Position Papers of the Federated Conference on Computer Science and Information Systems pp. 9–16
DOI: 10.15439/2018F373
ISSN 2300-5963 ACSIS, Vol. 16

Robotic Process Automation of Unstructured Data with Machine Learning

Anna Wróblewska*,†, Tomasz Stanisławek*,†, Bartłomiej Prus-Zajączkowski*,†, Łukasz Garncarek†
*Faculty of Mathematics and Information Science, Warsaw University of Technology
ul. Koszykowa 75, Warszawa, Poland
†Applica.ai
ul. Wiślana 8, Warszawa, Poland
E-mail: {anna.wroblewska, tomasz.stanislawek, bartlomiej.prus, lukasz.garncarek}@applica.ai

***Abstract*—In this paper we present our work in progress on building an artificial intelligence system dedicated to tasks regarding the processing of formal documents used in various kinds of business procedures. The main challenge is to build machine learning (ML) models to improve the quality and efficiency of business processes involving image processing, optical character recognition (OCR), text mining and information extraction. In the paper we introduce the research and application field, some common techniques used in this area and our preliminary results and conclusions.**

I. Introduction

NOWADAYS, business processes are not enough time and cost effective, so much effort is made to automate them. Researchers together with business partners are expecting that a vast majority of repetitive human work can be eliminated with the aid of artificial intelligence. Besides human work automation, there are also other advantages to gain: economic leverage of large scale, reduction of a process duration, repetitiveness, analytics of a process, higher specialization in core business, lower maintenance costs [1]

In a typical Robotic Process Automation (RPA) tool human work is automated by observing the user perform a task in the graphical user interface (GUI), and consecutively with the help of artificial intelligence repeating the same task in the GUI. There are a lot of companies that help to automate parts of business processes that fit that model, like 'Blue Prism', 'UiPath', 'Verint', 'WorkFusion' [2][3][4][5].

In our case, we want to focus on dealing with unstructured data (e.g. letters, e-mails, invoices). This is one of the hardest parts to solve because one does not take exactly the same actions to process each document. Examples of such tasks are: finding relevant monetary amounts in invoices, automatically understanding customer complaints, chat bots for call centers, etc.

The main contribution of this paper is to state and organize challenges associated with RPA based on unstructured data. Moreover, we present our baseline machine learning models for processing formal documents and try to challenge them with innovative approach incorporating deep learning methods, e.g. transfer learning using language models, state-of-the-art deep learning architectures, etc.

In the following sections we introduce requirements for RPA, show a general architecture of such a system (section II), and sketch state-of-the-art ML methods that are used for particular parts of our system (section III). Then we show our approach to automation of processing formal documents, describing use cases and preliminary results (section IV). Finally, we conclude our work with future plans and important tips for RPA with ML methods (section V).

II. Robotic Process Automation of Unstructured Data

It is observed that in big companies a lot of communication with clients or business partners takes place through some kind of documents (e.g. letters, e-mails, images, pdfs, text files). Nowadays, the typical solution is to create Business Process (BP) to handle that kind of information channel.

Robotic Process Automation can be defined as a support system which automates human work that involves routine tasks [6]. In our study we solely deal with business processes that handle unstructured data (we call them documents). According to Fung [7] not all of BPs can be automated, so we define criteria and a general architecture of a business process that may adopt RPA mechanism. Processes which meet these requirements could be automated and bring improvements to operational efficienvy of any organization.

Criteria that must be fulfilled by a business process to apply a RPA solution:

1) *High volume* – thousands and thousands of documents are processed every month, involving much workforce in a business process.
2) *Standardization* – a business process must be standardized at some level, i.e. a process must have specifications that are fixed most of the time. For instance, we must know what kind of information should be extracted from a document.

On the other hand, we also define criteria which must be met by a RPA solution:

1) *Simple architecture* – ensures easy integration at various stages of a business process.
2) *High scalability* – ensures ability to cope with a high volume of documents. RPA service should be easy to scale and it should process documents in a matter of milliseconds or seconds rather than minutes.

