Information Extraction from Invoices

Ahmed Hamdi[1](✉), Elodie Carel[2], Aurélie Joseph[2], Mickael Coustaty[1], and Antoine Doucet[1]

[1] Université de La Rochelle, L3i, Avenue Michel Crépeau, 17042 La Rochelle, France
{ahmed.hamdi,mickael.coustaty,antoine.doucet}@univ-lr.fr
[2] Yooz, 1 Rue Fleming, 17000 La Rochelle, France
{elodie.carel,aurelie.joseph}@getyooz.com

Abstract. The present paper 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. Common information extraction systems are model-driven, using heuristics and lists of trigger words curated by domain experts. Their performances are generally high on documents they have been trained for but processing new templates often requires new manual annotations, which is tedious and time-consuming to produce. Recent works on deep learning applied to business documents claimed a gain in terms of time and performance. While these systems do not need manual curation, they nevertheless require a large amount of data to achieve good results. In this paper, we present a series of experiments using neural networks approaches to study the trade-off between data requirements and performance in the extraction of information from key fields of invoices (such as dates, document numbers, types, amounts...). The main contribution of this paper 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, that are costly and impractical to produce in real-world applications.

Keywords: Invoices · Data extraction · Features · Neural networks

1 Introduction

Administrative documents (such as invoices, forms, payslips...) are very common in our daily life, and are required for many administrative procedures. Among those documents, invoices claim specific attention as they are related to the financial part of our activities, many of them are received every day, they generally need to be paid shortly, and any error is not acceptable. Most of those procedures are nowadays dematerialized and performed automatically. In this aim, information extraction systems extract key fields from documents such as identifiers and types, amounts, dates and so on. This automatic process of invoices has been formalized by Poulain d'Andecy *et al.* [18] and requires some specific features:

J. Lladós et al. (Eds.): ICDAR 2021, LNCS 12822, pp. 699–714, 2021.
https://doi.org/10.1007/978-3-030-86331-9_45

- handling the variability of layouts,
- minimizing the end-user effort,
- training and quickly adapt to new languages and new contexts.

Even if a formal definition has been proposed in the literature, current approaches rely on heuristics which describe spatial relationships between the data to be extracted and a list of trigger words through the use of dictionaries. Designing heuristics-based models is time-consuming since it requires human expertise. Furthermore, such models are dependent on the language and the templates they have been trained for, which requires annotating a large number of documents and labeling every piece of data from which to extract information. Thus, even once a system has reached a good level of performance, it is very tedious to integrate new languages and new templates. We can for instance cite recent data analysis approaches (CloudScan [22], Sypht [9]) which consider information extraction as a classification problem. Based on its context, each word is either assigned to a specific class. Their results are very competitive but these systems require huge volumes of data to get good results.

The problem of extracting some specific entities from textual documents has been studied by the natural language processing field and is known as Named Entity Recognition (NER). NER is a subtask of information extraction that aims to find and mark up real word entities from unstructured text and then to categorize them into a set of predefined classes. Most NER tagsets define three classes to categorize named entities: persons, locations and organizations [16]. Taking advantage of the development of neural-based methods, the performance of NER systems has kept increasing since 2011 [3].

In this paper, we propose to adapt and compare two deep learning approaches to extract key fields from invoices which comprise regular invoices, receipts, purchase orders, delivery forms, accounts' statements and payslips. Both methods are language-independent and process invoices regardless of their templates. In the first approach, we formulate the hypothesis that key fields related to document analysis can extend name entity categories and be extracted in the same way. It is based on annotated texts where all the target fields are extracted and labeled into a set of predefined classes. The system is based on the context in which the word appears in the document and additional features that encode the spatial position of the word in the documents.

The second approach converts each word of the invoice into a vector of features which will be semantically categorized in a second stage. To reduce the annotation effort, the corpus comes from a real-life industrial workflow and the annotation is semi-supervised. The corpus has been tagged with an existing information extraction system and manually checked by experts. Compared to the state of the art, our experiments show that by adequately selecting part of the data, we can train competitive systems by using a significantly smaller amount of training data compared to the ones used in state-of-the-art approaches.

This paper first introduces prior works on information extraction from invoices (Sect. 2). We then describe the data used to assess our work (Sect. 3). Our two approaches based on named entity recognition and document analysis are detailed in Sect. 4. The experiments are described in Sect. 5 to compare our models to state-of-the-art methods, before discussions and future work (Sect. 6).

2 Related Work

Data extraction from administrative documents can be seen as a sequence labeling task. The text is extracted from documents using an OCR engine. Then, each extracted token is to be assigned a label. Early sequence labeling methods are rule-based [5,8]. These rules were defined by humans and based on trigger words, regular expressions and linguistic descriptors. For instance, amounts can be found in the neighborhood of words such as "total", "VAT", etc.

