A template-independent approach for information extraction in real estate documents

Nicola Landro[1,*], Gabriele Destro[1], Stefano Taverni[1] and Ignazio Gallo[2]

[1]Digitiamo, Via Giuseppe Piermarini, 26, Varese, 21100, Italy
[2]University of Insubria, Via J.H. Dunant 3, Varese, 21100, Italy

Abstract

Business corporations manage tons of unstructured data daily, such as PDFs and websites. Recent advances in the deep learning field help find insight from this unstructured information. New models leverage the power of the Transformer architecture to accomplish natural language understanding tasks on these data, jointly using the raw image and its text content or directly the image without OCR. We propose an extraction pipeline that employs question-answering models to get insight from unstructured data, allowing fast and efficient information retrieval from different sources. We show an application of this technique to a specific set of documents and how we can scale this infrastructure to different types of records. Our solution can effectively handle large document corpora robustly, helping corporations exploit all the power coming from their data.

Keywords

Question Answering, Information Retrieval, Real Estate, Scraping

1. Introduction

Recent deep learning models significantly increase the efficiency of document analysis. In particular, the Transformers model [1] excels in a variety of text- and visual-based tasks, such as ChatGPT (recently used in financial research [2]), providing excellent opportunities to develop new algorithms and enhance existing ones, as well as to improve data retrieval.

Recent diffuser algorithms [3] appear to achieve better results today, and as with convolutions and transformers, they may become the state of the art in other subtasks such as document retrieval, but for the time being, transformers remain the state of the art in document extraction.

Language models based on transformers are highly effective for text embedding, such as Sentence Bert [4], which outperforms convolutional approaches such as Kim [5] or not such as Word2Vec [6] and Glove [7].

Also, the NER (Named Entity Recognition) problems can be covered by Bert [8], with that model we can extract tokens from a text that represents a particular entity.

To extract text from documents OCR (Optical Character Recognition) algorithms like Tesseract [9] or EasyOCR [10] are very useful also in recent days in which the PDFs are vectorial: in the wild, some documents can be scanned so they are raster documents.

Because PDFs and document scans contain a variety of unstructured data, new advances in document extraction, such as LayoutLMV3 [11], add visual document understanding of the text component. This approach combines OCR and a Bert-like model. Some contemporary solutions, such as Donut [12], present an end-to-end model that does not rely on any OCRs. This technique, in particular, employs the Swin transformer as the visual encoder and the first four-layer or multilingual Bart as the textual decoder. This method outperforms LayoutLM primarily in terms of time, but also in terms of normalized tree edit distance.

Our approach wants to restart from the first method but instead of using a Bert-like [13] model or Bart [14] general purpose and finetuned on documents we want to use question-answering models.

The T5 [15] Italian model is a multi-task text model that can be used for text summarization [16] and many other tasks, we select it for question answering by reaching a more generic method to extract specific information by documents that can be configured by setting a series of questions and keywords that extract precise information without mandatory retraining of deep models.

2. Proposed Approach

The proposed method(Fig. 1) starts from a web scraping pipeline that reaches to obtain documents and other information about buildings. Following the retrieval of

Ital-IA 2023: 3rd National Conference on Artificial Intelligence, organized by CINI, May 29–31, 2023, Pisa, Italy
*Corresponding author.
nicola.landro@digitiamo.com (N. Landro);
gabriele.destro@digitiamo.com (G. Destro);
stefano.taverni@digitiamo.com (S. Taverni);
ignazio.gallo@uninsubria.it (I. Gallo)
0000-0002-0565-7496 (N. Landro); 0009-0007-6770-5127 (G. Destro); 0009-0009-9880-2354 (S. Taverni); 0000-0002-7076-8328 (I. Gallo)