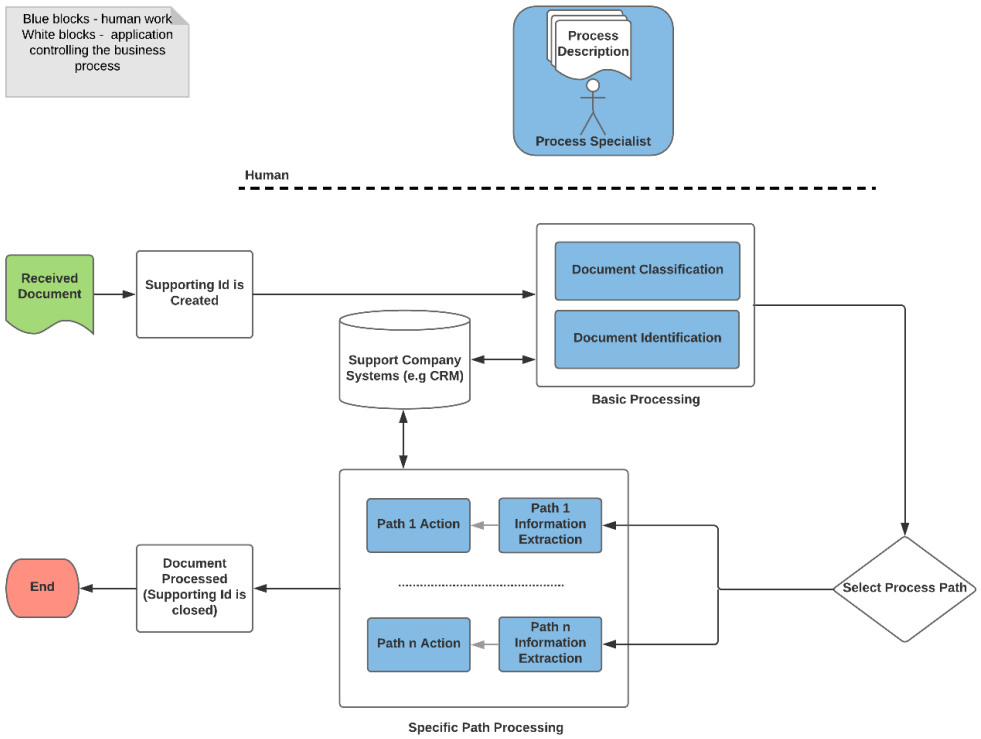

Fig. 1. General Business Process Architecture.

3) *Flexibility* – each model should be implemented as a separable module to make it easy to use.
4) *Quality* – machine learning models must be very accurate in order to ensure profitability from a business perspective.
5) *Security* – ensures appropriate access control to company's data and resources.

A. General Business Process Architecture

According to our criteria, we present a common architecture of a business process (Fig. 1), in which a RPA system can be applied. In the proposed architecture we can distinguish the following blocks:

1) *Document Received* – a document was sent to the process and assigned an ID as a marker. Commonly, companies use queues to store documents for further processing.
2) *Basic Processing* – at this stage we define two main modules: *Document Classification* and *Document Identification*. In the classification task a human specialist must assign a category/categories to the document, based on a taxonomy specified in the Process Description document. Each category has some specific meaning regarding the particular business process, e.g. e-mails with the 'Complaint' category will have a greater priority than e-mails with the 'Payment Confirmations' category. The main goal of the Document Identification module is to link the existing database records from the company's CRM system to each document, e.g. finding a client ID or a contract ID related to this document.
3) *Process Path Selection* – the business process owner defines a set of rules based on the information collected from the *Basic Processing* stage which determines a processing path for the document, e.g. the document category determines an object list that must be extracted in the next step. Those rules are implemented in an application to support the business process.
4) *Specific Path Processing* – when a path is selected, we can process the document based on Process Specification. Each path has two modules: *Information Extraction* and *Action*. It means that first we must extract specific objects from documents and then we should take a specific action related to the collected information. For instance, if we know that a client sent an e-mail with the 'Payment Confirmation' category, we can extract an amount of the payment and afterwards check if the amount is the same as the value retrieved from CRM. If the two amounts agree, we can unblock the user account, and if not, we can send an e-mail to the client. Additionally, we can store all extracted valuable information in the CRM system.
5) *Document Processed* – the final stage of the process. It means that the document is marked as processed and all relevant actions have been taken.

