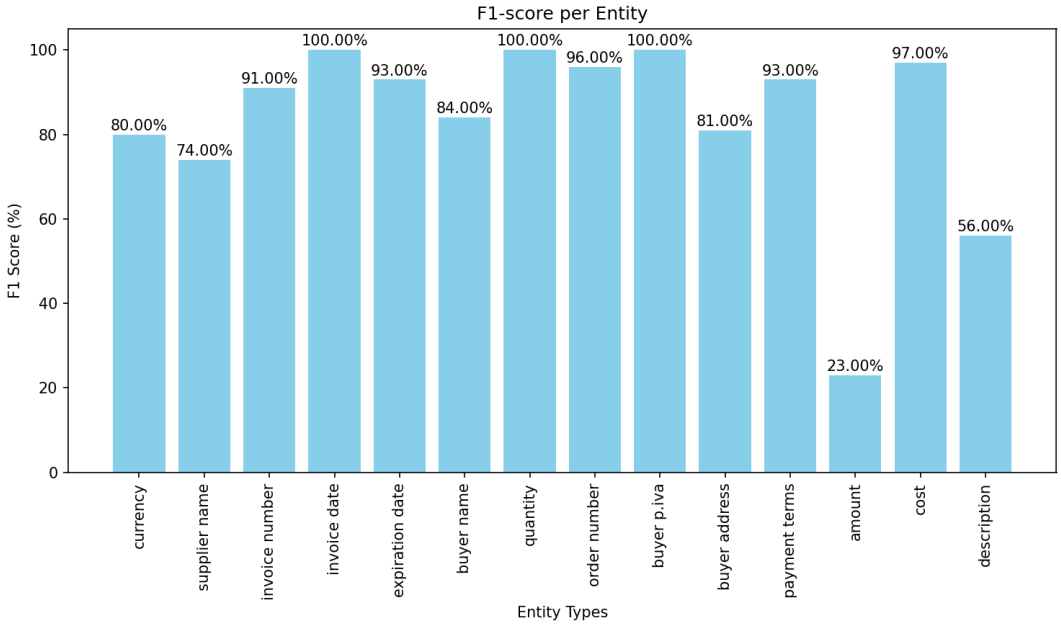


Figure 6.2: F1-score per Entity

6.2 Performance evaluation of the developed system

The performance of the NER model can be evaluated based on predefined criteria and metrics such as:

- Scalability: can be evaluated by testing it on large invoice dataset and be able to handle high volume invoices with good performance.
- Robustness: be able to handle variety of invoice formats
- Comparative analysis: the model can be compared with other document processing system and OCR frameworks.
- Accuracy: the model can also be measured with extracted information with ground truth using the metrics.

The OCR system framework was evaluated based on the metrics for each entity type. For instance, the precision for currency entity name is 98.97% and recall is 98.96%, these metrics gives insight to the OCR system framework to identify and classify entity in the document.

6.3 Discussion of findings and insights

The ai solution developed and aims to automate document processing to improve efficiency. Which primarily aims at building custom NER model to extract metadata from commercial invoice data formats and was implemented with spaCy, and its performance is evaluated by gathering metrics and results. My thesis also explores techniques to optimize quality of documents to improve OCR performance including Gaussian noise reduction techniques to increase invoice image quality.

During the experimentation some challenges been encountered such as ambiguity in different perspective at time of preprocessing and while working with prodigy. Among them "Description" label had separate data from region to region in terms of days and in other as in terms of product, I pointed it out earlier chapters. Occurrence level among labels used to exist with lower annotation rate for certain entities. Overall, during the work various unexpected findings and challenges were revealed. Impact on configuration, training data and parameters to the custom model in the other hand such as a learning rate used 0.001 which expected to improve model performance and during training four number of training or data samples used. The same for evaluation frequency and Adam

optimizer to update model performance. Document processing automation in combination to machine learning utilizing an OCR, Artificial Intelligence(AI) and natural language processing can make information extraction and do automate document processing by classifying unstructured text to a structured text. Convolutional neural network a deep learning model has been applied to OCR to recognize and interpret scanned text and Word2vec a word embedding methods were also used to represent semantic meanings of text in natural language processing.

6.4 Comparison with existing approaches or systems

The model is competitive with small data and achieved competitive result compared to state-of-the-art system which require training on large datasets. Another benefit is it extracts desired metadata from commercial invoice formats. And the system compares various optical character recognition to ensure using a better accurate text recognizer. To mention a draw back its a language limited support even though it supports many languages but comparing to other available competitors in market is limit.

To address the limitation and challenges it aims by encouraging the artificial intelligence service it focuses automating document processing using nlp and machine learning techniques which is a custom NER model to overcome the manual document processing.

6.5 Limitations and future research directions

This AI based document processing is mainly focused on developing solutions for automating documents using natural language processing and machine learning addressing challenges with information extraction. Scalability issues the thesis includes various OCR framework like EasyOCR, tesseract and PaddleOCR which may face scalability issues when processing huge number of documents and the performance of the custom model is assessed on the output of OCR system which could impact the efficiency. Because without it may not perform optimally if lack of sufficient and high-quality training data as used a denoising techniques to a scanned text to improve invoice images.

To address these limitations using a diverse and huge amount of data would be recommended and enhancing OCR performance with gaussian Noise reduction and Microsoft old photo restoration. Future work highlighting the key points is

to increase the accuracy of the Named Entity Recognition (NER). Researchers can improve the training data quality with quantity of the data and to explore advanced deep learning model for better entity information recognition. Optimizing OCR for specific document types can involve developing special OCR models best tailors to various document formats and improve image preprocessing techniques and using domain specific knowledge for better result. And exploring new AI techniques can include investigating the use of reinforcement learning , exploring transformer models for Named Entity Recognition and OCR and enhancing the unsupervised learning methods for extraction of data from unstructured documents.

Bibliography

[1] Ta Hang Chen. *An Artificial Intelligence Based Approach to Automate Document Processing in Business Area.* Massachusetts Institute of Technology, Cambridge, MA, June 2021.

[2] Rossum. Docile invoice dataset. 2023.

[3] R. Ramachandran and K. Arutchelvan. Named entity recognition on bio-medical literature documents using hybrid-based approach. *Journal of Biomedical Informatics*, 63:103383, March 2021.

[4] Dipali Baviskar. Multi-layout unstructured invoice documents dataset: A dataset for template-free invoice processing and its evaluation using ai approaches. *IEEE Xplore*, July 2021.

[5] Kamlesh Kumar, Prince Kumar, and Dipankar Deb. Artificial intelligence and machine learning based intervention in medical infrastructure: A review and future trends. January 2023.

[6] Graham A. Cutting and Anne-Françoise Cutting-Decelle. Intelligent document processing - methods and tools in the real world. *arXiv preprint arXiv:2112.14070*, 2021.

[7] J. Cardenosa and J. A. Espinosa. Document classification intelligent system in complex organizations. 2:1885–1888, October 1997.

[8] J. T. L. Wang and P. A. Ng. Texpros: an intelligent document processing system. Volume 3:103–135, 1994.

[9] Marcin Namysl and Iuliu Konya. Efficient, lexicon-free ocr using deep learning. *arXiv preprint arXiv:1906.01969*, June 2019.

[10] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*, October 2013.

[11] Data Science Stack Exchange. machine learning - does sum of embeddings make sense?, 2021.

[12] Abidi Ali Awan. What is named entity recognition (ner)? methods, use cases, and challenges. *Datacamp Blog*, September 2023.

[13] paginebianche.

[14] Communi italiani.

[15] Azienda.

[16] currency codes.