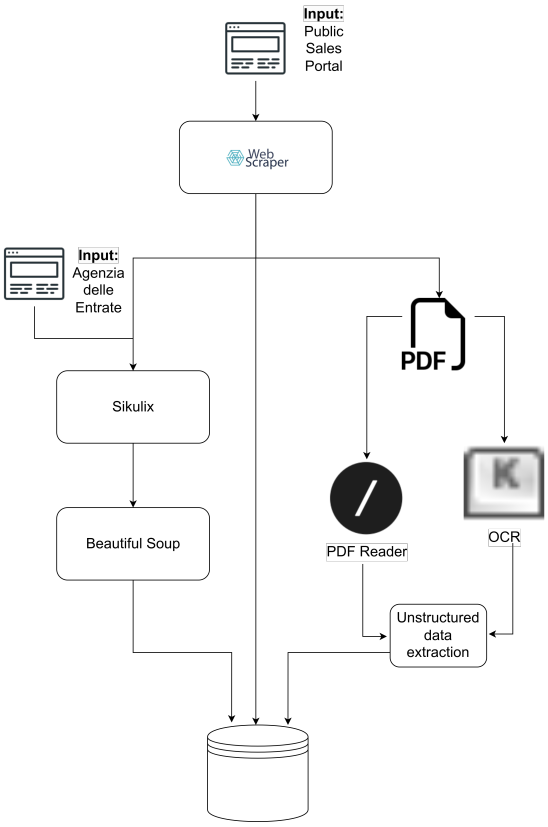

Figure 1: The flow of the proposed pipeline start from an easy web scraping of the Public Sales Portal that provides documents for the right branch that extracts information by text and the left branch that performs a deep scraping of the Agenzia delle Entrate.

the documents, the data extraction pipeline continues with an algorithmic approach that extracts information from structured documents. Deep neural models also allow the pipeline to directly manage unstructured documents, enabling effective information retrieval using a question-answering model.

The application domain is the real estate market, where information like quotations, bankruptcies, are square footage are extracted from different sources, specifically: The *Agenzia delle Entrate* official quotation website [1]; the Public Sales Portal, which publishes real estate auctions [2]; businesses' semi-annual reports.

2.1. Scraping

In order to obtain a satisfactory database, it is first necessary to build a dataset from scratch that contains an adequate amount of information on the commercial real estate market of bankrupt companies, which in particular contemplates fluctuations in property sales values, dates of sales, building leases, and square footage. The first step for the retrieval of the data includes a part of scraping from the public site of the Italian auction portal by means of graphical non-headless browser-based tools for the construction of a first dataset containing property values, property-related leases, and URLs strings related to PDF formatted documents with real estate information and appraisals. Further information on market prices for real estate in given locations is obtained through an automation tool for data extraction that uses image recognition powered by OpenCV to identify GUI components. This approach proves to be convenient in scenarios where there is limited accessibility to the internal workings of a graphical user interface or the source code of an application or web page. Varied workflows of different levels of complexity can be established and implemented by leveraging image recognition with automated tools that can be used effectively to create a parse tree that enables the extraction of data from the real estate database of the revenue agency website, especially when combined with HTML scraping tools for parsing HTML and XML documents. The last step for the construction of an ad hoc dataset and recovery of information regarding the square footage of the buildings by extracting data from documents consists in the creation of an algorithm that uses hybrid approaches to convert scanned documents into text by implementing a text extractor and a double layer of optical character recognition, consisting of the Google Tesseract model in combination with the model provided by the EasyOcr library which uses hardware acceleration (GPU). The combination of the two models also covers cases where there is no GPU available using Google Tesseract, which delivers higher performances for CPU-only-based uses. The approach proves effective and efficient. Text obtained in the previous step is finally processed by an algorithm that combines a convolutional neural network for sentence segmentation, a pattern matching algorithm that implements regular expressions mapping, and a Transformer based model, fine-tuned for the Italian language question answering, as shown in figure 1, which will be discussed further in section 2.3.

2.2. Document extraction pure algorithm

The information extraction pipeline is tailored to specific documents, namely businesses' semi-annual reports. The algorithm starts by adding an OCR layer to the original document thanks to `ocrmypdf` [3], a Python library that takes care of rasterizing, performing OCR, and saving the

