

# Automatic Information Extraction from Investment Product Documents

Federico Maria Scafoglieri<sup>1</sup>, Alessandra Monaco<sup>1</sup>, Giulia Neccia<sup>1</sup>, Domenico Lembo<sup>1</sup>,  
Alessandra Limosani<sup>2</sup> and Francesca Medda<sup>2,3</sup>

<sup>1</sup>Sapienza Università di Roma, Piazzale Aldo Moro, 5, 00185 Roma RM

<sup>2</sup>Commissione nazionale per le società e la Borsa, Via Giovanni Battista Martini, 3, 00198 Roma RM

<sup>3</sup>University College London, Gower Street, London, United Kingdom

## Abstract

In this paper we report on the activities carried out within a collaboration between Consob and Sapienza University. The developed project focus on Information Extraction from documents describing financial investment products. We discuss how we automate this task, via both rule-based and machine learning-based methods, and describe the performances of our approach.

## Keywords

Information Extraction, Financial documents, Financial Market Vigilance, Table Extraction

## 1. Introduction

In Italy, *Consob* (*Commissione Nazionale per le Società e la Borsa*) is the supervisory and regulatory authority of the financial market. Among the several functions, Consob has the role of monitoring and supervising any financial instruments that are issued, with the ultimate aim of detecting and enforcing against illicit conduct.

The specific project we will discuss in the following, besides Consob, involves researchers from Sapienza University of Rome. To support Consob in its supervision activities, and in particular in the task of verifying the correctness and completeness of the information it collects about financial investment products, within this collaboration a solution for the automatic extraction of structured information from free text documents has been developed. Through such solution, relevant data contained in the documents are not to be manually identified and extracted, but can be collected through an automated process which makes them available in a machine processable format. To this aim, two approaches to Information Extraction (IE) have been adopted: rule-based and machine learning (ML)-based. In the following we describe them, explain why both are useful for our goals, and report on first the results we obtained in the project.

---

SEBD 2022: *The 30th Italian Symposium on Advanced Database Systems, June 19-22, 2022, Tirrenia (PI), Italy*

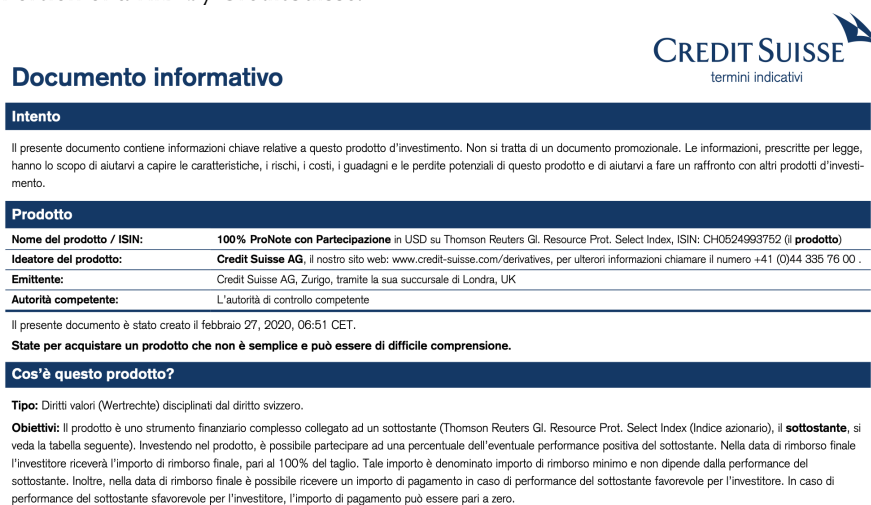
✉ scafoglieri@diag.uniroma1.it (F. M. Scafoglieri); monaco.1706205@studenti.uniroma1.it (A. Monaco);  
neccia.1847033@studenti.uniroma1.it (G. Neccia); lembo@diag.uniroma1.it (D. Lembo); i.tiddi@vu.nl (A. Limosani);  
f.medda@ucl.ac.uk (F. Medda)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

Figure 1: Portion of a KID by CreditSuisse.



