The World's Local Training Provider

NobleProg

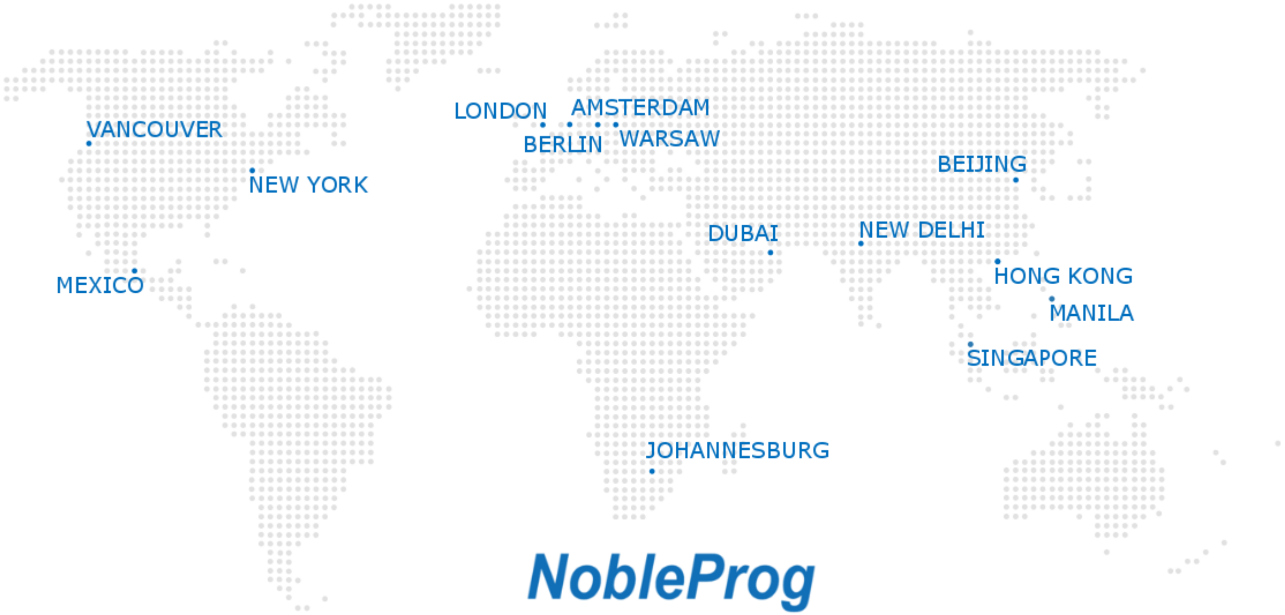
UiPath for Document Understanding

Pablo Maldonado, PhD

November 28th – December 2nd 2022

London, United Kingdom.

About NobleProg



NobleProg
The World's Local Training Provider

2005

15 +
years of experience

13 +
offices all over the world

600 +
trainers cooperating with NobleProg

1400 +
course outlines offered

6100 +
companies that entrusted us

58 k. +
satisfied participants

How to Start a UiPath Document Understanding Project

Building the Scenario - Checklist

Category	Description
Source	• What is the source of the documents? • Frequency on which they are processed?
File Structure	• What type of files (PDF, jpeg, etc)? • Password protected? • How is data presented in these files?
Classification	• What are the different types? • Can a file correspond to multiple categories? • Do the files need to be split before DU? • ➔ **Recommended** 10-50 docs per layout, single page.
Required Data	• What data is required? • In which format is it presented? • Is it structured/semi-structured/unstructured?
Document Type Structure	• What is the layout of the document? • How many different layouts per document type?
Validations	• What are the business rules required to verify the accuracy of the data? • Is manual verification possible?
Post-processing	• What is the destination system?

Planning the Solution

Process 1

Document Collection

Process 2

Document Understanding

Process 3

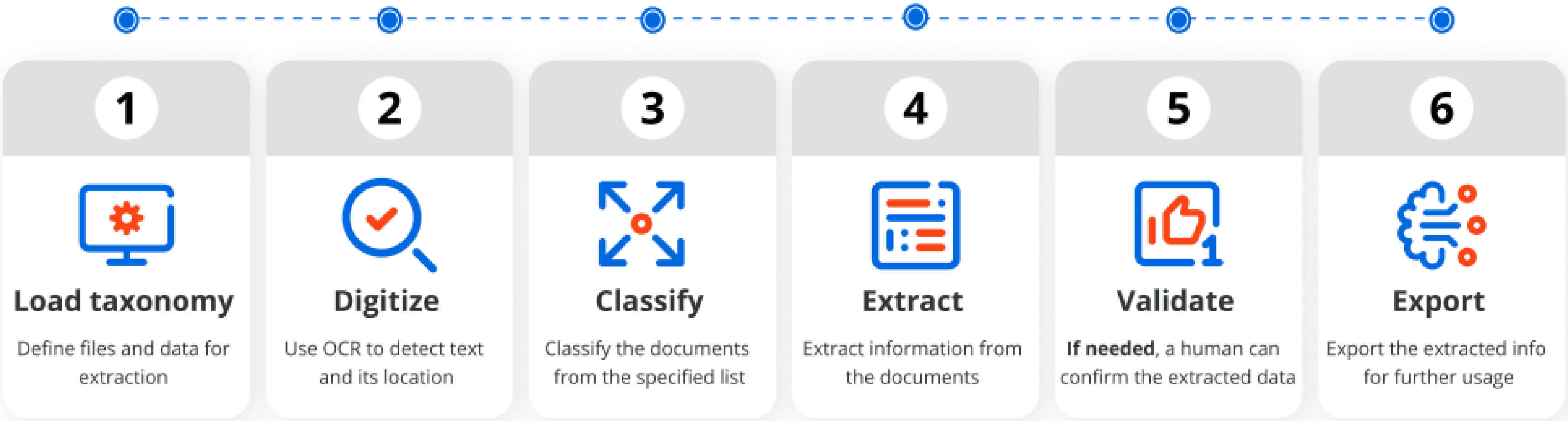
Post-processing

Document Collection Module

- Besides collecting and preprocessing documents, it is recommended to have a mechanism to ensure we don't process the same document multiple times in case the process runs into an unexpected error and needs restarting.