More recent sequence labeling methods are based on neural networks and have been shown to outperform rule-based algorithms [2,6,11,13]. Devlin *et al.* [7] introduced the bidirectional encoder representations from transformers (BERT), which have become popular among researchers. BERT is pre-trained on large data and known for its adaptability to new domains and tasks through fine-tuning. Aspects such as the easiness to fine-tune and high performance, have made the use of BERT widespread in many sequence labeling tasks.

However, administrative documents such as invoices and receipts contain only a few word sequences in natural language. Their content is structured and models can be designed to locate data. For this reason, many other systems combined the textual context with the structural one. Candidates for each field are generally selected and ranked based on a confidence value, and the system returns the best one. Rusiñol *et al.* [21] and their extended work, Poulain d'Andecy *et al.* [18], capture the context around data to be extracted in a star graph. The graph encodes pairwise relationships between the target and the rest of the words of the document which are weighted based on an adaptation of the TF-IDF metric. The system is updated incrementally. The more the system receives documents from a given supplier, the more the graph will be efficient. Once they have been designed for a domain, model-driven systems are often efficient in terms of performance. But processing a real-world document flow is challenging because its input is constantly evolving over time. Hence, extraction systems should cope with unseen templates. Eventually, they should also be able to cope with multiple languages. Designing heuristics requires an expert in the domain. In the production phase, this step has to be done for each new model which cannot be processed properly by the extraction system. It is time-consuming and error-prone. Updating the extraction system implies very tedious work to check the regressions in order to keep high performance. Some systems, such as that of Poulain d'Andecy *et al.* [18], try to limit user intervention by labeling only the target data. However, this process requires a pre-classification step to be able to recognize the supplier.

Recent works proposed deep learning based approaches to solve the extraction problem. Palm *et al.* [17] presented CloudScan, a commercial solution by Tradeshift. They train a recurrent neural network (RNN) model over 300k invoices to recognize eight key fields. This system requires no configuration and does not rely on models. Instead, it considers tokens and encodes a set of contextual features of different kinds: textual, spatial, related to the format, etc. They decided to apply a left-to-right order. However, invoices are often written in both vertical and horizontal directions. Other works have been inspired by CloudScan. In their work, they compare their system to alternative extraction systems and they claim an absolute accuracy gain of 20% across compared fields. Lohani *et al.* [12] built a system based on graph convolutional networks to extract 27 entities of interest from invoices. Their system learns structural and semantic features for each entity and then uses surrounding information to extract them. The evaluation of the system showed good performances and achieves an overall F1-score of 0.93. In the same context, Zhao *et al.* [26] proposed the cutie (Convolutional Universal Text Information Extractor) model. It is based on spatial information to extract key text fields. cutie converts the documents to gridded texts using positional mappings and uses a convolutional neural network (CNN). The proposed model concatenates the CNN network with a word embedding layer in order to simultaneously look into spatial and contextual information. This method allowed them to reach state-of-the-art results on key information extraction.

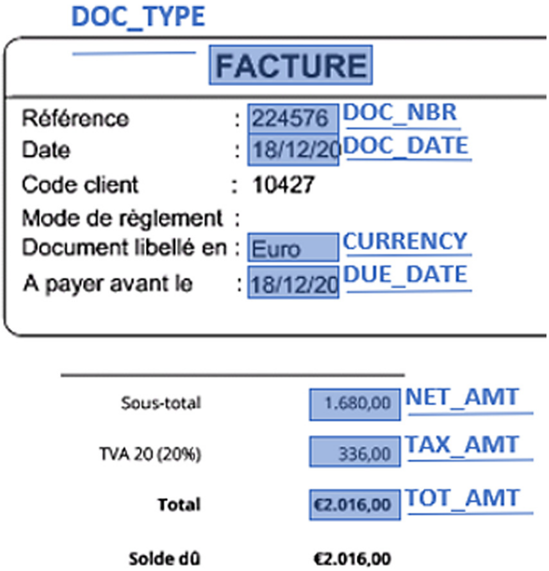
In this work, similarly to [11,13], we combine sequence labeling methods. However, we add neural network layers to our systems so as to encode engineered textual and spatial features. One of these methods is based on named entity recognition using the Transformer architecture [24] and bert [7] that, to our knowledge, have not been reported in previous research on the specific tasks of processing administrative documents. The second method is based on word classification that, unlike previous works, do not require neither pre- nor post-processing to achieve satisfactory results.

3 Dataset

Datasets from business documents are usually not publicly available due to privacy issues. Previous systems such as CloudScan [22] and Sypht [9] use their own proprietary datasets. In the same line, this work is based on a private industrial dataset composed of French and English invoices coming from a real document workflow provided by customers. The dataset covers 10 types of invoices (orders, quotations, invoice notes, statements...) with diverse templates.