**CREDIT SUISSE**  
termini indicativi

### Documento informativo

#### Intento

Il presente documento contiene informazioni chiave relative a questo prodotto d'investimento. Non si tratta di un documento promozionale. Le informazioni, prescritte per legge, hanno lo scopo di aiutarvi a capire le caratteristiche, i rischi, i costi, i guadagni e le perdite potenziali di questo prodotto e di aiutarvi a fare un raffronto con altri prodotti d'investimento.

#### Prodotto

Nome del prodotto / ISIN:	100% ProNote con Partecipazione in USD su Thomson Reuters GI Resource Prof. Select Index, ISIN: CH0524993752 (il prodotto)
Ideatore del prodotto:	Credit Suisse AG, il nostro sito web: <a href="http://www.credit-suisse.com/derivatives">www.credit-suisse.com/derivatives</a> , per ulteriori informazioni chiamare il numero +41 (0)44 335 76 00 .
Emittente:	Credit Suisse AG, Zurigo, tramite la sua succursale di Londra, UK
Autorità competente:	L'autorità di controllo competente

Il presente documento è stato creato il febbraio 27, 2020, 06:51 CET.

**State per acquistare un prodotto che non è semplice e può essere di difficile comprensione.**

#### Cos'è questo prodotto?

**Tipo:** Diritti valori (Wertrechte) disciplinati dal diritto svizzero.

**Obiettivi:** Il prodotto è uno strumento finanziario complesso collegato ad un sottostante (Thomson Reuters GI Resource Prof. Select Index (Indice azionario), il **sottostante**, si veda la tabella seguente). Investendo nel prodotto, è possibile partecipare ad una percentuale dell'eventuale performance positiva del sottostante. Nella data di rimborso finale l'investitore riceverà l'importo di rimborso finale, pari al 100% del taglio. Tale importo è denominato importo di rimborso minimo e non dipende dalla performance del sottostante. Inoltre, nella data di rimborso finale è possibile ricevere un importo di pagamento in caso di performance del sottostante favorevole per l'investitore. In caso di performance del sottostante sfavorevole per l'investitore, l'importo di pagamento può essere pari a zero.

## 2. The use case

In the EU, the creators of financial products (a.k.a. financial manufacturers) are obliged by law to make information related to so-called PRIIPs (Packaged Retail Investment and Insurance-based Investments Products) publicly available. The NCAs (National Competent Authorities) have supervisory duties on such products, so that they can be safely placed on the respective national markets. The legislation requires information about PRIIPs to be communicated to NCAs through documents called KIDs (Key Information Documents). In the practice, this means that features to be checked are cast into text reports, typically formatted as pdf files (cf. Figure 1, containing a KID for the Italian market), and extracting structured data from them (to bootstrap control activities), is actually in charge to the authority (In Italy, Consob). Due to the massive amount of documents to be analyzed (e.g., more than 1 million KIDs received by in 2020), this process cannot be carried out manually, but still it is only partially automated to date.

## 3. The approach

The KIDs present key information about financial products through free text or tables. We decided to extract relevant data contained in the free text part of the documents through a rule-based mechanism [1, 2]. This choice lies in the fact that the KIDs follow a quite rigid template imposed by the European authority, a characteristic that makes domain experts able to express quite precise patterns for the identification of wanted data, and AI experts to encode them into extraction rules. This approach has also the great advantage to provide the final user with a precisely explainable IE mechanism, a characteristic which is particularly critical in the context we operate.

However, rule-based mechanisms do not lend themselves well to the extraction of data contained in tables, which are particularly difficult to identify, depending also on the way in which the table is originally formatted [3]. For this reason we decided to resort to an approach

based on learning systems that treat a pdf file as if it were an image and aim to identify the set of pixels that are tables, together with the corresponding cells. Afterwards, the textual characters contained inside the detected cells can be recognized through OCR methods. In the following we discuss in more detail the two approaches, highlighting their complete pipelines and showing the results obtained.

