2011 International Conference on Document Analysis and Recognition

Table Content Understanding in smartFIX

Florian Deckert, Benjamin Seidler, Markus Ebbecke, Michael Gillmann
Insiders Technologies GmbH
Brüsseler Straße 1, 67657 Kaiserslautern, Germany
{*f.deckert, b.seidler, m.ebbecke, m.gillmann*}*@insiders-technologies.de*

Abstract**—The analysis of table structures and the retrieval of table contents is widely agreed to be a difficult challenge in the area of document analysis systems. Instead of extracting the layout of tables, we are interested in understanding their content. In this paper, we present and discuss the smartFIX approach to table recognition and content extraction. Rather than relying on layout features only, we recognize tables by taking into account the presence and semantics of data entities that we expect to find contained in a table. The relationship of a document, including a table, to a specific business process aids in shaping helpful knowledge and expectations about the table's content. smartFIX is a commercial document analysis system complying with the complete bandwidth of industrial requirements. Therefore, smartFIX must locate the tables and extract its business process relevant information with high reliability.**

Keywords**-document analysis; smartFIX; table recognition; table analysis; table understanding; table content extraction.**

I. Introduction

Document capturing systems like smartFIX [1] process millions of documents in several hundred customer installations every day. The software supports the automation of business processes (BPs) triggered by incoming documents like letters, forms, invoices, or orders provided by document scanners, fax and e-mail servers (cf. Figure 1). smartFIX gathers specific information needed within the BPs, e.g. the name and address of the customer and a set of article numbers on an order document. Typically, smartFIX starts by classifying an incoming document, relating it to a BP. Afterwards the system extracts BP-specific information using a number of extraction strategies. Finally mathematical and logical constraints [2] as well as fuzzy database matching strategies help to unify results and to provide confidence metrics. Accurately and unambiguously recognized information is ready for further processing in subsequent systems. Inaccurately recognized, i.e. "uncertain", values are forwarded to a verification workplace for human review. Afterwards, the quality-controlled data is used for deriving "learn rules", i.e. extraction rules used in addition to manually configured extraction strategies, improving the recognition process. Finally, the collected and verified data is directed into desired subsequent systems – e.g. an enterprise resource planning system like SAP – for further processing.

Many of the documents processed in smartFIX contain tables of vital information. Examples are position tables on invoices or tables containing requested items on orders. Table recognition and information retrieval from tables is understood as a difficult task [3], [4]. The structure of tables is usually defined as a labeled grid of data. The key issue concerning tables is that the 2D layout carries certain relationships between contained data entities – their logical structure (cf. [5]). Like Lopresti, we approach the table structure from a relational database perspective. Further, we understand the table's interpretation as the retrieval of the logical structure. Recognizing the logical structure is made difficult in tables due to a lack of physical clearness, complex and diverse structure among table instances, and missing or noisy information. Figure 2 shows an invoice table row, demonstrating that table content understanding is not necessarily straightforward. There is no clear layout that