The dataset has been annotated in a semi-automatic way. A first list of fields was extracted from the system currently in use, and finally checked and completed by an expert. The main advantage of this process is its ability to get a large volume of documents. However, even if most of the returned values have been checked, we should take into account a part of noise which is expert-dependent. In other words, some fields may be missed by the system or by the expert, as for instance redundant information in a document are typically annotated only once.

The dataset includes two databases which respectively comprise 19,775 and 134,272 invoices. We refer to them by database-20k and database-100k. 8 key fields have been extracted and annotated from these invoices (cf. Fig. 1).



FACTURE

Référence : 224576
Date : 18/12/20
Code client : 10427
Mode de règlement :
Document libellé en : Euro
A payer avant le : 18/12/20

Sous-total	1.680,00
TVA 20 (20%)	336,00
Total	€2.016,00
Solde dû	€2.016,00

DOC_TYPE
DOC_NBR
DOC_DATE
CURRENCY
DUE_DATE
NET_AMT
TAX_AMT
TOT_AMT

Fig. 1. Part of an invoice from our dataset. The blue text is the label of the field (blue boxes) (Color figure online)

Table 1 provides statistics on the databases. Each one was split into 70% for training and 30% for validation.

Table 1. Statistical description of the in-house dataset

Fields	Database-20k	Database-100k
DOC_TYPE	14,301	105,753
DOC_NBR	18,131	131,133
DOC_DATE	17,739	130,418
DUE_DATE	9,554	68,718
NET_AMT	7,323	63,064
TAX_AMT	12,509	91,216
TOT_AMT	15,505	113,372
Currency	12,111	87,965

As mentioned above, this dataset can be noisy, and we therefore decided to evaluate our methods on a manually checked subset. Thus, all the following experiments have been evaluated on a clean set of $4k$ French and English documents, that are not part of the training or validation datasets but come from the same workflow. They are similar in terms of content (i.e. invoices, multi-templates) and have the same ratio of key fields.

4 Methodology

As we mentioned in the introduction, we define and compare two different methods on information extraction that are generic and language-independent: a NER-based method and a word classification-based one (henceforth, we respectively denote them NER-based and class-based). To the best of our knowledge, no research study has adapted NER systems for invoices so far. The NER-based evaluates the ability of NLP mainstream approaches to extract fields from invoices by fine-tuning BERT to this specific task. BERT can capture the context from a large neighborhood of words. The class-based method adapted the features of CloudScan [17] in order to fit our constraints and proposed some extra features to automatically extracts the features, with no preprocessing step nor dictionary lookup. Thus, our methods can easily be adapted to any type of administrative documents. These features significantly reduced the processing of the class-based method on the one hand and allow BERT to deal with challenges related to semi-structured documents on the other hand. Both systems assign a class to a sequence of words. The classes are the key fields to be extracted. We assign the class "undefined" to each word which does not correspond to a key field. Both can achieve good performance with a small volume of training data compared to the state of the art. Each word of the sequence to be labeled is enriched with a set of features that encode contextual and spatial information of the words. We therefore extract such features prior to data labeling. The same features are used for both methods.

4.1 Feature Extraction

Previous research showed that spatial information is relevant to extract key fields from invoices [26]. We therefore attempt to combine classic semantic features to spatial ones, defined as follows.

- Textual features: refer to all the words that can define the context of the word to be labeled w_i in semi-structured documents. These words include the framing words of w_i such as the left and right words as well as the bottom and the top words. These features also include the closest words in the upper and lower lines.
- Layout features: they encode the position of the word in the document, block and line as well as its coordinates (left, top, right, bottom). These features additionally encode page descriptors such as the page's width, height and margin.
- Pattern features: each input word is represented by a normalised pattern built by substituting each character with a normalized one such as C for upper-cased letters, c for lower-cased ones and d for digits. Items such as emails and URLs are also normalised as <EMAIL> and <URL>.
- Logical features: these features have Boolean values indicating whether the word is a title (i.e. if the word is written in a larger font than other words) or a term/result of a mathematical operation (sum, product, factor) using trigrams of numbers vertically or horizontally aligned.

4.2 Data Extraction Using the NER-Based Method

The first contribution of this paper relies on the fine-tuning of BERT [7] to extract relevant information from invoices. The reason for using the BERT model is not only because it is easy to fine-tune, but it has also proved to be one of the most performing technologies in multiple tasks [4,19]. Nonetheless, despite the major impact of BERT, we aim in this paper to evaluate the ability of this model to deal with structured texts as with administrative documents.