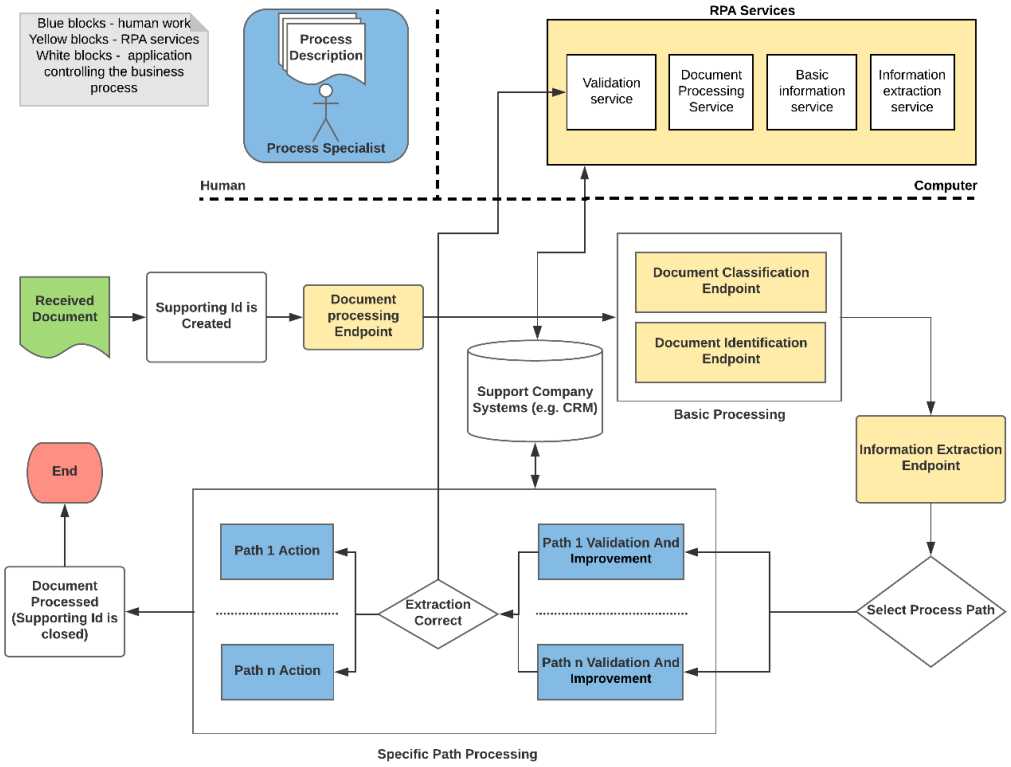

Fig. 2. Business Process Architecture with RPA Services.

B. Automation of Business Process

As we can see in Fig. 1 a lot of work is done manually by company employees in any business process. Most of such tasks can be replaced by machine learning methods such as:

1) *Image processing* – scanning letters, invoices, attachments from emails, etc.
2) *Classification* – labeling documents according to a taxonomy described in Process Specification documents, linking CRM information to documents.
3) *Information extraction* – extracting specific objects from image/text that are needed to process documents.

We offer our own system architecture (Fig. 2), in which most of modules are replaced by services using machine learning methods. That solution reduces the amount of work done by human workforce. To make reading easier blue blocks indicate tasks done by humans and yellow ones indicate tasks done by RPA services.

The workflow of the business process is mostly the same. From the company's perspective this is a big advantage because they do not need to change much of their business process environment. Additionally, our solution assumes that all communication between RPA services and a business process system is based on REST web-services, which is a standard solution these days. The difference is only in three extra blocks, which were added for integration purposes:

1) *Document Processing Service* – a service to pre-process documents for further use in all RPA services (perform OCR on images, process a text by NLP tools, etc.).
2) *Validation And Improvement Path* – at this stage humans check if the data have been correctly classified and extracted. If some models made a mistake it is sent to *Validation Service*.
3) *Validation Service* – a service to ensure high accuracy of models. This service is provided with information from model errors which are diagnosed and used to correct our predictions in the future.

III. Machine Learning Methods

In this chapter we describe some state-of-the-art and classic techniques that are used in tasks similar to these that we need in our RPA solution.

A. Image Processing

The most common and standard methods to improve the quality of images involve image filtering in the spatial or frequency domain. In a spatially filtered image, the value of each output pixel is the weighted sum of neighboring input pixels. The weights are provided by a matrix called the convolution kernel or filter. There are following kinds of filters: texture filtering to smooth an image or to sharpen it, filters that extract edges, morphological filters, etc. [8]. There are environments to do these operations in an easy way: Matlab environment [9] and OpenCV library with API for the most common programming languages [10].

To perform easier tasks there are still common classical methods that need preparing feature extraction from images and then building ML models using these features as an input. These models can be: random forest [11], linear regression, etc.

Currently the most popular ML methods for computer vision are deep learning (DL) neural networks. The latest research shows that DL can outperform even humans in tasks of object class recognition, etc. [12]. For example, to detect and classify particular (previously defined) classes of objects, the most efficient architecture is 'Faster-RCNN' [13].