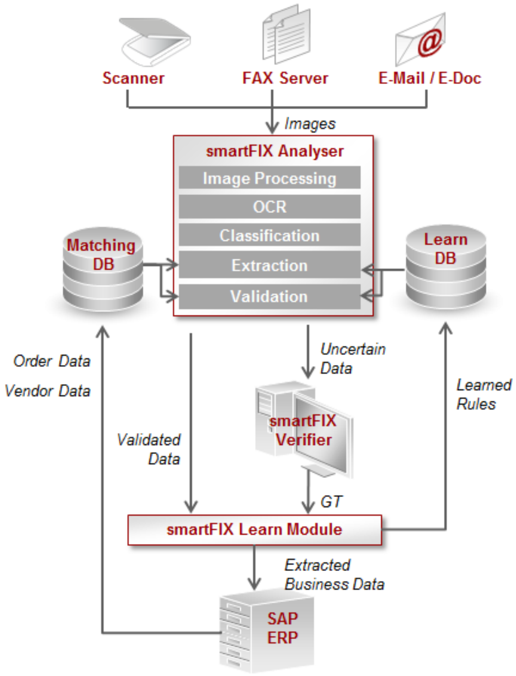
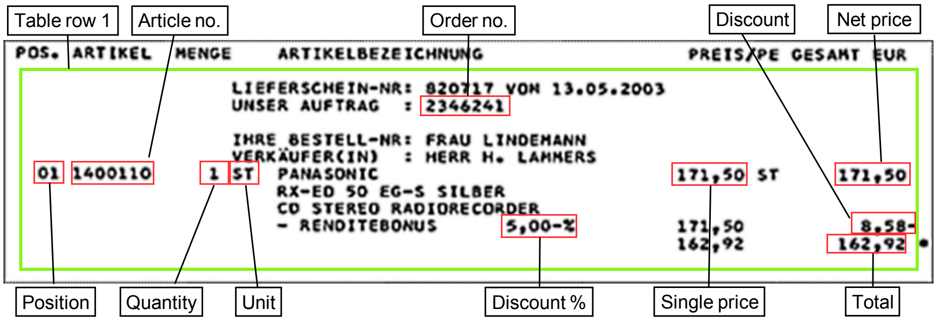


Figure 1. smartFIX is a document capturing system, capable of processing any kind of written communication. It assigns incoming documents to business processes, afterwards extracting and treating contained data and finally feeding results into e.g. enterprise resource planning systems.

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.104



```
POS. ARTIKEL  MENGE    ARTIKELBEZEICHNUNG                    PREIS/PE GESAMT EUR

                       LIEFERSCHEIN-NR: 820717 VON 13.05.2003
                       UNSER AUFTRAG   : 2346241

                       IHRE BESTELL-NR: FRAU LINDEMANN
                       VERKÄUFER(IN)   : HERR H. LAMMERS
 01 1400110      1 ST  PANASONIC                            171,50 ST     171,50
                       RX-ED 50 EG-S SILBER
                       CD STEREO RADIORECORDER
                       - RENDITEBONUS        5,00-%         171,50          8,58-
                                                            162,92        162,92 •
```

Figure 2. Table row with labeled columns on an invoice. Note that relevant data is inter-stratified by irrelevant data. The geometrically irregular arrangement of relevant data further complicates its retrieval.

helps to identify columns and the BP only needs a subset of the provided information. Our experience is therefore that we cannot only rely on a physical structure. We must consider expectations about the presence and semantics of certain data entities in order to understand a table's content. Table extraction in smartFIX is based on these expectations. In the following, we review related work concerning table structure analysis and content extraction. Furthermore, we present our approach, show results of its performance in multiple customer use contexts and conclude with a few directions we want to pursue to further improve our algorithm.

II. Related Work

Table structure recognition has been subject of much research [4], [6]. In addition to advancements in table detection [7], [8], the community has developed sophisticated methods of decomposing tables into their entities. Recent approaches are based on varying definitions of a table's appearance and their goal is to provide an estimate of the location of table columns, rows, and/or cells. Approaches may include robustness features like allowing for row-wise interleaving of text [9]. By starting with a hypothesis generated from geometric properties of text segments on document images, [10] introduces an approach that applies statistical methods. Resulting table entity positions are refined through an iterative updating scheme. Further tuning is achieved using parameters determined by training on a set of artificial data. [11] uses Conditional Random Fields to label table lines in plain text data. This statistical model imposes a weighted set of constraints on geometric features and characters contained in a text line. Relations between the table's entities enable inference of correct labels. [12] shows that it is promising to use a classifier trained on features which combine certain character sequences typical for tables. Layout features like white-spaces or separators thereby help to determine the label of text segments.

The approaches mentioned above differ from ours in several aspects. First of all, most of them aim at obtaining a table structure as their result. However, we need to understand the table's content in order to extract information relevant to a BP. With a table structure at hand it may still be difficult to understand the contents as [3] points out. Information contained in tables may be ambiguous and subject to interpretation in order to be distinct. Secondly, the variety of the tables' appearance is a challenge that has not been sufficiently addressed and remains the "ultimate goal of table understanding", according to [4]. [3] identifies challenges that we have to deal with when retrieving information from tables.