BERT consists of stacked encoders/decoders. Each encoder takes as input the output of the previous encoder (except the first which takes as input the embeddings of the input words). According to the task, the output of BERT is a probability distribution that allows predicting the most probable output element. In order to obtain the best possible performance, we adapted this architecture to use both BERT word embeddings and our proposed features. At the input layer of the first encoder, we concatenate the word embedding vector with a fixed-size vector for features in order to combine word-level information with contextual information. The size of the word embedding vector is 572 (numerical values) for which we first concatenate another embedding vector that corresponds to the average embedding of the contextual features. The obtained vector of size 1,144 is then concatenated with a vector containing the logical features (Boolean) and the spatial features (numerical). The final vector size is 1,155.

As an embedding model, we rely on the large version of the pre-trained CamemBERT [14] model. For tokenization, we use CamemBERT's built-in tokenizer, which splits text based on white-spaces before applying a Byte Pair Encoding (BPE), based on WordPiece tokenization [25]. BPE can split words into character n-grams to represent recognized sub-words units. BPE allows managing words with OCR errors, for instance, 'in4voicem' becomes 'in', '##4', '##voi', '##ce', '##m'. This word fragment can usually handle out of vocabulary words and those with OCR errors, and still generates one vector per word.

At this point, the feature-level representation vector is concatenated with the word embedding vector to feed the BERT model. The output vectors of BERT are then used as inputs to the CRF top layer to jointly decode the best label sequence. To alleviate OCR errors, we add a stack of two transformer blocks (cf. Fig. 2) as recommended in [1], which should contribute to a more enriched representation of words and sub-words from long-range contexts.

The system converts the input sequences of words into sequences of fixed-size vectors (x_1,x_2,...,x_n), *i.e.* the word-embeddings part is concatenated to the feature embedding, and returns another sequence of vectors (h_1,h_2,...,h_n) that represents named entity labels at every step of the input. In our context, we aim to assign a sequence of labels for a given sequence of words. Each word gets a pre-defined label (e.g. DOCNBR for document number, DATE for dates, AMT for amounts ...) or O for words that are not to be extracted. According to this system, the example sentence[1] "*Invoice No. 12345 from 2014/10/31 for an amount of 525 euros.*" should be labeled as follows: "*DOCTYPE O DOCNBR O DATE O O O O AMT CURRENCY*".

[1] All the examples/images used in this paper are fake for confidentiality reasons.

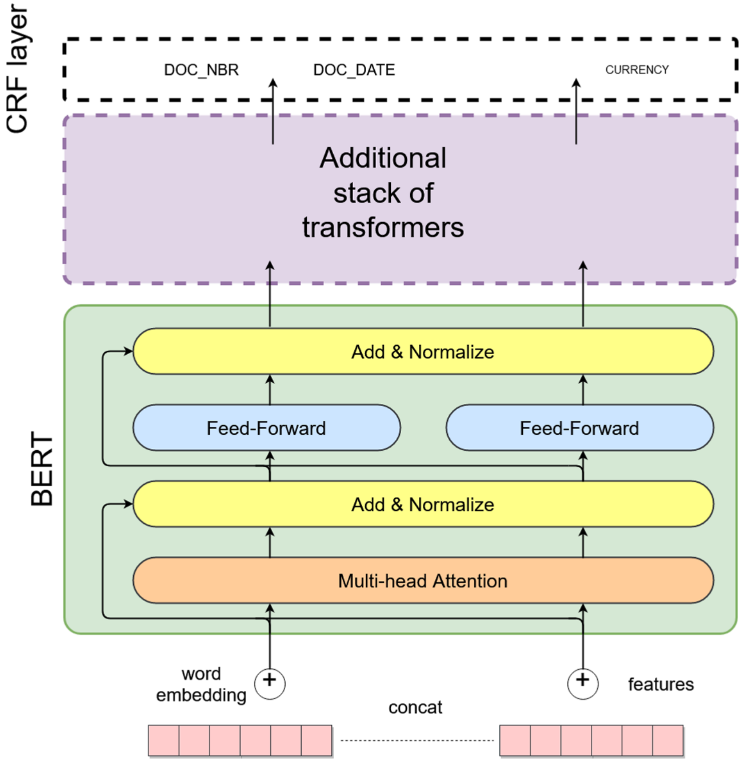
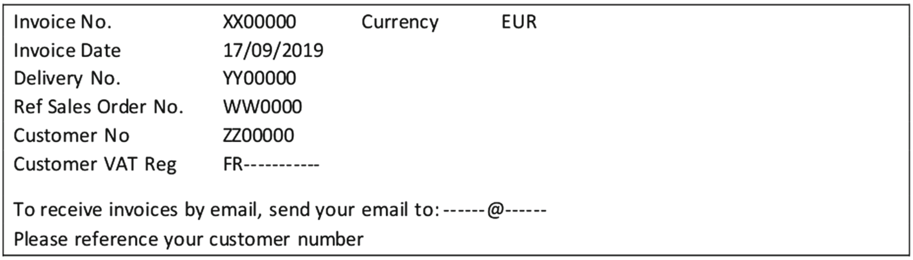