### 3.1. Rule-based IE

We defined around 100 rules to extract 16 data fields, like product name, manufacturer, investment risk, from the free text contained in a KID. Below we briefly comment on the data preparation and the annotation modules that are part of our rule-based IE pipeline. A third module is in charge of exporting the extracted fields in CSV format.

**Data Preparation.** This module transforms the PDF into plain text and clean it from errors. We performed it by means of PDFBox<sup>1</sup>, an Apache state-of-the-art and highly customizable library for PDF documents.

**Annotation.** The annotation task is carried out through CoreNLP<sup>2</sup>, a popular open-source JAVA toolkit for natural language processing developed at Stanford University. For our development we relied on the CoreNLP-it module [4], specifically developed for the Italian language. We exploited fundamental services offered by CoreNLP, such as tokenization, lemmatization, POS tagging, and in particular made use of the TokensRegex component [5], which extends traditional regular expressions on strings by working on tokens instead of characters and defining pattern matching via a stage-based application. These extensions of regular expressions allow for writing *extraction rules*, i.e., rule-based extractors matching on additional token-level features, such as part-of-speech annotations, named entity tags, and custom annotations.

As an example, consider the following TokensRegex extraction rule useful for extracting International Securities Identification Numbers (ISINs) from KIDs:

```
$StartISIN = (
    /ISIN/ /:/|/Codice/ /del/ /Prodotto|prodotto/ /:/|
    ... )
$EndISIN = ( /*/ ... )
$code = "/([A-Za-z][A-Za-z][0-9]{10})/"
{ ruleType: "tokens",
  pattern: (
    ($StartISIN) (?$CodeISIN [{word:$code} &
    {SECTION:"SECTION_PRODUCT"}]+?) ($EndISIN)),
  action: ( Annotate($CodeISIN, ISIN, "ISIN")) }
```

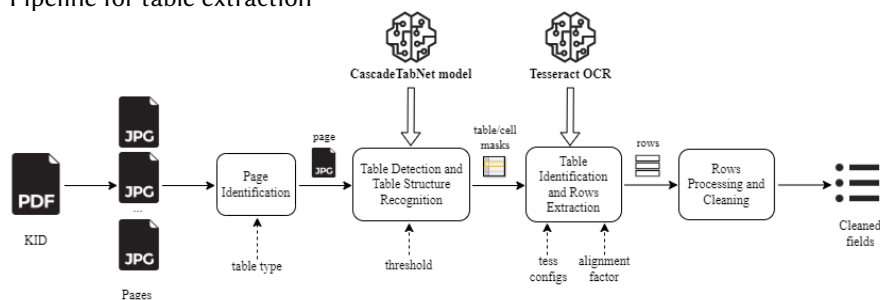
The ISIN is an alphanumeric sequence of 12 characters identifying a PRIIP. This sequence begins with two characters followed by ten numbers. In the rule above, the ISIN format is compiled into the regular expression assigned to the variable \$code. \$StartISIN and \$EndISIN represent the set of sequences of tokens (separated from each other through the symbol |) preceding and following the ISIN code, respectively. ruleType specifies that our

---

<sup>1</sup><https://pdfbox.apache.org>

<sup>2</sup><https://stanfordnlp.github.io/CoreNLP/>

**Figure 2:** Pipeline for table extraction



rule works on "tokens", whereas `pattern` contains the pattern to be matched over the text. Through the pattern above the rule looks for a matching of the regular expression `$code` over all the tokens in the product section (i.e., annotated with `SECTION_PRODUCT`), and such that `$code` is preceded by the tokens in `$startISIN` and followed by the ones in `$endISIN`. Finally `action` describes what annotation to apply. In our example, we annotate the token identified by the group `$CodeISIN` with the annotation `ISIN`. By applying this rule over the KID in Figure 1 we annotate `CH0524993752` with `ISIN`.