From our industrial perspective, there are a few concerns that we would like to stress or add. Table samples presented in approaches mentioned earlier suggest that available approaches to table structure recognition seem to be designed for working with a well-formed, regular shaped table layout. However, we observe very frequently that tables do not comply with this expectation to tables. In fact, a table may not exhibit a regular structure wrt. data entities that are to be extracted. As we see in Figure 2, relevant information is scattered throughout multi-line rows/cells. Even worse, this information is inter-stratified by irrelevant data. A further practical concern is that there is no world-wide standard in place, defining the layout of certain document types, say invoices. This fact results in a diversity of table layouts so that uniform assumptions about layout are difficult to devise. The inhomogeneous nature of tables is well addressed in [5]. Additionally, meta information, e.g. headers or labels, may be present in some tables but not in others. Finally, because the majority of tables we process originate from scanned documents, our table extraction has to be robust to errors caused by previous optical character recognition or poor scan quality.

There are approaches that distinguish themselves in that they address subsets of issues pointed out above. Relating the table's data entities to one another through Part-of-Speech tagging is an idea expressed in [13]. There, expectations towards the table's content are used in order to give meaning to a table's data entities. However, the described system is specific to retrieving tables of contents, only. An approach that could help to deal with the variety of tables' appearance is the use of repeated structure within tables [14]. The basic assumption is that table rows have a similar appearance within one document. The algorithm is able to identify a table's data entities on a document, e.g. an invoice, provided user annotations specify the structure of the first table row. However, the need for user action would render the approach as inapplicable within a document capturing system. Still, its authors state that first results of combing this with an automatic annotation appear promising.

In the following we show an approach that tries to address challenges above. Its goal is to extract and understand the contents of a table in order to interpret them correctly according to domain-specific requirements.

III. Approach

smartFIX's table understanding algorithm is expectation-driven. As shown in Figure 1, the first processing step is the classification of a document, i.e. the assignment to a BP. The data model of this BP defines expectations for a set of columns. Given an invoice, this might be an article number, quantity, single and total net price. Not all columns on the document need to be part of the data model, and not all columns defined by the BP have to be present on the document (e.g. discounts). In essence, the table is assigned domain-specific expectations concerning the table's content defined through semantics given by a BP. Those expectations provide crucial information for understanding the table's content (cf. [13]).

Entities of a table – mainly its cells – are subsets of the words in a document. Let C be the set of columns defined by the data model. For each column $c \in C$, smartFIX has a set of expectations E_c. Those result from the BP's content constraints, like regular expressions, possible headings, alignment (left, right, centered), etc. Furthermore, we have expectations resulting from implicit table constraints, like geometrical dependencies. In addition to column-specific expectations we consider cross-column constraints, named global expectations E_g. For example, two columns have to be disjunctive, all column headings should have comparable vertical positions, etc.

Let $\hat{W}$ be the set of all words in a document. We have to find a subset $\hat{S}_c \subset \hat{W}$ for each column $c \in C$, which matches the expectations E_c as best as possible.

Let $S = (S_1, ...S_n)_{n=|C|}$ denote a possible column configuration, i.e. a solution. Then we define a quality measure $Q(S_c)$ using a metric $m(S_c, e)$ that estimates how well a set of words S_c matches one expectation e.

$$Q(S_c, E_c) = \sum_{e \in E_c} m(S_c, e) \tag{1}$$

To account for the global expectations E_g we use another metric $g(S)$ to obtain an overall quality measure.

$$Q(S) = \sum_{c \in C} Q(S_c, E_c) + g(S) \tag{2}$$

An optimal solution $\hat{S}$ maximizes this quality measure.

$$\hat{S} = \arg\max Q(S) \tag{3}$$

The above optimization problem suffers from a severe combinatorial explosion and is therefore helplessly ill-posed. Additionally, we have to choose appropriate expectations and metrics for it. In doing so, we have to respect the input data which often contains many inaccuracies. Examples are OCR recognition errors, or inexact alignments. In addition, table design and structure vary highly, and hence we have to respect the fact that we cannot define expectations that are sufficient for all shapes of, say invoices. To cope with the combinatorial problem of solving the optimization, we resort to heuristics that help us to eliminate input sets or column configurations S in early stages of the algorithm. A first measure we take is to reduce the size of the initial search space, i.e. the number of words the algorithm takes as its first input set.

