Case-based reasoning for invoice analysis and recognition

Hatem HAMZA[*,**], Yolande BELAID[**], Abdel BELAID[**]

[*]ITESOFT, Parc d'Andron, Le Squoia. 30470 Aimargues. France.
[**]University Nancy 2, LORIA. Vandoeuvre-les-Nancy, France.

hamza,ybelaid,abelaid@loria.fr
http://www.loria.fr/research/equipes/read/

Abstract. *This paper introduces the approach CBRDIA (Case Based Reasoning for Document Invoice Analysis) which uses the principles of case-based reasoning to analyze, recognize and interpret invoices. Two CBR cycles are performed sequentially in CBRDIA. The first one consists in checking whether a similar document has already been processed, which makes the interpretation of the current one easy. The second cycle works if the first one fails. It processes the document by analyzing and interpreting its structuring elements (adresses, amounts, tables, etc) one by one.The CBR cycles allow processing documents from both knonwn or unknown classes. Applied on 923 invoices, CBRDIA reaches a recognition rate of 85,22% for documents of known classes and 74,90% for documents of unknown classes.*

Key words: case-based reasoning, document case, structure case, invoice analysis, invoice interpretation, structure extraction.

1 Introduction

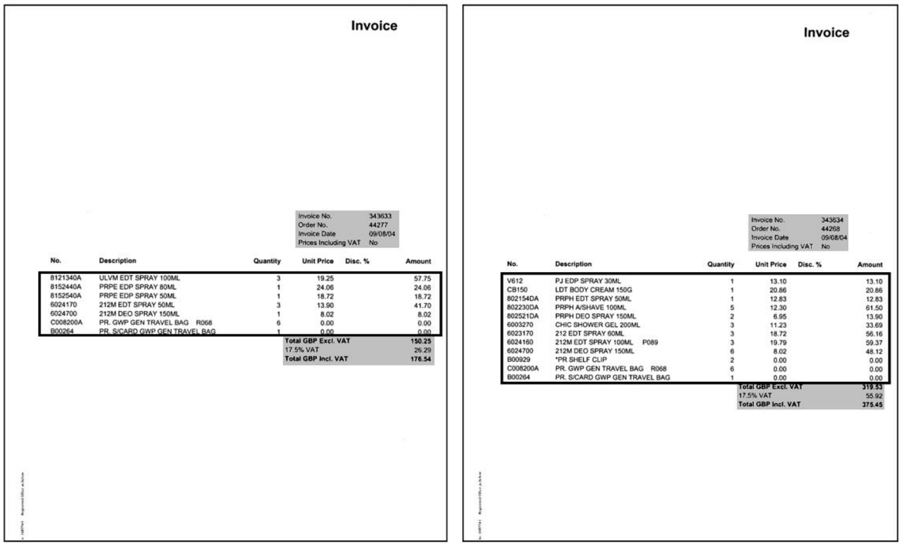
Form and invoice analysis systems in real production chains are often faced with a huge quantity of documents requiring a high processing speed and a continuous adaptation capacity to the structure variation. The manual and even semi-automatic solutions which consist in building manually the model of each set of new documents can no longer be used because of the heavy modeling phase they require [1].

Invoices have variations depending on many factors : the company issuing the invoice, the client, etc. Most current information to be extracted are: addresses (delivery, billing...), total amounts and table lines showing details of services, purchased products... Two types of documents occur in invoice processing:

- documents of known class i.e similar documents have been already processed,
- new documents from an unknwon class.

In two documents from the same class, information blocks (addresses, amounts, tables...) are organized in the same way and have the same relative positions in the documents. However, their absolute positions vary from a document to another, depending on the specific content of each document. Figure 1 shows two documents from the same class where the information to be extracted are gray tone or boxed. We can see that the absolute position of the total amount zone changes between the documents.

For a new document from an unknown class, the problem consists in building a generic and reliable model for all the documents of this class. Figure 2 shows two invoices from different classes. We can clearly see that they have different structures.