Fig. 2. Architecture of BERT for data extraction

We ran this tool over texts extracted from invoices using an OCR engine. The OCR-generated XML files contain lines and words grouped in blocks, with extracted text aligned in the same way as in regular documents (from top to bottom and from left to right). As OCR segmentation can lead to many errors with the presence of tables or difficult structures, we only kept words from OCR and rebuilt the lines based on word centroid coordinates. The left context therefore allows defining key fields (cf. Fig. 5). However, invoices, as all administrative documents, may have or contain particular structures which should rather be aligned vertically. In tables, for example, the context defining target fields can appear only in the headers. For this reason, we define sequences including the whole text of the document and ensure the presence of the context and the field to be extracted in the same sequence (Fig. 3).



Invoice No.	XX00000	Currency	EUR
Invoice Date	17/09/2019		
Delivery No.	YY00000		
Ref Sales Order No.	WW0000		
Customer No	ZZ00000		
Customer VAT Reg	FR----------		

To receive invoices by email, send your email to: ------@------
Please reference your customer number

Fig. 3. Sample of an invoice

4.3 Data Extraction Using the Class-Based Method

In parallel to NER experiments, and as our end goal is classification rather than sequence labeling, we decided to compare our work to more classical classification approaches from the document analysis community. Our aim is to predict the class of each word within its own context based on our proposed feature vector (textual, spatial, structural, logical). The output class is the category of the item, which can either one of the key fields or the undefined tag. Our work is similar to the CloudScan approach [17], which is currently the state of the art approach for invoice field extraction, our classification is mainly based on features. However, unlike them, our system is language-independent and does not require neither resources nor human actions as pre- and post-processing. Indeed, they build resources such as a lexicon of towns to define a pre-tagging. The latter is represented by Boolean features that check whether the word in processing corresponds to a town or a zip code. In the same way, they extract dates and amounts. In this work, our system is resourceless and avoids any pre-tagging. We define a pattern feature to learn and predict dates and amounts. In this way, we do not need to update lists nor detect language. We also added new Boolean features (cf. Sect. 4.1) to define titles and mathematical assertions (i.e.: isTitle, isTerm, isProduct).

In order to accelerate the process, we proposed a strategy to reduce the volume of data injected to train our models. To this end, we kept N-grams which are associated with one of the ground-truth fields and reduced the volume of undefined elements. In other terms, for each annotated field, we randomly select five words with "undefined" classes as counter-examples. This strategy allowed us to be close to the distribution of labeled terms in natural language processing tasks. For instance, in named entity recognition, it is estimated to 16% the ratio of labeled named entities compared to all the words in a text [20,23]. Furthermore, keeping only 40 n-grams for each document with 8 target fields to be extracted, showed better performance than the classification using all the words of every document.

Finally, our experiments were conducted on the Ludwig[2] open-source toolbox designed by Uber as an interface on top of TensorFlow [15]. This toolbox proposes

[2] https://uber.github.io/ludwig/.

a flexible, easy and open platform that facilitates the reproducibility of our work. From a practical point of view, a list of items combined with its feature vector is provided to Ludwig as input features. We worked at an n-gram level and the textual features were encoded with a convolutional neural network when they were related to the word itself. When they were spatially ordered (e.g. all words to the top, left, right, bottom) we used a bidirectional lstm-based encoder. A concat combiner provided the combined representation of all features to the output decoders. The model was trained with the Adam optimizer [10] using mini-batches of size 96 until the validation performance had not improved on the validation set for 5 epochs. The combiner was composed of 2 fully connected layers with 600 rectified linear units each. We applied a dropout fraction of 0.2 in order to avoid over-fitting.

5 Results

In order to evaluate our methods, we used two traditional metrics from the information retrieval field: precision and recall. While precision is the rate of predicted key fields correctly extracted and classified by the system, recall is the rate of fields present in the reference that are found and correctly classified by the system. The industrial context involves particular attention to the precision measure because false positives are more problematic to customers than missing answers. We therefore aim to reach a very high precision with a reasonable recall.

We report our results on 8 fields: the **docType** and **docNbr** respectively define the type (i.e. regular invoices, credit notes, orders, account statements, delivery notes, quotes, etc.) and the number of the invoice. The **docDate** and **dueDate** are respectively the date on which the invoice is issued and the due date by which the invoice amount should be paid. We additionally extract the net amount **netAmt**, the tax amount **taxAmt** and the total amount **totAmt** as well as **the currency**. Table 2 shows results of the first experiment which has been conducted using the NER-based model and the class-based system.