A table detection algorithm is suitable to limit the area on the document that may contain a table and therefore the words to be further evaluated. We do this by using geometrical information similar to [9] in order to minimize possible search regions. Furthermore, we use customer knowledge about the document. Their databases usually contain the recipient of the document at hand. This gives us a hint about the position of the table on the document, because e.g. in orders the table is located below the recipient in most cases. Thereby we can further constrain the amount of input words.

After we obtain a constrained set of words $W \subset \hat{W}$, our algorithm continues by generating initial column candidates, i.e. subsets $S_c^i \subset W$. Although we have greatly reduced the number of input sets, evaluating every possible subset of W is not feasible. Therefore, we use a subset of expectations $E_c^{init} \subset E_c$ for generating $\{S_c^i \subset W\}_{i=1,...,m}$. We rate words $w \in W$ regarding initial expectations $e_c^{init} \in E_c^{init}$ by a function $f_{e_c^{init}}(w)$. By defining

$$m(S_c^i, e_c^{init}) = \sum_{w \in S_c^i} f_{e_c^{init}}(w) \tag{4}$$

we improve the quality rating $Q(S_c^i, E_c)$ whenever $f_{e_c^{init}(w)} > 0$. This allows us to add promising words one-by-one to a column candidate. Expectations that we use in our approach for that purpose are matching regular expressions derived from the semantics of column c given by a BP. For example consider a BP dealing with invoices. Then, if c is the "single price" column, we expect that its content must be numbers, periods or commas. Additionally, we use alignment information, e.g. for the "single price" column contents we may use the knowledge that it is usually right-aligned.

After initial column candidate generation, we continue by rating each candidate using the quality measure. Initially, we compute $Q(S_c^i, E_c)$ for each column candidate $S_c^i, c \in C, i = 1, \dots, m$. The remaining question is which expectations lead to obtaining good results. We found that a combination of layout and content-related expectations performs well.

- Based on the BP's database column c we define regular expressions that match terms in a dictionary of headings h_c. If a term above the bounding box of S_c matches h_c we increase the quality measure. For another column $g \in C$, if there is a term above S_g that matches h_c, we reduce the quality measure for S_c.
- Depending on a BP we might know that, e.g. the rightmost column of a position table on an invoice contains the respective position's price. Based on the geometries of words contained in the column candidates we obtain a geometric ordering. We use that to match these geometries against the preferred position of the database column.
- Further we evaluate an expected top coordinate. The larger the distance of a candidate S_c to the expected top coordinate, the lower the rating.
- For customizing purposes, the candidates can be evaluated using a quality metric written in a scripting language. This allows customers to optimize the algorithm concerning special characteristics of certain kinds of documents and BPs.

After obtaining local quality measures for each column candidate we narrow down their number by rejecting those with low ratings. The number of remaining candidates is then small enough to solve the optimization problem by exhaustive enumeration. To do that we combine column candidates and try all remaining column configurations $\mathfrak{S}$. We evaluate each $S \in \mathfrak{S}$ by imposing global expectations (cf. Equation 2), among them most importantly:

- We expect that the geometry, i.e. bounding box, of a good quality column candidate $S_c \in S$ is separate from another $S_g \in S$. Thus, we decrease the quality of S if we find that $geo(S_c) \cap geo(S_g)$, $geo(R)$ denoting the bounding box of a set of words R.
- Further we reward a column candidate configuration that contains candidates that share approximately the same height.
- If we know the semantics of a column we may use mathematical relationships between the data. For instance a column containing totals should contain larger numbers than a column containing single prices.