Invoice

Invoice No.	343633
Order No.	44277
Invoice Date	09/08/04
Prices Including VAT	No

No.	Description	Quantity	Unit Price	Disc. %	Amount
8121340A	ULVM EDT SPRAY 100ML	3	19.25		57.75
8152440A	PRPE EDP SPRAY 80ML	1	24.06		24.06
8152540A	PRPE EDP SPRAY 50ML	1	18.72		18.72
6024170	212M EDT SPRAY 50ML	3	13.90		41.70
6024700	212M DEO SPRAY 150ML	1	8.02		8.02
C008200A	PR. GWP GEN TRAVEL BAG R068	6	0.00		0.00
B00264	PR. S/CARD GWP GEN TRAVEL BAG	1	0.00		0.00
			Total GBP Excl. VAT		**150.25**
			17.5% VAT		26.29
			Total GBP Incl. VAT		**176.54**

Invoice

Invoice No.	343634
Order No.	44268
Invoice Date	09/08/04
Prices Including VAT	No

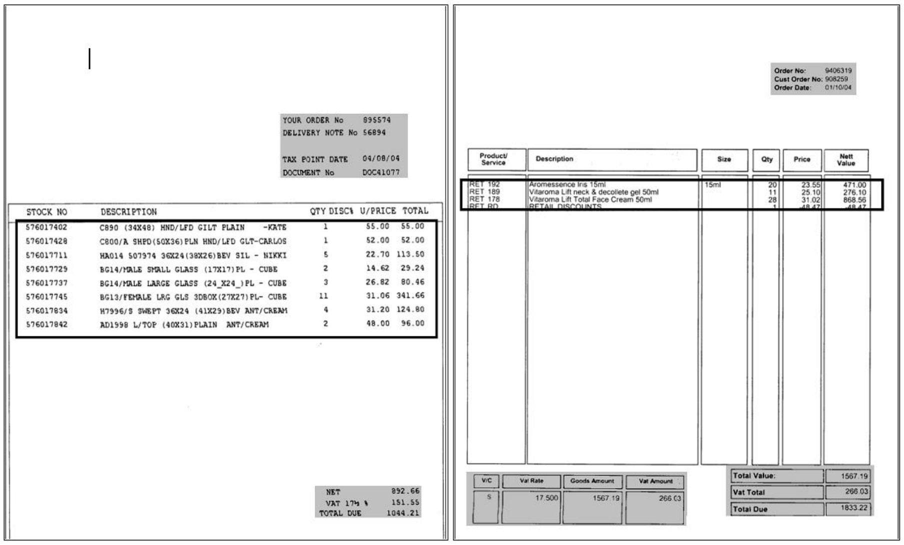
No.	Description	Quantity	Unit Price	Disc. %	Amount
V612	PJ EDP SPRAY 30ML	1	13.10		13.10
CB150	LDT BODY CREAM 150G	1	20.86		20.86
802154DA	PRPH EDT SPRAY 50ML	1	12.83		12.83
802230DA	PRPH A/SHAVE 100ML	5	12.30		61.50
802521DA	PRPH DEO SPRAY 150ML	2	6.95		13.90
6003270	CHIC SHOWER GEL 200ML	3	11.23		33.69
6023170	212 EDT SPRAY 60ML	3	18.72		56.16
6024160	212M EDT SPRAY 100ML P089	3	19.79		59.37
6024700	212M DEO SPRAY 150ML	6	8.02		48.12
B00929	*PR SHELF CLIP	2	0.00		0.00
C008200A	PR. GWP GEN TRAVEL BAG R068	6	0.00		0.00
B00264	PR. S/CARD GWP GEN TRAVEL BAG	1	0.00		0.00
			Total GBP Excl. VAT		**319.53**
			17.5% VAT		55.92
			Total GBP Incl. VAT		**375.45**

Fig. 1. Two invoices from the same class

The paper is organized as the following: section 2 presents some related works. Section 3 introduces the use of CBR the system. Sections 4, 5 and 6 present CBRDIA's architecture. Finally, the seventh section shows the obtained results and their interpretation.