These first results show that the class-based system outperforms NER-based on most fields. Except for amounts, NER-based has a good precision for all while the class-based system rightly manages to find all the fields with high precision and recall. Despite the high performance of NER-based in the NER task, the system showed some limits over invoices which we explain by the ratio between undefined words and named entities which is much bigger in the case of invoices. Having many undefined tokens tends to disturb the system especially when the fields are redundant in the documents (i.e. amounts) unlike fields that appear once in the document, for which results are quite good. One particularity of the amount fields is that they often appear in a summary table which is a non-linear block that contains many irrelevant words.

Table 2. First results using the NER-based model and the class-based system over the database-20k. "Support" stands for the number of occurrences of each field class in the test set. Best results are in **bold**.

Fields	Support	Recall		Precision	
		NER-based	class-based	NER-based	class-based
docType	3856	0.79	**0.84**	**0.97**	**0.97**
docNbr	3841	0.66	**0.87**	0.74	**0.86**
docDate	3897	**0.78**	0.73	0.94	**0.95**
dueDate	2603	0.73	**0.78**	**0.94**	0.92
netAmt	3441	0.47	**0.72**	0.58	**0.78**
taxAmt	3259	0.47	**0.70**	0.65	**0.86**
totAmt	3889	0.45	**0.85**	0.59	**0.89**
Currency	2177	0.79	**0.97**	**0.83**	**0.83**

In order to improve the results, we firstly visualized the weights of the features in the attention layer at the last encoder of the NER-based neural network (cf. Fig. 4). These weights indicate relevant features for each target field.

Figure 4 indicates the weights of the best performing epoch in the NER-based model. We can notice from the figure that many features have weights close to zero (with ocean blue) for all the fields. Features such as the position of the word in the document, block and line are unused by the final model and considered as irrelevant. Furthermore, it is clear that the relative position of the word in the document page (rightMargin, bottomMargin) are high-weighted in the predictions of all the fields. For the amount fields, the logical features as well as the relative margin position of the word on the left and on the top are also relevant. It is shown using white or light blue colors. We therefore conducted a second experiment, keeping only the most relevant features. We trained new models without considering the right and the bottom relative positions of the word and its positions in the document, line and block.

Table 3 shows practically better results for all the target fields. Except for the recall of docDate using the class-based system which is considerably degraded, all the other results are either improved or kept good performance. Even if the results are improved using relevant features, the NER-based system nevertheless showed some limits to predict amounts. This is not totally unexpected given that NER systems are mainly adapted to extract information from unstructured texts while amounts are usually indicated at the end of tables with different reading directions as shown in Fig. 5. We assume that an additional feature defining fields in tables can particularly improve the amounts' fields.

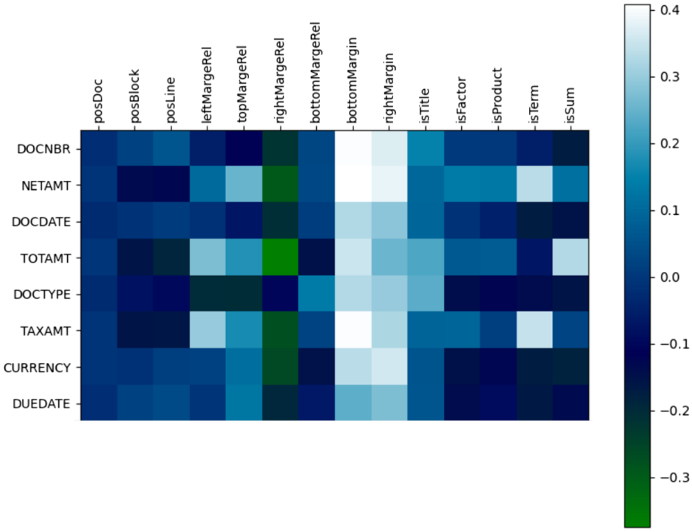


Fig. 4. Weights of features used by the NER based method. Features: position of the word in the document, line and block (table, paragraph or frame) **posDoc, posLine, posBlock**; relative position of the word compared to its neighbours **leftMargeRel, topMargeRel, rightMargeRel, bottomMargeRel**; relative position of the word in the document page **rightMargin, bottomMargin**; Boolean features for titles and mathematical operations **isTitle, isFactor, isProduct, isSum, isTerm**.