### 3.2. Machine Learning-based IE

We used ML techniques to extract data from the following tables:

- **Performance scenarios:** it provides data about potential risks and yield. We are interested in extracting the potential refund and the average yield per year in all the possible scenarios: stress, unfavorable, moderate, favorable, each at fixed time periods (initial, intermediate, recommended holding).
- **Costs over time:** it shows the cost trend. We want to extract data about total costs and reduction in yield (RIY) in percentage at the same time periods considered for the performance scenarios.
- **Composition of costs:** it describes the components of costs. We aim to extract the RIY for various cost categories: una tantum input and una tantum output costs, recurrent wallet transactions, other recurrent costs, performance and overperformance fees.

We modeled the problem as a Computer Vision task, converting each page of the KID into a set of images. In such a setting, the Extraction involves (1) a Table Detection task, aiming at detecting bordered or borderless/semi-bordered tables, (2) a Table Structure Recognition task, able to identify the cells belonging to the detected table, and (3) a Text Extraction task, to get the textual information contained inside each cell.

We propose an automatic, learning-based approach, exploiting the power of Deep Learning and Convolutional Neural Networks (CNN) for image processing. In particular, for subtasks 1-2 we use CascadeTabNet [6], a recent implementation of a CNN model which was trained by the authors on the detection of border/borderless tables and cells first on general tables (e.g. word, latex documents), then on ICDAR-19<sup>3</sup>. For subtask 3 instead, we use Tesseract [7], a popular

<sup>3</sup>ICDAR-19 is a dataset containing table data, that was released for a Table Detection and Recognition competition at ICDAR in 2019

Optical Character Recognition (OCR) engine.

The pipeline we realized is shown in Figure 2. The main module are:

- **Page Identification Module**, which identifies the pages of the document containing the tables we want to extract. Such tables are indeed contained in specific sections of the document, which we can recognize through a simple lookup of certain keywords, like ‘performance scenarios’, ‘costs over time’, ‘composition of costs’.
- **Table Detection and Table Structure Recognition Module**, which uses the pre-trained CascadeTabNet model to identify the masks (i.e. bounding boxes) of borderless/bordered tables and cells. The masks are returned if and only if their level of confidence is higher than the input threshold, that we experimentally tuned to 0.6. Cells are then assigned to the correspondent table depending on their positions.
- **Table Identification and Row Extraction Module**, that applies the OCR to each cell bounding box. Based on the text extracted by the OCR, we are able to understand if the considered table is the one of interest. In a single page in fact, many different tables can be present. We use the top coordinate value of the cell masks to organize them into rows.
- **Row Processing and Cleaning Module**, which produces the final output of the extraction. In particular, from an inspection of the rows created in the previous step, and on the basis on their attended structure that we know from the regulation, we are able to assign the extracted information to the correspondent information field. A final Cleaning step, deals with missing punctuations, extra space removal, OCR error correction and, numerical formats standardization.

The effectiveness of the results often depends on the table template: there are few tables that the algorithm fails to recognize at all, or for which some cells are not detected (mostly the table headers). Table/cell detection is indeed a challenging task, due to different reasons. Firstly, the heterogeneity of table templates adopted by the PRIIPs authors makes the task harder for the neural network; templates are slightly different from the ones in ICDAR-19, because they are modern, often semi-bordered or borderless, semi-coloured, with multi-line cells. The same table type may present different number of rows or columns, just because the specific kind of PRIIP requires it, or because of a different arrangement of the information in the table (for instance, a single cell contains both a cost and a RIY). Moreover, we noticed issues in identifying multi-line cells: few times true multi-line cells are detected as two distinct cells or, on the contrary, distinct cells that appear one under the other, probably very close, are detected as a single multi-line cell. We addressed this last case in the Rows Cleaning Module: using our knowledge of the table structures, knowing which field types appear one under the other, we split the wrongly multi-line cells and assign the data to the proper field. Regarding cell masks, we noticed also that, in few cases, the detected bounding box was quite inaccurate, cropping the text and consequently causing troubles in OCR. In order to avoid it, we consider an enlargement factor such that we run the OCR on a slightly larger area.