2 Related work

The most promising approaches are those which can process documents of either known or unknown classes [2] [3]. In [3], after a first step realted to document classification, the document is interpreted via its structures (key-words) by combining two levels of knowledge: intra- and inter-classes knowledge. If the document class is recognized, the system looks for the solution using the intra-class knowledge (tags, relative positions of the related object, etc.) and the inter-class



YOUR ORDER No 895574
DELIVERY NOTE No 56894

TAX POINT DATE 04/08/04
DOCUMENT No DOC41077

STOCK NO	DESCRIPTION	QTY	DISC%	U/PRICE	TOTAL
576017402	C890 (34X48) HND/LFD GILT PLAIN -KATE	1		55.00	55.00
576017428	C800/A SHPD(50X36)PLN HND/LFD GLT-CARLOS	1		52.00	52.00
576017711	HA014 507974 36X24(38X26)BEV SIL - NIKKI	5		22.70	113.50
576017729	BG14/MALE SMALL GLASS (17X17)PL - CUBE	2		14.62	29.24
576017737	BG14/MALE LARGE GLASS (24_X24_)PL - CUBE	3		26.82	80.46
576017745	BG13/FEMALE LRG GLS 3DBOX(27X27)PL- CUBE	11		31.06	341.66
576017834	H7996/S SWEPT 36X24 (41X29)BEV ANT/CREAM	4		31.20	124.80
576017842	AD1998 L/TOP (40X31)PLAIN ANT/CREAM	2		48.00	96.00

NET	892.66
VAT 17½ %	151.55
TOTAL DUE	1044.21

Order No: 9406319
Cust Order No: 908259
Order Date: 01/10/04

Product/ Service	Description	Size	Qty	Price	Nett Value
RET 192	Aromessence Iris 15ml	15ml	20	23.55	471.00
RET 189	Vitaroma Lift neck & decollete gel 50ml		11	25.10	276.10
RET 178	Vitaroma Lift Total Face Cream 50ml		28	31.02	868.56
RET RD	RETAIL DISCOUNTS		1	-48.47	-48.47

V/C	Vat Rate	Goods Amount	Vat Amount
S	17.500	1567.19	266.03

Total Value:	1567.19
Vat Total	266.03
Total Due	1833.22

Fig. 2. Two invoices from different classes

knowledge (summarizing knowledge in different invoice classes). If the document is not recognized, then only inter-classes knowledge is used to interpret the extracted information. The application of this approach is however limited to isolated keywords not taking into account more frequent and important structures in forms such as tables, addresses, etc. Concerning table analysis, Belad presented in [4] a morphological tagging approach for invoice analysis. This approach was used in order to tag table columns and fields. However, the processed tables are already extracted before tagging. Contrary to these methods, CBR-DIA extracts and interprets data associated with both table lines and key-words. It can also process documents from both nkown and unknown classes.

To our best knowledge, no directly related work has been published in CBR field. However, we can link this work to other works on textual CBR (TCBR)[5] or on CBR in image processing [6]. In our approach, our cases will be represented either by strings or graphs.Cunningham [7] shows that the use of graphs in case representation can be useful in TCBR.

Another type of related works concerns systems using multiple CBR reasoners. In CBRDIA, we will use 2 CBR reasoners (one for invoices of known class, and another for invoices of unknown class). These reasoners will be sequential. Some other approaches [8] use parallel CBR reasoners in order to enhance the system performance.

3 Case-based reasoning in our approach

CBR is a solving strategy using previous experiences to process new problems that have not been processed before [9].