[1] https://www1.agenziaentrate.gov.it/servizi/geopoi_omi/index.php
[2] https://pvp.giustizia.it/pvp/it/homepage.page
[3] https://github.com/ocrmypdf/OCRmyPDF

result back to PDF format. This new document is then read in memory using an extended version of `camelot` [4], a PDF table extraction utility written in Python. `camelot` exposes its internal API to handle PDFs manipulation and layout reading. This API has been extended to support, in addition to table reading, horizontal and vertical text extraction. A set of query functions has been added to allow efficient information retrieval, like extracting text in range, extracting words at a specific location, matching lines with a specific set of words, and extracting text in column format. Once a PDF is loaded as raw text in memory, the information extraction phase takes place. An extractor is defined as an atomic information searcher, which is a single class specialized in the retrieval and cleaning of chunks of text. A set of ad hoc extractors is created to perform an exhaustive search of all the information. Since documents might have several formats, an extractor for each document type is defined to retrieve the same information but with different queries. For each PDF, the whole set of extractors is executed to retrieve as much information as possible. Since more than one extractor might return successfully, a majority voting phase ensures that only one among the valid extractor is used. In the case of a tie, each extractor has a reliability score that determines its accuracy, and only the most accurate is chosen.

2.3. Document extraction Deep

Creating a scalable model with an algorithmic approach is a challenging task, as documents containing the desired information often have no pattern or similar regular patterns. For this reason, we present an approach that combines a pattern-matching algorithm and an artificial intelligence algorithm to effectively scale data extraction from documents.

The algorithm in detail implements a first part, or text extractor, comprising two OCRs and one extractor to retrieve text from documents. This first part of the algorithm can recognise pdf documents. Where a document has no scanned text, the PDF Reader (Fig. 1) retrieves the text. Where the first extractor does not find text one of two OCR engines, Tesseract and EasyOCR, comes into play. Both have advantages and disadvantages. Tesseract has about 94% accuracy in identifying numbers, and 98% accuracy in identifying letters, with an image-to-text conversion, instead EasyOCR has an accuracy of 98% in identifying numbers and 95% in identifying letters, with an image-to-text. Another important difference is that EasyOCR can use GPU; so we decide to use Tesseract (that has more accuracy on most of the text) when we run scripts on CPU, while we use EasyOCR on GPU devices to speedup the works 1.

[4]https://camelot-py.readthedocs.io/en/master/

Table 1
Execution time of different OCRs calculated by generating some random document with 1000 characters.

OCR	CPU time (s)	GPU time (s)
`Tesseract`	0.71	-
`EasyOCR`	0.89	0.1

In the second step of the conversion, the text is split into sentences using a sentence segmentation process provided by a convolutional neural network based on word2vec and GloVe [5]. This step leads to the generation of sentences that will be processed for correct information retrieval. The sentence segmentation process defines semantically relevant sentences in the context from which the desired information can be retrieved. The next step after sentence splitting is the application of a mapping algorithm that retrieves the desired information from the sentences by searching for relevant words and contemplating the results found by matching regular expressions in a neighbourhood of n words from the matching index. This ensures that part of the irrelevant values is filtered out by the first extraction layer. Subsequently, the relevant values for each sentence are entered into a list related to the reference sentence and mapped into a dictionary by the key-value structure, where the keys are represented by the sentences that possibly contain searched values and as values the list of values contained in the sentence. It is noticeable that retrieved values here can present some inconsistencies, as pattern matching has an inherent limitation related to the language property. From this step, artificial intelligence comes into play, with a further layer consisting of a fine-tuned T5 transformer for question answering and trained for the Italian language [6]. The transformer makes it possible to search for values in the filtered sentences and then retrieve the data having as input the keys of the dictionary mapped in the previous step, guaranteeing a second skimming of irrelevant data and checking that the value retrieved from each sentence is consistent with the values retrieved from the text. To further improve the results, an adjacency check to remove repeated values was implemented on the strings, which discards strings with the same values if their index is equal to n +1 where n is the index of the sentence under consideration. In this way, the algorithm can return the sentences with results filtered on the text with their relative values. In the case of documents concerning the real estate market, the need to calculate the total square footage for each property led to the integration of a final layer that checks by means of regular expressions and pattern matching whether the values in the text report a total quantity,

[5]https://spacy.io/models/it#it_core_news_lg
[6]https://huggingface.co/it5/it5-base-question-answering

Table 2
Experimental results of the extraction phase on a batch of 60K documents. Excluding 542 invalid documents, the pipeline successfully extracts valid information from around 18K documents. The remaining PDFs turned out to be invalid relations or false-negative parsing, with an error rate below 5%.