## 4. First results

Table 1 provides some information regarding the two datasets on which we tested our approach. Dataset-1 was created specifically as a test scenario. It is representative of document hetero-

geneity and includes KIDs from the years 2018-2020. Dataset-2 was selected randomly from a sample of KIDs from the first half of 2021.

**Table 1**  
Dataset Info

Dataset	Total KIDs	Size	Manufacturers
DATASET-1	1240	250MB	36
DATASET-2	7736	3GB	52

Table 2 reports an average of precision and recall over all information extracted through the rule-based approach. The performances we got testify that the effort put in designing the rules definitely pays off, since we reach very high level of precision and recall, as required in the considered application scenario.

**Table 2**  
Results for the rule-based approach

Dataset	Precision	Recall	F-Measure
DATASET-1	98%	96%	97%
DATASET-2	98%	94%	96%

Table 3 instead shows the number of tables extracted through our ML-based approach for each table type, before the Fine-Tuning phase that we will describe in the next section. We obtained the worst recall for the Costs Evolution table. This is probably due to the fact that the extraction of this table mostly relies on just two header cells in the Table Identification step, therefore, if the model fails for any reason in detecting just those two cell masks, the table is not extracted even though the numerical cell masks are often correctly detected. This issue is less critical for the other table types, since more cells can be used for Table Identification.

**Table 3**  
Results of the Extractor for the different table types.

		Dataset	
		DATASET-1	DATASET-2
<b>Performance scenarios</b>	<b>Extracted</b>	1218	6435
	<b>Missing</b>	22	1301
<b>Costs over time</b>	<b>Extracted</b>	752	3925
	<b>Missing</b>	488	3811
<b>Composition of costs</b>	<b>Extracted</b>	1153	6886
	<b>Missing</b>	87	850

## 5. Fine-Tuning to improve Information Extraction from tables

As said, to extract information from tables, we started from a model that was trained on the detection of generic tables. Since the tables in KIDs documents may present complex layouts, we fine-tuned the model using custom data, in order to improve its effectiveness. Below we describe the Fine-Tuning process we adopted.

**Table 4**

Comparing results of the Original and Fine-Tuned Extractor for the different table types.

		Model		Gap (%)
		Original	Fine-Tuned	
<b>Performance scenarios</b>	<b>Extracted</b>	1218	1237	+1.5%
	<b>Missing</b>	22	3	
<b>Costs over time</b>	<b>Extracted</b>	752	1223	+62,63%
	<b>Missing</b>	488	17	
<b>Composition of costs</b>	<b>Extracted</b>	1153	1167	+ 1.2%
	<b>Missing</b>	87	73	

**Labeling Data with Label Studio.** The first step involves creating a suitable dataset for training purposes. Here, 'suitable' means 'labeled': for each image of the dataset we have to annotate the coordinates (bounding boxes) of the tables and the cells it contains. To this aim we have used Label Studio<sup>4</sup>, an open-source flexible data annotation tool, that can be adapted to different labeling tasks (image classification, object detection, event recognition, etc.) and data type (images, audio, text, time-series, etc.), allowing different users to collaborate on the annotation process, that is well known to be highly time consuming and to require a lot of manual effort. For our object detection annotation task we customized labels in 'bordered', 'borderless' and 'cell' and we drew the bounding boxes of each label using the mouse. The tool automatically generates the bounding boxes coordinates and allows to export the annotations in different formats. We exported annotations in Pascal VOC (Visual Object Classes) XML, that creates a different xml file for each image, containing <object> elements providing annotation information, in particular the label and the coordinates of the bounding box.

**Training the Model.** The extracted Pascal VOC data are then converted into COCO (Common Objects in Contexts) format, typically required by object detection models. We trained the model using MMDetection, an open source object detection toolbox, loading the pre-trained weights of CascadeTabNet model and training for 10 epochs using our annotated data. 239 images (KID pages converted to png files) were used for the training, trying to maximize the variance of the KIDs authors and therefore, the templates. Each image page may contain more than a single table. In total, our training dataset contained 10844 objects, respectively 71 bordered tables, 373 borderless tables and 10400 cells.