The problems we are facing in this work are the following:

- document structure extraction: this is a difficult and time consuming problem in industry. Structure extraction is done for every document in order to be interpreted. However, when a whole set of documents (coming from the same client) has the same structuring elements (example : a table, an amount block, and an address block), then the whole set can be represented by a generic model. This is generally done by a user who takes into account a certain number of documents in order to build such a model.
- document classification: there is a continuous flow of documents that have to be processed (read and interpreted). We do not know a priori to which class of documents the processed one belongs to. It is obvious that if the system has processed a similar document before, then it is a real waste of time not to take advantage of such a knowledge. Otherwise, a new document model has to be built in order to extract the desired information.
- document analysis and interpretation: this task (interpreting words, fields, or tables) is really hard. It has to be done either by a user who supervises every document, or automatically by a reliable system. For example, interpreting the word "total" means associating it with the numerical value related to it on the document. The system, in the interpretation phase should
  - generalize easily based on the previous document processing experiences
  - understand the current document, and make profit of the extracted and interpreted information it contains
  - be as quick as possible. If possible, we have to avoid a classical training process as other machine learning techniques.
  - self adaptable to any new class of documents. In classical machine learning techniques (example: neural networks), as soon as a new class of data appears, these techniques fail generally in recognizing it. A new learning has to be done in order to overcome these difficulties. However, in CBRDIA, when the document class is completely new, the system can find solutions (partial solutions if not total ones) by trying to exploit the previous knowledge.

For all these reasons, the choice of CBR was natural. In CBRDIA, two sorts of cases are defined: a document case and a structure case. As shown in Figure 3, the flow of our approach is based on three main steps: problem elaboration, global problem solving and local problem solving.

This flow contains two CBR phases (global solving and local solving). This sequential solving strategy allows the system to deal with the problem at different levels.

Problem elaboration consists in indices extraction from the document. These indices are either key-words (KW) and their spatial relationships, or table rows. This problem is then solved using either global solving process or local solving one.

Global Solving (first CBR cycle on Figure 3) consists in checking if a similar document case exists already. If yes, then the system solves the problem by applying the solution of the database case to this problem. Otherwise, the problem

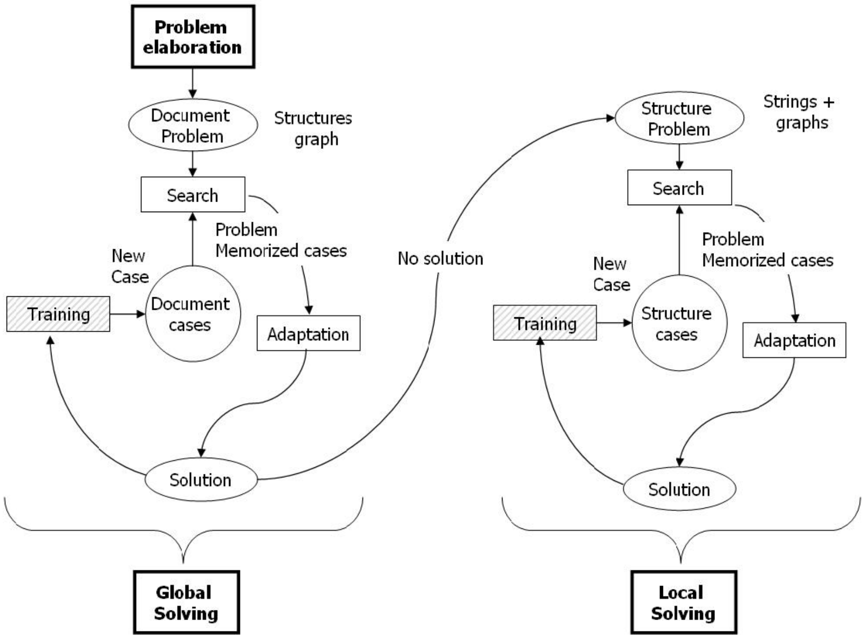


Fig. 3. CBRDIA flow

is decomposed into sub-problems, and solved via the solving of its sub-problems. The second CBR cycle corresponds to this step, and is called local solving. The use of global solving and local solving makes our system able to process any kind of invoice documents.