These DL techniques take advantage of the knowledge gleaned from very big open datasets (e.g. ImageNet [14], COCO [15]) and models built to classify, detect and even segment objects in them (e.g. AlexNet, ResNet, VGG architectures [16]). This technique is called transfer learning.

B. Classification

The problem of classification can be defined as a prediction of a discrete class or classes y given a vector of features $\mathbf{x} = (x_1, \ldots, x_k)$. The domain of y is a set of possible classes depending on a problem. In machine learning area a whole range of methods may be used to tackle this problem.

In recent years, linear models (e.g. logistic regression or support vector machines) have seen an increase in usage, mainly when dealing with big data. This is thanks to their ability to scale very well when using appropriate optimization techniques (e.g. stochastic gradient descent) and possibility of on-line learning [17][18][19]. However, we can observe a great improvement in classification accuracy when using deep neural networks[20][21] or XGBoost method [22].

1) Logistic Regression: Logistic regression (LR) is in its basic form a linear binary model. It tries to estimate the probability of a positive class given the input observations. Formally, it can be expressed by the following equation

$$P(Y = 1|x) = \sigma(\mathbf{x^T w})$$

where $\mathbf{x}$ is input data, $\mathbf{w}$ is a vector of model weights (or coefficients) and σ is sigmoid function which transforms the value of $\mathbf{x^T w}$ from $(-\infty, +\infty)$ into $[0, 1]$.

To find parameters $\mathbf{w}$ LR uses *log loss* function during optimization, defined as

$$L(y, \mathbf{wx}) = \log(1 + e^{-y\mathbf{wx}}).$$

2) Support Vector Machines: Support Vector Machines (SVMs) were proposed by Cortes and Vapnik in 1995 [23] and later employed for text classification by Joachims in [24]. A SVM model attempts to find a hyperplane separating positive and negative training examples with additional condition of maximizing the margin (i.e. the distance of training examples to the hyperplane). In case of linearly separable classes, to this end one needs to use only a small fraction of training examples called *support vectors* – hence the name.

Assume we are given a sequence of data points $\mathbf{x}_i \in \mathbb{R}^d$ together with their class labels $y_i \in \{\pm 1\}$. The SVM assigns to each point a score by the formula

$$A(\mathbf{x}) = \mathbf{x}^T\mathbf{w} + b,$$

where $\mathbf{w} \in \mathbb{R}^d$, and $b \in \mathbb{R}$ are the trainable parameters of the model.

The optimization objective is defined with help of the *hinge loss* function which penalizes not only misclassification, but also, to a lesser extent, predictions with low confidence. The loss to be minimized is defined as

$$\frac{C}{N}\sum_{i=1}^{N}\max\{0, 1 - (y_i A(\mathbf{x}_i))\} + \frac{1}{2}\|\mathbf{w}\|^2,$$

where C is a regularization constant.

3) Convolutional Neural Networks (CNN) : Convolutional neural networks improve their efficiency not only in image processing techniques, as they were initially used, but also in text classification. Overall, CNNs are extremely effective in mining semantic clues in contextual windows [25]. The latest research underlines their great achievements in text field [26].

Assume we have an embedding matrix $\mathbf{W} \in \mathbf{R}^{nxd}$ which is a $\mathbf{d}$-dimensional representation of embeddings of $\mathbf{n}$ words (or letters). This input layer can be pre-trained or initialized randomly. Convolutions are performed on consecutive parts of this input embedding layer (on a consecutive row of input embedding matrix).

The convolutional layer involves a filter $\mathbf{k} \in \mathbf{R}_{hxd}$ which is applied to a window of $\mathbf{h}$ words to produce a new feature and a non-linear activation function, for example, ReLU. The filter $\mathbf{k}$ is applied to all possible windows using the same weights to create the feature map. A set of convolutional filters of different widths slides over the entire embedding matrix and they extract patterns of n-grams. Convolutions are usually followed by max-pooling, which subsamples the input to provide fixed-length output and keep the most salient n-gram features across the whole input [25].

C. Information extraction

In the information extraction task we try to find a set of objects (e.g. Person, Localization, Amount, etc.) in unstructured text data relevant to a specific domain [27]. The most common approach is to represent a text as a sequence of tokens (words or characters) denoted as vector $\mathbf{x} = (x_1, \ldots, x_n)$, and the corresponding vector $\mathbf{y} = (y_1, \ldots, y_n)$, representing output class labels (each value $\mathbf{x_i}$ has its own output class $\mathbf{y_i}$ with domain $\mathbf{D}$), where $\mathbf{i}$ is a position of a token in the text. As a baseline we have used Conditional Random Fields (CRF) but this is being replaced by deep learning architecture involving state-of-the-art recurrent neural networks.