- The source system may provide us with information to identify new documents. Hence, it's essential to use a secondary data store to maintain a record of items. The monitoring can be done through Orchestrator Queues, Data Service, Excel, or a SQL database.

Document Understanding



1 **Load taxonomy**
Define files and data for extraction

2 **Digitize**
Use OCR to detect text and its location

3 **Classify**
Classify the documents from the specified list

4 **Extract**
Extract information from the documents

5 **Validate**
If needed, a human can confirm the extracted data

6 **Export**
Export the extracted info for further usage

NobleProg

Document Understanding (cont.) - Taxonomy

- It is essential to define the document types at the most granular level according to business requirements and group them accordingly.

- Avoid scenarios where each type contains multiple types of documents. For example, "Requisition for Release of Information" may contain court orders, court reports, email communications, requisition forms, and many more.

Document Understanding (cont.) - Digitize

- No OCR is perfect, and it is hard to know which will perform better on any given task.
- If possible, during the pilot phase create a benchmark of different OCR engines.

Document Understanding (cont.) - Classify

- Use **Keyword Based Classifier** if the documents are straightforward and if you can easily define the unique keywords

- Use **Intelligent Keyword Classifier** if the file contains multiple document types that require splitting

- Use **Machine Learning Classifier** if the documents are mainly unstructured and have many variations making it challenging to specify unique keywords

Document Understanding (cont.) – Extract

Depending on the document layout, one can use different extractors:

1. Structured and has a fixed number of rows in tables? **Forms AI**, **RegEx Based Extractor**
2. Structured, but doesn't have a fixed number of rows in tables? Check out **Form Extractor** (multiple templates), **Forms AI**.
3. Semi-structured and many layouts? Loot at **ML Extractor** (custom or out-of-the-box)
4. Unstructured – language analysis models based on the values to be extracted (**NER**, **Semantic Analysis, Classification**).
5. Is it required to check the availability of signatures and checkboxes? Use **Form Extractor**
6. Is it required to identify a signature and extract it for comparisons? Object detection models or **Form Extractor** (crop and extract using signature area coordinates).
7. If you see a combination of all of the above, we can use multiple extractors as required. We can use multiple extractors in scenarios where one extractor isn't performing well on certain documents or patterns. The secondary extractor can help on such occasions.

References

UiPath Document Understanding Solution Architecture and Approach | by Lahiru Fernando | Medium

How to Start a UiPath Document Understanding Project | Community Blog

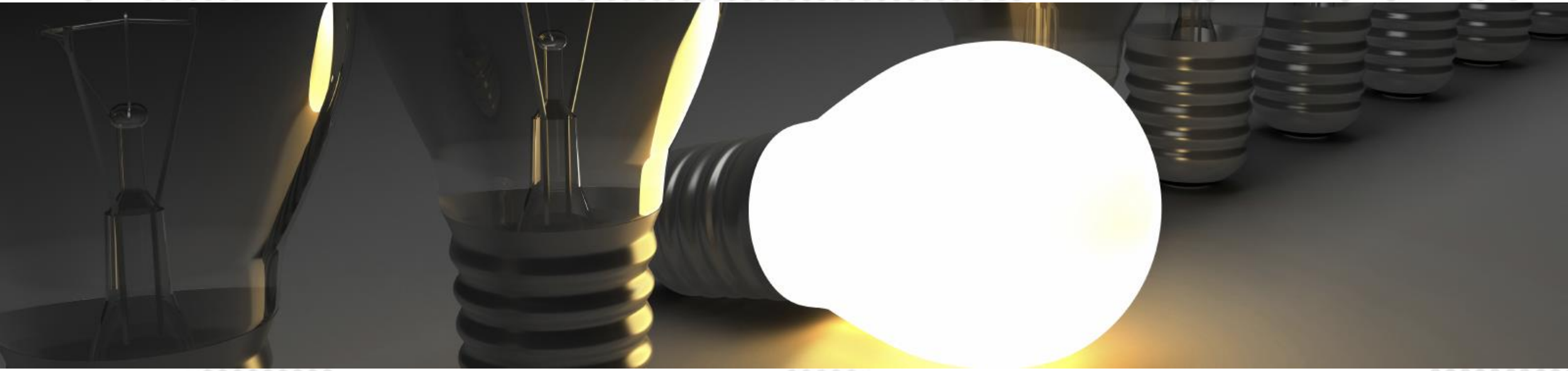
The Business Process for Document Understanding - UiPath Studio Template Explained (lahirufernandocreates.com)

AI-Based Document Data Extraction | Community Blog (uipath.com)

Any questions?

Q&A

Questions & Answers



Thank you!

Contact

NobleProg UK, Ireland and Netherlands
Training Coordinator

London: +44 (0)208 089 0990
Email: training@nobleprog.co.uk

Dublin: +353 (0)19 069 666
Email: ireland@nobleprog.ie

Amsterdam: +31 208 080 666
Email:opleidingen@nobleprog.com

www.nobleprog.co.uk
www.nobleprog.ie
www.nobleprog.nl