Table 3. Results of the NER-based model and the class-based system over the database-20k using relevant features. Best results are given in **bold**. * denotes better results compared to Table 2 (*i.e.* without feature selection)

Fields	Support	Recall		Precision	
		NER-based	class-based	NER-based	class-based
docType	3856	0.81*	**0.85***	**0.98***	0.97
docNbr	3841	0.67*	**0.86**	0.74	**0.86**
docDate	3897	**0.78**	0.33	**0.95***	0.92
dueDate	2603	**0.74***	0.70	**0.93**	0.91
netAmt	3441	0.47	**0.78***	0.58	**0.82***
taxAmt	3259	0.49*	**0.78***	0.66*	**0.87***
totAmt	3889	0.49*	**0.87***	0.61*	**0.89**
Currency	2177	0.82*	**0.96**	**0.83**	**0.83**

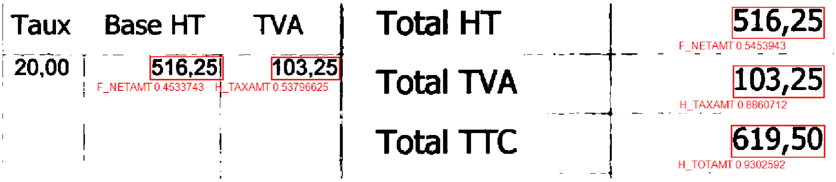

Taux
Base HT
TVA
20,00
516,25
F_NETAMT 0.4533743
103,25
H_TAXAMT 0.53796625
Total HT
Total TVA
Total TTC
516,25
F_NETAMT 0.5453943
103,25
H_TAXAMT 0.8860712
619,50
H_TOTAMT 0.9302592

Fig. 5. Different reading directions on the same invoice

Finally, we evaluated the impact of the volume of documents used to train the systems. New models have been generated on the 100k-database documents. Table 4 summarises the performance measures. By increasing the volume, we also increase the number of different templates available in the dataset. We also report the results of the CloudScan tool [17] for comparative reasons. Even though we are not using the same dataset compared to CloudScan, we believe this can give a global idea of the performances achieved in the state of the art using an accurate system focusing on invoices and being evaluated in the same fields. Table 4 indicates the best CloudScan's results reached using a model trained on more than 300k invoices with the same templates as those in the test set.

Table 4. Results of the NER-based model and the class-based system over the database-100k using relevant features. Best results are given in **bold**.

Fields	Recall			Precision		
	CloudScan [17]	NER-based	class-based	CloudScan [17]	NER-based	class-based
docType	–	0.79	**0.90**	–	**0.99**	0.97
docNbr	0.84	0.69	**0.89**	0.88	0.85	**0.89**
docDate	0.77	0.78	**0.94**	0.89	**0.96**	**0.96**
dueDate	–	0.74	**0.90**	–	**0.96**	0.93
netAmt	**0.93**	0.47	0.81	**0.95**	0.62	0.86
taxAmt	**0.94**	0.49	0.79	**0.94**	0.70	0.90
totAmt	**0.92**	0.44	0.87	0.94	0.63	**0.95**
Currency	0.78	0.76	**0.98**	**0.99**	0.90	0.84

The results in Table 4 are quite promising regarding the small volume of data we used in these experiments. For some fields (e.g. document number), they can even be compared to our baseline. Unsurprisingly, the results are clearly improved for both the recall measure and the precision measure.

All in all, we can appreciate that, 100k sample is much less-sized than the corpus used to demonstrate the CloudScan system (more than 300k) and moreover, with only 20k training samples, the performance is yet very honorable.

6 Conclusion

This paper is dedicated to the extraction of key fields from semi-structured documents. We have conducted an evaluation on both NER-based approaches and word classification-based works, with a constraint of reduced training sets. Our main contribution considerably reduces the amount of required data by only selecting reliable data for the training and development. Our solution can easily cope with unseen templates and multiple languages. It neither requires parameter settings nor any other prior knowledge. We got comparable performances to the CloudScan system with a much smaller volume of training data. Reducing the amount of training data required is a significant result since constructing training data is an important cost in business document processing, to be replicated for every language and template.

We have implemented a network that uses neither heuristics nor post-processing related to the specific domain. Therefore, this system could easily be adapted for other documents such as payslips. Processing invoices raise issues because of the specific organization of the information in the documents, which rarely contain full sentences or even natural language. Key fields to be extracted are very often organized as label-value pairs in both the horizontal and the vertical direction. That makes the information extraction step particularly challenging for capturing contextual data. NLP approaches are seriously challenged in such a context. We also showed, in this paper, that it was possible to highly decrease the processing time to train a model which were still efficient and which fit our industrial context better. With only a small volume of data, we obtain promising results by randomly choosing some undefined items for each annotated field. As we consider the string format of the words, this kind of filtering is not dedicated to invoices. The experiment which has been conducted on a bigger volume shows an improvement of the recall measure. The precision value is also a bit better, although not significantly. Thus, it seems that a bigger volume of documents can make the system more generic because there are more different templates available. To make the precision value increase, we assumed that the more efficient way would be to work on the quality of the training data. Indeed, this neural network has been trained on an imperfect ground-truth, generated by the current system. In addition, we had to assume that the end-user had checked the output of the information extraction system, but this is not always true since in practice only the information required by his company is kept.