1) Conditional Random Fields: Conditional Random Fields (CRF) was proposed in 2001 by Lafferty [28] and for a long time it was considered one of the most accurate methods of sequence labeling.

CRF is a probabilistic graphical model that defines a single log-linear distribution over label sequences $\mathbf{y} = (y_1, \ldots, y_n)$,

given a particular observation sequence $\mathbf{x} = (x_1, \ldots, x_n)$,. Lafferty [28] defines that probability as a normalized product of potential functions

$$\mathbf{P}(\mathbf{y}|\mathbf{x}, \lambda) = \frac{1}{Z(\mathbf{x})} exp \sum_{i=1}^{N} \sum_{j} \lambda_j f_j(y_{i-1}), y_i, \mathbf{x}, i)$$

where $Z(\mathbf{x})$ is a normalization factor, λ_j is a parameter to be estimated and $f_j(y_{i-1}), y_i, \mathbf{x}, i)$ is a feature function that expresses some characteristic of the empirical distribution that we wish to hold in the model distribution [29].

2) Recurrent Neural Networks: Although recently challenged by *temporal convolutional networks*, *recurrent neural networks* (RNNs) are the traditional neural architectures used for sequence labeling [30][31][26]. Given a sequence of vectors

$$\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$$

of arbitrary length n, a RNN model produces an output sequence $(\mathbf{y}_i)$ of the same length, which in general can be described by the following recurrent formula

$$\mathbf{y}_i = \begin{cases} F_0(\mathbf{x}_0) & \text{if } i = 0, \\ F(\mathbf{x}_i, \mathbf{y}_{i-1}) & \text{otherwise,} \end{cases}$$

where F_0 and F are functions of a prescribed form with trainable parameters. The two most common choices for these functions lead to *long short-term memory* (LSTM) and *gated recurrent unit* (GRU) modules. We refrain from presenting the exact formulas, as they are a bit complex and out of scope of this paper.

What is important though is the fact that suitably defined recurrent modules, such as LSTMs or GRUs, exhibit a behavior that can be described as possessing *memory*. Namely, the model may learn to store information encountered in earlier steps in the output vectors $\mathbf{y}_i$, and use it later, when needed.

IV. Application

A. Use-case Description

In this section we describe a problem of automating BP from a debt collector company. The company receives more than 150 000 documents each month sent by courts and bailiffs as letters via the post office. Such an enormous information stream is hard to handle by human workers alone. It is also important to process each document immediately because some debt collection cases have limited time to react to. Below we present the results of our baseline solution, which is currently being improved by employing new innovative techniques.

First, each document is scanned and company workers assign the main category to it: either 'Bailiff document' or 'Court document'. Next, the process looks very similar to the one presented in Fig. 2:

1) Document Received – supporting ID is created.
2) Document Preprocessing – documents are preprocessed and OCR-ed.
3) Basic Processing – documents are classified and then Case and Client are identified and matched to the existing client's database.
4) Information Extraction – relevant information is extracted from the document, based on its category.
5) Extraction Validation – data is verified and sent to a validation service if not correct.
6) Action – a specific action is taken based on the extracted information.
7) Document Processed – supporting ID is closed.

One of the many challenges in the process is to interpret and understand Polish language texts that are retrieved with OCR. Thus we make an effort to process document scans and obtain texts in as good quality as possible to improve further processing. Having texts after OCR we try to recognize important entities (e.g. persons, addresses, ID numbers of documents, dates), extract monetary amounts, etc. These tasks are very challenging regarding that NLP tools for Polish are not of as high a quality as NLP for English, mostly because Polish is a highly inflected language with a complex grammar.

B. Approach and Results

1) Image Processing: In order to improve OCR results, we need to have high quality document images. Thus we need image processing methods to remove noisy background from images, detect important parts of images that should be treated differently from other parts of scans, e.g. tables and multiple columned texts.

To improve document image quality, we use morphological filtering and correcting text line skewness with Hough transform techniques [8]. Then to detect table headers, which are particularly difficult to OCR mechanism, we classify them as having particular grayish background or inverse colors of the background and text using image feature extraction and random forest models. Then we can distinguish those different backgrounds and take proper further steps.