4 Problem elaboration

The system requires the precise definition of the problem. This precision is required in every step of the flow (case retrieval, solution adaptation). The system input is a raw document given by OCR (optical character recognition). The OCR file written in XML contains the list of words and their coordinates. The document is represented by a set of words $D = W_i$, $i = 1..n$.

4.1 Data extraction and coding

First, each word W is given a list of attributes:

- position (coordiantes in the document)
- KW: If a word in the current document matches a word in a predefined list of keywords, then it is tagged as a KW. This list is enriched gradually as new keywords are discovered.

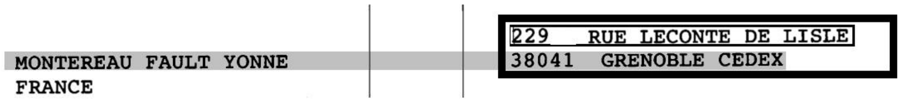
- nature: represented by an alphabetic character: for example, A for numerical, B for alphabetic, etc.

In the next step, fields (Fj) are constituted from D by gathering neighbour words horizontally. Each successive pair of words $d(W_i, W_j)$ in F verifies $d(W_i, W_j) < \alpha$ where α is a threshold depending on the words' size. F is also characterized by a list of attributes:

- position
- nature: the nature of a field is deduced from its words' natures. For example, if Fj contains an alphabetical and a numerical word, then it will be given the tag C for alphanumerical.

From fields, we extract horizontal and vertical lines (HL, VL). We use also the fields' neighbourhoods and the fields' alignments to constitute HL and VL. A vertical line VL is a set of fields F vertically aligned. Two vertical filelds $d(F_i)$ and $d(F_j)$ are in the same vertical line if $d(F_i, F_j) < \delta$ where δ is a threshold depending on the fields' size and position. Similarly, we use a threshold for horizontal fields. A line is described by the following attributes:

- position
- pattern: a string composed of fields' tags list. For example, if the fields' tags in the line are: A, B, B and C, then the pattern is ABBC. These patterns will be very helpful in the extraction of tables.



229 RUE LECONTE DE LISLE
MONTEREAU FAULT YONNE 38041 GRENOBLE CEDEX
FRANCE

Fig. 4. A VL in the big box, a HL in gray tone, a field in the small box

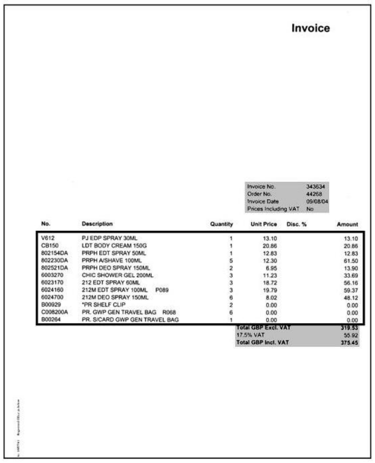
After these elementary information are extracted, we can extract high level structuring elements (S) which can be either pattern structures (PS) when related to tables or key-word structures (KWS) when related to local arrangements of key-words (KW. The final document problem will be defined thanks to PS and KWS.

4.2 Structures Extraction

Figure 5 shows a document containing 2 KWS and a PS.

PS extraction PS are composed of a list of consecutive HLs having similar patterns. This is the case of a table. Figure 5 shows a document containing a PS composed of 12 HLs having the pattern AABBB.

The PS detection process contains three steps:



Invoice

Invoice No.	343634
Order No.	44268
Invoice Date	09/08/04
Prices Including VAT	No

No.	Description	Quantity	Unit Price	Disc. %	Amount
V612	PJ EDP SPRAY 30ML	1	13.10		13.10
CB150	LDT BODY CREAM 150G	1	20.86		20.86
802154DA	PRPH EDT SPRAY 50ML	1	12.83		12.83
802230DA	PRPH A/SHAVE 100ML	5	12.30		61.50
802521DA	PRPH DEO SPRAY 150ML	2	6.95		13.90
6003270	CHIC SHOWER GEL 200ML	3	11.23		33.69
6023170	212 EDT SPRAY 60ML	3	18.72		56.16
6024160	212M EDT SPRAY 100ML P089	3	19.79		59.37
6024700	212M DEO SPRAY 150ML	6	8.02		48.12
B00929	*PR SHELF CLIP	2	0.00		0.00
C008200A	PR. GWP GEN TRAVEL BAG R068	6	0.00		0.00
B00264	PR. S/CARD GWP GEN TRAVEL BAG	1	0.00		0.00
			Total GBP Excl. VAT		319.53
			17.5% VAT		55.92
			Total GBP Incl. VAT		375.45

Fig. 5. An invoice containing 2 KWS and a PS

- For each HL, we constitute a list of HL neighbours HLN using edit distance on their strings (i.e. patterns). We use the threshold 1 (usually equal to 1 order to accept only 1 transformation between strings) between HL patterns to find neighbours.
- The list of each HL neighbours is studied based on the fields' positions.

A	B	B	
A		B	B

Fig. 6. Two patterns with edit distance=0

In Figure 6, the edit distance between the patterns is null, as they represent the same string ABB. However they do not correspond to the same PS because of the difference of the spatial positions. To avoid such confusions when the edit distance is null, we take into account patterns' fields positions as the following:

- for every list HLN we compute a new matching value.
- This value depends on the number of exact vertical alignment of fields having the same tag.
- The final matching value is the ratio:

$$RT = \frac{|matching\, fields|}{|fields\, in\, HLN|}$$

where $|X|$ is the number of elements in X.

The higher RT is, the more probable HLN is a PS. In fact, if RT tends to 1, then two possibilities exist:
  - $RT = 1$, HLN is a singleton (this case will be eliminated because it is meaningless for table), or HLN is a perfect table.
  - $RT < 1$, meaning the case of a possible table.
- After processing the whole document, the chosen HLN is the one maximizing RT. PS is then the best HLN candidate. This method can extract tables only when there are at least two table lines in the document.

KWS extraction Key-word structures (KWS) are local arrangements of key-words (KW) like 'road', 'zip-code', 'name', etc for an address. These KW occur frequently in administrative documents and can be in several languages. KW are extracted thanks to a specific software developped by ITESOFT. Its details are outside the aim of this paper. KW can be written in different manners but have always the same meaning. For example, "total", "tot", "total amount" represent the same information but are written differently from an invoice to another. In order to avoid confusions and to be able to propose general cases, KW with the same meaning are given the same KW tag. We use graphs to represent this key-word association (key-words in vertices, and relative spatial relationships on edges). This association maintains the real positions of KW in the document as well as the semantic proximity between them. We preferred using relative positions instead of absolute positions when tagging the edges in order to have a better generalisation of a case. For example when a homogeneous set of documents is processed, it is usual that the absolute positions of a KW changes from a document to another one. However, its relative position to the other KW in the document does not change.

4.3 Document structure extraction

A document structure is a gathering of all its sub-element structures (PS and KWS). We use a graph for its representation. In order to have a harmonious graph representation, useful for future comparisons, we consider all the vertices visible at the same level. This means that the difference between vertices is characterized by edges which are either 'spatial' (left, right, top, bottom) when they designate spatial relationships or 'contain' when they designate a structure component (as between a KWS and each of its KW). This kind of graph representation gives flexibility to CBRDIA as it is just articulated around KWS and PS relative positions. It is also helpful for document comparisons (See 5.1). We preferred the graph representation to the vector representation as the latter does not take into account any position in the document. Moreover, classical vector representations such as the tf-idf technique does not give any information about the position in the document, nor about the relative position with another KW. In addition to all that, [10] [7] show clearly that using the graph representation gives much better results in document classification that the vector based representation. All these reasons lead us to represent KWS and documents in graphs.