As future work, we are considering an interactive incremental approach in order to improve the performance with a minimal set of information. The main idea would be to use the current output to initiate the training process and then regularly train the network again. The more the end-user will check the output, the more the extraction will improve. Moreover, the error analysis showed that 2D information could improve performance and that 2D transformers are a promising prospect for a future work.

Acknowledgements. This work is supported by the Region Nouvelle Aquitaine under the grant number 2019-1R50120 (CRASD project) and AAPR2020-2019-8496610 (CRASD2 project), the European Union's Horizon 2020 research and innovation program under grant 770299 (NewsEye) and by the LabCom IDEAS under the grant number ANR-18-LCV3-0008.

References

1. Boroş, E., et al.: Alleviating digitization errors in named entity recognition for historical documents. In: Proceedings of the 24th Conference on Computational Natural Language Learning, pp. 431–441 (2020)
2. Chiu, J.P., Nichols, E.: Named entity recognition with bidirectional LSTM-CNNs. arXiv preprint arXiv:1511.08308 (2015)
3. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. J. Mach. Learn. Res. **12**, 2493–2537 (2011)
4. Conneau, A., Lample, G.: Cross-lingual language model pretraining. In: Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 32, pp. 7059–7069. Curran Associates, Inc. (2019). http://papers.nips.cc/paper/8928-cross-lingual-language-model-pretraining.pdf
5. Dengel, A.R., Klein, B.: smartFIX: a requirements-driven system for document analysis and understanding. In: Lopresti, D., Hu, J., Kashi, R. (eds.) International Workshop on Document Analysis Systems, DAS 2002. LNCS, vol. 2423, pp. 433–444. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45869-7_47
6. Dernoncourt, F., Lee, J.Y., Szolovits, P.: Neuroner: an easy-to-use program for named-entity recognition based on neural networks. arXiv preprint arXiv:1705.05487 (2017)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
8. Grishman, R., Sundheim, B.M.: Message understanding conference-6: a brief history. In: COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics (1996)
9. Holt, X., Chisholm, A.: Extracting structured data from invoices. In: Proceedings of the Australasian Language Technology Association Workshop 2018, pp. 53–59. Dunedin, New Zealand, December 2018
10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
11. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360 (2016)
12. Lohani, D., Belaïd, A., Belaïd, Y.: An invoice reading system using a graph convolutional network. In: Carneiro, G., You, S. (eds.) ACCV 2018. LNCS, vol. 11367, pp. 144–158. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21074-8_12
13. Ma, X., Hovy, E.: End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. arXiv preprint arXiv:1603.01354 (2016)
14. Martin, L., et al.: Camembert: a tasty French language model. arXiv preprint arXiv:1911.03894 (2019)
15. Molino, P., Dudin, Y., Miryala, S.S.: Ludwig: a type-based declarative deep learning toolbox (2019)

16. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investigationes **30**(1), 3–26 (2007)
17. Palm, R.B., Winther, O., Laws, F.: CloudScan - a configuration-free invoice analysis system using recurrent neural networks. CoRR abs/1708.07403 (2017), http://arxiv.org/abs/1708.07403
18. Poulain d'Andecy, V., Hartmann, E., Rusinol, M.: Field extraction by hybrid incremental and a-priori structural templates. In: 13th IAPR International Workshop on Document Analysis Systems, DAS 2018, Vienna, Austria, 24–27 April 2018, pp. 251–256, April 2018. https://doi.org/10.1109/DAS.2018.29
19. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training (2018)
20. Reimers, N., Eckle-Kohler, J., Schnober, C., Kim, J., Gurevych, I.: GermEVAL-2014: nested named entity recognition with neural networks (2014)
21. Rusiñol, M., Benkhelfallah, T., D'Andecy, V.P.: Field extraction from administrative documents by incremental structural templates. In: ICDAR, pp. 1100–1104. IEEE Computer Society (2013). http://dblp.uni-trier.de/db/conf/icdar/icdar2013.html#RusinolBD13
22. Sage, C., Aussem, A., Elghazel, H., Eglin, V., Espinas, J.: Recurrent neural network approach for table field extraction in business documents. In: 2019 International Conference on Document Analysis and Recognition, ICDAR 2019, Sydney, Australia, 20–25 September 2019, pp. 1308–1313, September 2019. https://doi.org/10.1109/ICDAR.2019.00211
23. Sang, E.F., De Meulder, F.: Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. arXiv preprint cs/0306050 (2003)
24. Vaswani, A., et al.: Attention is all you need. arXiv preprint arXiv:1706.03762 (2017)
25. Wu, Y., et al.: Google's neural machine translation system: bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144 (2016)
26. Zhao, X., Niu, E., Wu, Z., Wang, X.: Cutie: learning to understand documents with convolutional universal text information extractor. arXiv preprint arXiv:1903.12363 (2019)