	Quantity	%
Total	59'939	100
Unprocessable	542	0.01
Readable	17'929	29.91
Unreadable	41'468	69.18

calculating the value if this is not returned by the first check. The algorithm is being improved by implementing an AI Named Entity Recognition module and a second transformer-based module that exploits sentence similarity logic on the tokenization of words. The idea would allow considerable improvements in terms of accuracy and completeness in handling the information sought without having to resort to pattern matching and regular expressions layers.

3. Dataset and Experiments

The dataset used involves a batch of around 60K documents. We run the pure algorithmically extraction phase on the entire dataset. The machine used for the experiments is equipped with 20 CPUs at 3.50GHz, 128GB of RAM, running Ubuntu [7] with kernel `4.4.0-170-generic`. Table 2 shows the experimental results. Apart from 542 unprocessable documents (they are encrypted, or they are not PDFs), we successfully parsed 17 929 PDFs, extracting all the relevant parts. The remaining 41 468 documents turned out to be invalid relations or relations with an unrecognized format. A manual investigation shows that less than 5% are false negatives. As a side note, the experiment required about 7 days of full computation, taking into account the overhead introduces by storing the OCR files and the extracted information on an external cloud provider.

4. Conclusion

With the purely algorithmic approach, we just reach to extract any information from data from documents with high precision, but it does not scale quickly to others document formats. The deep proposed approach is the first idea to scale document extraction information to all documents and start to fill the gap between Deep Neural models and this research field.

[7] https://ubuntu.com/

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[2] M. Dowling, B. Lucey, Chatgpt for (finance) research: The bananarama conjecture, Finance Research Letters (2023) 103662.

[3] P. von Platen, S. Patil, A. Lozhkov, P. Cuenca, N. Lambert, K. Rasul, M. Davaadorj, T. Wolf, Diffusers: State-of-the-art diffusion models, https://github.com/huggingface/diffusers, 2022.

[4] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2019. URL: https://arxiv.org/abs/1908.10084.

[5] Y. Chen, Convolutional neural network for sentence classification, Master's thesis, University of Waterloo, 2015.

[6] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781 (2013).

[7] J. Pennington, R. Socher, C. Manning, GloVe: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1532–1543. URL: https://aclanthology.org/D14-1162. doi:10.3115/v1/D14-1162.

[8] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, CoRR abs/1810.04805 (2018). URL: http://arxiv.org/abs/1810.04805. arXiv:1810.04805.

[9] R. Smith, An overview of the tesseract ocr engine, in: Ninth international conference on document analysis and recognition (ICDAR 2007), volume 2, IEEE, 2007, pp. 629–633.

[10] C. Jeeva, T. Porselvi, B. Krithika, R. Shreya, G. S. Priyaa, K. Sivasankari, Intelligent image text reader using easy ocr, nrclex & nltk, in: 2022 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS), IEEE, 2022, pp. 1–6.

[11] Y. Huang, T. Lv, L. Cui, Y. Lu, F. Wei, Layoutlmv3: Pre-training for document ai with unified text and image masking, in: Proceedings of the 30th ACM International Conference on Multimedia, 2022, pp. 4083–4091.

[12] G. Kim, T. Hong, M. Yim, J. Nam, J. Park, J. Yim, W. Hwang, S. Yun, D. Han, S. Park, Ocr-free document understanding transformer, in: European

Conference on Computer Vision (ECCV), 2022.
[13] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423. doi:10.18653/v1/N19-1423.
[14] S. Broscheit, M. Poesio, S. P. Ponzetto, K. J. Rodriguez, L. Romano, O. Uryupina, Y. Versley, R. Zanoli, BART: A multilingual anaphora resolution system, in: Proceedings of the 5th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Uppsala, Sweden, 2010, pp. 104–107. URL: https://aclanthology.org/S10-1021.
[15] G. Sarti, M. Nissim, It5: Large-scale text-to-text pretraining for italian language understanding and generation, ArXiv preprint 2203.03759 (2022). URL: https://arxiv.org/abs/2203.03759.
[16] N. Landro, I. Gallo, R. La Grassa, E. Federici, Two new datasets for italian-language abstractive text summarization, Information 13 (2022) 228.

A. Online Resources

We provide an HuggingFace Space that will be published if accepted.X