At the same time we detect multicolumn texts, e.g. tables with monetary amounts calculations using DL methods. We adopt transfer learning – 'VGG-19' learned on ImageNet dataset and 'Faster-RCNN' architecture. Then we train them on our small dataset of about 400 documents.

With these technique we achieve preliminary results on a test set of about 300 documents to identify tables with about 91% of precision and 87% of recall, even without preprocessing steps.

The next process is to select important regions of document images to OCR them with adjusted parameters for these particular tasks, e.g. OCR of tables. Additionally we try to improve OCR results in order to help information extraction methods and extract columns and rows of tables, e.g. to extract proper monetary amounts and their classes. Here we use heuristics that have a table and column area in the images and OCR results (that is bounding boxes and recognized text tokens). Thus we have prepared text documents of as high a quality as possible.

2) Document classification: The first and the most important task during the whole process is to classify incoming documents into appropriate categories. There are two possible sources of documents: either bailiff or court documents. Bailiff documents are classified into one of 18 categories and court documents are classified into one of 25 categories. This is a single-label multi-class categorization problem. The task is really important because further processing depends on the classification results.

The training set for court documents contains 26639 documents and for bailiff documents – 25339. Additionally, we have independently prepared test sets and they have 2501 and 2691 documents for court and bailiff respectively.

The solution presented here uses SVM classifier to complete this task. Throughout years SMVs were shown to achieve good performance in text classification tasks and are a good starting point [32]. This is why we chose it as a baseline method to compare our future results with. Yet our current experiments show that SVMs can be outperformed using more sophisticated methods, even with less training data.

In the beginning we perform classic document preprocessing (lowercasing, removing diacritics and stopwords, tokenizing, etc.). We take unigrams and bigrams as features, weighted with TF-IDF algorithm. Features so prepared are passed to the SVM classifier. For SVM training we use popular LIBLINEAR library [18].

With such a baseline solution, we were able to achieve 92.08% accuracy for bailiff documents and 92.36% accuracy for court documents on the test sets. Moreover, we have verified our predictions on real data and reached about 91% on court and about 93% accuracy on bailiff documents.

As mentioned above, document classification is an essential part of the whole process, as it directly impacts further processing. That is why we put a lot of effort into improving its quality. Currently, we are experimenting with different deep learning methods for classification. We tried convolutional neural networks [21] and temporal convolutional networks [26]. We were able to improve upon SVM performance by about 0.5-1.0 pp.

Moreover, apart from deep learning efforts, we experiment with transfer learning [33]. To do this we have built a language model based on a very large data set and this model is then used as an input to a classifier. Preliminary results show that it is a very promising approach. Not only can we obtain better accuracy, but also do it with much less data.

3) Document identification: After classifying a document, we need to assign it to an existing case using the information from our client's database. This process has four steps.

The first one is information extraction. We extract such objects as court reference numbers, personal identification numbers, addresses, debt amounts etc. We typically either find only a subset of information that could be extracted from the document or we end up with noisy information that render extracted objects unhelpful in further steps. This step ends with an incomplete set of textual features, some of which may contain errors.

The second step is to search the client's database for cases which partially match the extracted information. We call these cases 'candidates'.

After finding the candidates, we once again compare each candidate's database entry with the extracted information, and use compatibility measures to produce numerical features, thus describing each candidate by a real-valued vector.

In the last step we pass the feature vectors through a machine learning model (our baseline solution uses logistic regression) that produces a probability that a given candidate is the correct match for the document in question, and return the candidate with the highest probability, provided that it exceeds some fixed threshold.

With our benchmark solution we are able to achieve 99.84% precision and 76.02% recall. Obtained recall results are still below our expectations but we are optimistic that using a pre-trained language model to improve quality of the text will result in a much better performance of document identification.

4) Information extraction: Apart from the information extracted in the process of document identification (section IV-B3), we are currently dealing with the problem of extracting monetary amounts of different kinds, which appear directly in the document. There are a few groups of amounts we are working on and each of them requires a different architecture.

In general, this problem can be modeled as follows. Given a document $\mathbf{x}$ represented as a sequence of characters

$$\mathbf{x} = (x_1, \ldots, x_n),$$

we may infer a sequence

$$\mathbf{p} = (p_1, \ldots, p_n),$$

where $p_i \in [0, 1]$ is the probability that the character x_i belongs to the amount to be extracted. Having produced such a sequence of probabilities, we extract maximal contiguous subsequences of the form $(x_k, x_{k+1}, \ldots, x_\ell)$ such that $p_i > T$ for all $k \leq i \leq \ell$ and a given threshold value T. These excerpts then undergo validation and normalization procedures, mainly due to the noise in OCR-ed texts, but also to eliminate some obvious errors such as partial extraction.