**Results on table extraction after Fine-Tuning.** The fine-tuned model has been tested on Dataset-1<sup>5</sup>, showing a great improvement in extracting data from 'Costs over time' tables. Whereas the first model (non-fine-tuned) was not able to extract 488 tables (~40% of the total tables), the fine-tuned model misses just 17 tables. The gap between the two extractions is impressive, which testifies that Fine-Tuning has been particularly effective. A slight improvement has been observed in the other table types, too.

In Table 4 we give the results of the extractions of tables by comparing the first model with the fine-tuned model, whereas in Table 5 we report the average precision, recall and f1-scores

<sup>4</sup><https://labelstud.io/>

<sup>5</sup>We also tested our results on Dataset-2, but our evaluation, due to dataset size, was done on a sample basis with results similar to that reported for Dataset-1.



on all the fields extracted from the 3 table types for Dataset-1.

**Table 5**  
Results of Fine-Tuning.

Dataset	Precision	Recall	F-Measure
DATASET-1	~99%	~79%	~88%

## 6. Conclusion

We are currently working to complete IE from KIDs, by attacking some of their portions that have not been so far involved in the extraction process. The next project steps concern with post-extraction data analysis, aimed to identify anomalous documents, correct errors and missing mandatory data, discover interesting correlations among data.

We are also investigating how to link collected data to a domain ontology, to exploit reasoning services and enhance the entire data acquisition process according to the Ontology-based data access and integration [8, 9, 10]

## References

- [1] J. Doleschal, B. Kimelfeld, W. Martens, Database principles and challenges in text analysis, *SIGMOD Record* 50 (2021) 6–17.
- [2] D. Lembo, F. M. Scafoglieri, Ontology-based document spanning systems for information extraction, *Int. J. of Semantic Computing* 14 (2020) 3–26.
- [3] E. Oro, M. Ruffolo, PDF-TREX: an approach for recognizing and extracting tables from PDF documents, in: *Proc. of ICDAR*, IEEE Computer Society, 2009, pp. 906–910.
- [4] A. P. Apro시오, G. Moretti, Italy goes to stanford: a collection of corenlp modules for italian, *Comp. Res. Rep. (CoRR) abs/1609.06204* (2016). URL: <http://arxiv.org/abs/1609.06204>.
- [5] A. X. Chang, C. D. Manning, Tokensregex: Defining cascaded regular expressions over tokens, *Stanford University Computer Science Technical Reports. CSTR 2* (2014) 2014.
- [6] D. Prasad, A. Gadpal, K. Kapadni, M. Visave, K. Sultanpure, Cascadetabnet: An approach for end to end table detection and structure recognition from image-based documents, *Comp. Res. Rep. (CoRR) abs/2004.12629* (2020). URL: <https://arxiv.org/abs/2004.12629>.
- [7] R. Smith, An overview of the tesseract OCR engine, in: *Proc. of ICDAR*, volume 2, 2007, pp. 629–633. doi:10.1109/ICDAR.2007.4376991.
- [8] D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, R. Rosati, Ontology-based data access and integration, in: L. Liu, M. T. Özsu (Eds.), *Encyclopedia of Database Systems*, Second Edition, Springer, 2018.
- [9] F. M. Scafoglieri, D. Lembo, A. Limosani, F. Medda, M. Lenzerini, Boosting information extraction through semantic technologies: The kids use case at consob, volume 2980 of *CEUR*, [ceur-ws.org](http://ceur-ws.org), 2021.
- [10] D. Lembo, Y. Li, L. Popa, K. Qian, F. Scafoglieri, Ontology mediated information extraction with MASTRO SYSTEM-T, volume 2721 of *CEUR*, [ceur-ws.org](http://ceur-ws.org), 2020, pp. 256–261.