After having obtained an overall quality estimate concerning all remaining column configurations, we use the configuration maximizing Equation 3 as the final result.

Given the position of all the columns to be extracted, the problem combining the columns' contents to rows remains, i.e. assign each $w \in \hat{S}_c, c \in C$ to a row r so that $w \in T_r$, where $T \subset W$ is the set of words $w \in W$ in row r. This is not always trivial [3] since values belonging together – in one row – often do not have the same vertical positions on the document (cf. Figure 2). For that purpose, we use a rating h respecting attributes like column size and their top coordinate to find the best column B, i.e. $B = \arg\max h(\hat{S}_c)$. Each word in B is assigned a row r, so that $\forall_{w \in B} \exists_r T_r = \{w\}$. For the remaining columns $\hat{S}_c$, the words $w \in \hat{S}_c$ are assigned to the best matching row r column by column.

However, on the one hand, not every column needs to have a value in each row, i.e. it is possible that $\exists_{c \in C} \nexists_{w \in T_r} w \in \hat{S}_c$. On the other hand, an additional row is added if a column has more values and there is enough space between the rows of the best column.

The initially created table grid is now adapted concerning constraints E_r like overlapping cells, the row's number of entries, their geometrical position (e.g. too low) and their content. Furthermore, we clip the table from below if a certain pattern z usually representing a table end – e.g. "Total" – occurs on the document, i.e. rows are removed if $\exists_{zinZ} \forall_{w \in T_r} y(w) > y(z)$, where Z is a set of table end patterns and $y(w)$ is the distance of w from the document's top. This appropriate expectation provides useful information for understanding the table based on semantic knowledge. The set of words describing table ends can be adapted according to the customer's requirements.

Finally, the contents of all cells are extracted from the deduced table grid, which represents the table.

Additionally, learning can be applied to certain table attributes – e.g. regular expression and column position – which improves results over time.

IV. Results

The presented approach is successfully used for processing e.g. invoice and order documents by over 150 of our customers. They utilize smartFIX's table recognition to extract and understand table contents in various contexts. Table I shows results of our approach in different real-world scenarios.

1) ”Order”: tables are recognized on orders for industrial parts. The tables are lists of ordered articles and the desired amounts. Often they contain article numbers, prices, and arranged discounts.
2) ”Invoice”: tables are extracted from commercial invoices. These are lists of goods and services with amounts, prices, discounts, taxes, etc.
3) Finally, ”Medical” shows results from medical invoices created by doctors, dentists, or hospitals. These contain lists of treatments.

We used smartFIX's verification process (cf. Introduction) to collect the ground truth data from human experts.

	# Docs	# Cells	+	V+	V-
Order	663	15778	57.4 %	40.5 %	2.52 %
Invoice	3366	19190	68.9 %	59.5 %	0.85 %
Medical	13298	157802	84.5 %	78.1 %	2.86 %

Table I
TABLE UNDERSTANDING RESULTS ON REAL WORLD CUSTOMER DATA

The table item recognition rate "+" is the percentage of correctly extracted cells – both column, row and content have to match. It is obvious that the recognition performance is much higher in the Medical scenario than in Order and Invoice. Medical tables are less complex, i.e. they often have a clear layout and the number of columns used by the business process is comparatively low. "V+" is a subset of "+" containing cells that are rated as "verified", i.e. the system classifies this data as correct (cf. [15]). Hence, these cells do not have to be verified manually. "V-" is the substitution rate, i.e. the ratio of cells wrongly classified as correct. If a table row has not been extracted, all cells of this row are rated V-. In practice, V- fields are not critical in most cases due to additional restrictions checked by the system. E.g. the sum of the total column is compared to the total invoice amount.