A single model can be also used to extract multiple related objects. In this case we just use $p_i \in [0, 1]^k$, where k is the number of extracted objects.

To produce the score sequence we use state-of-the-art character-level recurrent neural networks (section III-C2). Architectures that we employ consist of an embedding layer, possibly followed by some convolutional layers, a single-layer GRU module, and some densely connected layers producing the final scores. Intuitively, using convolutional layers means that we try to look not only at single characters, but also syllables or even words – depending on the parameters of the convolutional module. We are also experimenting with applying a pre-trained character based language model instead of a classical embedding layer.

TABLE I
PERFORMANCE OF MONETARY AMOUNT EXTRACTORS

Task	Precision	Recall	F1 Score	# of test documents	# of test objects	# of train documents	# of train objects
Bailiff costs	83.07%	86.40%	84.70%	3316	10729	12500	40637
Court costs	77.46%	97.77%	86.44%	20557	6131	126665	38058

We did not found a significant difference when using LSTM instead of GRU, and increasing the number of recurrent layers beyond 1 leads to unacceptable prediction times.

There is a technical issue here, related to the potentially unlimited length of the documents. Unlike some other applications, information extraction requires us to process the whole document. We solve the problem by cutting the text into overlapping fragments of the form

$$(x_{k(\ell-s)}, x_{k(\ell-s)+1}, \ldots, x_{k(\ell-s)+\ell-1})$$

of prescribed length ℓ and overlap size s. We pass these fragments to the model separately, and suitably interpolate the resulting probability sequences on overlapping fragments, once again obtaining a single sequence of probabilities.

Now, let us get back to the problem at hand. We are working on two groups of amounts: attorney representation costs, and costs related to various actions performed by the bailiff in search of debtor's assets. Table I presents the performance of extractors trained with 80%-20% train-test split. It is worth noting that our current data sets were obtained in a partially automated manner. Thus they need to be verified by linguists to achieve results of a higher quality.

V. CONCLUSIONS

In this paper we described general business processes for documents flow and current approaches to automating them; we also presented our baseline solutions for a specific business process. Our particular use-case is dedicated to a debt collecting process. We described possible applications of ML methods to improve the efficacy of these processes. This is a significant benchmark for all future work in this field.

We are devoted to an ongoing aim to improve our existing models. Each part of our RPA solution is constantly being upgraded. Much emphasis is given to further research into deep learning methods and into transfer learning, using previously trained language models.

We believe that the core value of our solutions in the near future will be the reduction of training data size required to achieve performance comparable to the currently used state-of-the-art methods. Corporate clients will need to provide significantly less data and the whole process of implementing automation will be a lot more efficient (lower cost, quicker integration, robust, etc.).

Besides, we plan to experiment with automation of *Action* (Fig. 1) stage. We want to be able to automatically take actions for more complicated use-cases, e.g. responding to a client via e-mails based on collected information from *Basic Processing* and *Information Extraction* modules.

ACKNOWLEDGMENT

This paper provides a description of a project currently conducted at Applica.ai, an AI business application firm. As such, the paper has a large number of contributors, including, but not limited to, Adam Dancewicz and Piotr Surma, our managing directors, and Paweł Dyda, Przemysław Lipka, Arakdiusz Trzepacz, Szymon Sikorski, Paweł Józiak and a team of linguists.

The authors would like to acknowledge the support the Applica.ai project has received as being co-financed by the European Regional Development Fund (POIR.01.01.01-00-0144/17-00).

REFERENCES

[1] M. Kukreja and A. singh Nervaiya, "Study of robotic process automation (rpa)," *International Journal on Recent and innovation trends in computing and communication*, vol. 6, pp. 434–437, 2016.

[2] Robotic process automation, "Robotic process automation — Wikipedia, the free encyclopedia," 2018, [Online; accessed 7-June-2018]. [Online]. Available: https://en.wikipedia.org/wiki/Robotic_process_automation

[3] Capgemini Consulting, "Robotic Process Automation – Robots conquer business processes in back offices," 2016, accessed: 2018-06-07. [Online]. Available: https://www.capgemini.com/consulting-de/wp-content/uploads/sites/32/2017/08/robotic-process-automation-study.pdf

[4] PWC, "Rethinking retail: Artificial Intelligence and Robotic Process Automation," 2017, accessed: 2018-06-07. [Online]. Available: https://www.pwc.be/en/documents/20171123-rethinking-retail-artificial-intelligence-and-robotic-process-automation.pdf

[5] David Schatsky and Craig Muraskin and Kaushik Iyengar, "Robotic process automation A path to the cognitive enterprise," 2016, accessed: 2018-06-07. [Online]. Available: https://www2.deloitte.com/content/dam/insights/us/articles/3451_Signals_Robotic-process-automation/DUP_Signals_Robotic-process-automation.pdf

[6] S. Aguirre and A. Rodriguez, "Automation of a business process using robotic process automation (rpa): A case study," in *Workshop on Engineering Applications*. Springer, 2017, pp. 65–71.

[7] H. P. Fung, "Criteria, use cases and effects of information technology process automation (itpa)," 07 2014.

[8] W. K. Pratt, *Digital Image Processing: PIKS Inside*, 3rd ed. New York, NY, USA: John Wiley & Sons, Inc., 2001.

[9] Matlab official website, "Matlab," 2018, [Online; accessed 7-June-2018]. [Online]. Available: https://www.mathworks.com/products/matlab.html

[10] OpenCV official website, "Opencv," 2018, [Online; accessed 7-June-2018]. [Online]. Available: https://opencv.org

[11] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015. [Online]. Available: http://arxiv.org/abs/1506.01497

[12] J. Howard and S. Ruder, "Fine-tuned language models for text classification," *CoRR*, vol. abs/1801.06146, 2018. [Online]. Available: http://arxiv.org/abs/1801.06146

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[14] ImageNet dataset official website, "Imagenet," 2018, [Online; accessed 7-June-2018]. [Online]. Available: http://www.image-net.org

[15] COCO dataset official website, "Coco," 2018, [Online; accessed 7-June-2018]. [Online]. Available: http://cocodataset.org

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[17] X. Li, "Classification with large sparse datasets: Convergence analysis and scalable algorithms," 2017.

[18] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Jun. 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=1390681.1442794

[19] K.-c. Lee, B. Orten, A. Dasdan, and W. Li, "Estimating conversion rate in display advertising from past performance data," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '12. New York, NY, USA: ACM, 2012, pp. 768–776. [Online]. Available: http://doi.acm.org/10.1145/2339530.2339651

[20] H. T. Le, C. Cerisara, and A. Denis, "Do convolutional networks need to be deep for text classification ?" *CoRR*, vol. abs/1707.04108, 2017. [Online]. Available: http://arxiv.org/abs/1707.04108

[21] Y. Kim, "Convolutional neural networks for sentence classification," *CoRR*, vol. abs/1408.5882, 2014. [Online]. Available: http://arxiv.org/abs/1408.5882

[22] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: ACM, 2016, pp. 785–794. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939785

[23] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep 1995. [Online]. Available: https://doi.org/10.1007/BF00994018

[24] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *Machine Learning: ECML-98*, C. Nédellec and C. Rouveirol, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 137–142.

[25] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *CoRR*, vol. abs/1708.02709, 2017. [Online]. Available: http://arxiv.org/abs/1708.02709

[26] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *CoRR*, vol. abs/1803.01271, 2018. [Online]. Available: http://arxiv.org/abs/1803.01271

[27] J. Piskorski and R. Yangarber, *Information Extraction: Past, Present and Future*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 23–49. [Online]. Available: https://doi.org/10.1007/978-3-642-28569-1_2

[28] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 282–289. [Online]. Available: http://dl.acm.org/citation.cfm?id=645530.655813

[29] H. M. Wallach, "Conditional random fields: An introduction," *Technical Reports (CIS)*, p. 22, 2004.

[30] J. P. C. Chiu and E. Nichols, "Named entity recognition with bidirectional lstm-cnns," *CoRR*, vol. abs/1511.08308, 2015. [Online]. Available: http://arxiv.org/abs/1511.08308

[31] Z. Yang, R. Salakhutdinov, and W. W. Cohen, "Multi-task cross-lingual sequence tagging from scratch," *CoRR*, vol. abs/1603.06270, 2016. [Online]. Available: http://arxiv.org/abs/1603.06270

[32] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, Mar. 2002. [Online]. Available: http://doi.acm.org/10.1145/505282.505283

[33] Z. Yang, R. Salakhutdinov, and W. W. Cohen, "Transfer learning for sequence tagging with hierarchical recurrent networks," *CoRR*, vol. abs/1703.06345, 2017. [Online]. Available: http://arxiv.org/abs/1703.06345