V. CONCLUSION

In this paper, we discuss the smartFIX approach of table understanding. As smartFIX is a commercial software product, it does not only have to locate tables on any kind of document but also extract information relevant to business processes with a very high degree of correctness. The approach is expectation-driven and takes the knowledge about certain business process data entities into account. We present results of smartFIX's table recognition performance in multiple customer settings. These show that the approach yields an applicable document analysis tool that fulfills main industrial requirement – highly reliable content extraction rates on documents with high variability in table layouts.

Directions that we want to pursue to further improve our algorithm include intensifying learning mechanisms concerning the table analysis. We will not only expand learning to more table attributes in our approach, but also utilize "on-the-fly" ground truth data collected automatically when human interactors add or correct table content during a verification step in the smartFIX workflow. These further improvements of our approach strive for even better table understanding results on highly variable documents.

ACKNOWLEDGMENT

The work presented in this paper was performed in the context of the Software-Cluster project EMERGENT (www.software-cluster.org). It was partially funded by the German Federal Ministry of Education and Research (BMBF) under grant no. 01IC10S01. The authors assume responsibility for the content.

REFERENCES

[1] B. Klein, A. Dengel, and A. Fordan, "smartFIX: An adaptive system for document analysis and understanding," in *Reading and Learning – Adaptive Content Recognition*, ser. Lecture Notes in Computer Science, A. Dengel, M. Junker, and A. Weisbecker, Eds., vol. 2956. Springer, 2004, pp. 166–186.

[2] A. Fordan, "Constraint solving over ocr graphs," in *Proc. of the Applications of Prolog, 14th Int. Conf. on Web knowledge management and decision support*. Springer, 2003, pp. 205–216.

[3] J. Y. Hu, R. S. Kashi, D. P. Lopresti, G. Nagy, and G. T. Wilfong, "Why table ground-truthing is hard," in *ICDAR*, 2001, pp. 129–133.

[4] D. W. Embley, M. Hurst, D. P. Lopresti, and G. Nagy, "Table-processing paradigms: a research survey," *IJDAR*, vol. 8, no. 2-3, pp. 66–86, 2006.

[5] D. Lopresti and G. Nagy, "Automated table processing: An (opinionated) survey," in *Proc. of the Third IAPR Int. Workshop on Graphics Recognition*, 1999, pp. 109–134.

[6] R. Zanibbi, D. Blostein, and J. R. Cordy, "A survey of table recognition," *IJDAR*, vol. 7, no. 1, pp. 1–16, 2004.

[7] F. Shafait and R. Smith, "Table detection in heterogeneous documents," in *Proc. of the 9th IAPR Int. Workshop on DAS*, 2010, pp. 65–72.

[8] V. Long, "An RDF-based blackboard architecture for improving table analysis," in *ICDAR*, 2009, pp. 916–920.

[9] T. Kieninger and A. R. Dengel, "Applying the T-recs table recognition system to the business letter domain," in *ICDAR*, 2001, pp. 518–522.

[10] Y. Wang, I. T. Phillips, and R. M. Haralick, "Table structure understanding and its performance evaluation," *Pattern recognition*, vol. 37, no. 7, pp. 1479–1497, 2004.

[11] D. Pinto, A. McCallum, X. Wei, and W. B. Croft, "Table extraction using conditional random fields," in *Proc. of the 26th annual Int. ACM SIGIR Conf. on Research and development in informaion retrieval*, 2003, pp. 235–242.

[12] H. T. Ng, C. Y. Lim, and J. L. T. Koo, "Learning to recognize tables in free text," in *Proc. of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, 1999.

[13] A. Belaïd, L. Pierron, and N. Valverde, "Part-of-Speech tagging for table of contents recognition," in *ICPR*, 2000, pp. 451–454.

[14] E. Bart and P. Sarkar, "Information extraction by finding repeated structure," in *Proc. of the 9th IAPR Int. Workshop on DAS*, 2010, pp. 175–182.

[15] B. Seidler, M. Ebbecke, and M. Gillmann, "smartfix statistics: Towards systematic document analysis performance evaluation and optimization," in *Proc. of the 9th IAPR Int. Workshop on DAS*, 2010, pp. 333–340.