4.4 Problem enriching using a set of homogeneous documents

In the previous sections, we have shown how to extract a document problem starting from one single document. This can be sufficient when the extraction is done easily and when the proposed document does not contain any noise (very good OCR results, tables with multiple lines). However, this is not usually the case. In many cases, extracting a document problem without checking whether this problem is representative of the set of documents it comes from can cause many errors in the solving process.

In this paragraph, we show how to extract and to enrich a problem starting from a set of documents. A set of documents means a set of homogeneous documents i.e they are all issued by the same company, they have the same physical structure. The similarity of documents in the same set is high and processing one document can help a lot in processing the remaining documents. In order to help having available and representative problems, we use a whole set of documents in the problem elaboration process. The final problem is representative of all the problems of the processed documents. In other words, the systems extract the problem from each document in the set and adds this problem to the previous ones. As the document problems are graphs, the final problem is a graph representing all the extracted graphs. Such a graph can only be the Minimum Common Supergraph (MCS) of all the extracted graphs. Formally speaking, strating from a set of graphs G_i, $i = 1..n$ an MCS is a graph such that it has a subgraph isomorphism with every graph in G_i. This MCS is very helpful. It represents the whole class of documents and allows a better generalization in both the steps of elaboration and solving.

Bunke [11] introduces the notion of weighted MCS. It is an MCS where the vertices and edges have weights corresponding to their frequencies in the set G_i. These weights can be useful as they allow the distinction between real information and noisy information. A noisy information (a vertex which can be in our application a key-word, a KWS or a PS) is usually characterized by a very low frequency. By using a threshold, we can filter the undesired information.

The redundancy of some information in the problem enriching phase can also help finding and modelling future solutions. For example, the redundancy of the field ”phone number + XX.XX.XX.XX.XX” can be helpful in the solving process. If this redundancy is detected, the solution of the KW ”phone number” is known in advance, and it becomes unnecessary to look for it once the solving phase starts. Similar ideas concerning problem enriching are under study.

4.5 CBRDIA Cases

CBR requires the definition of cases: a problem and its corresponding solution. Let C be a case, C= {P_C, S_C}. According to the problem elaboration step, two cases are possible:

1. Structure case which can be a KWS case or a PS case

- KWS case: where P_KWS is the graph of key-words contained in a structure and S_KWS is the interpretation of each KW. For example, the solution of the KW "street" is the name of the street and the number corresponding to the address (example: 12 Decker street). In this case, KWS solution is the set of KW solutions. S_KWS = {S_KW} where KW is a particular case of KWS containing just one key-word.
- PS case: where P_PS is the pattern (e.g ABBB) representing the table and S_PS is the interpretation of each table column.

2. Document case: P_DC is the document graph and S_DC is the solution of all its structures: S_DC = {S_KWS, S_PS}. P_DC consists in the graph of all the structures of the document. The graph vertices are the structures or the kew-words contained in the structures. Two different edge labels exist. If the edges connect between structure vertices or between KW inside a KWS, then their labels are spatial labels (above, below, right left) representing the relative positions of structures (arrows in full line on figure 7) . Otherwise, their labels are "contain"' type, meaning that a certain information is contained in a structure (arrows in dotted line on figure 7) .

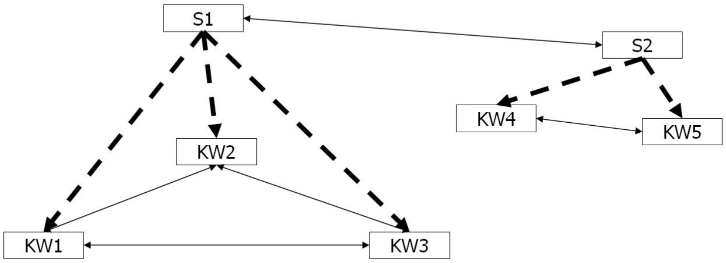


Fig. 7. Example of a document graph. KW1 to KW5 are key-words. S1 and S2 are two KWS.

Figure **??** shows a simple example of a graph of a document problem.

5 Global Solving

5.1 Similar case retrieval

For graph comparison, many measures can be used [10] . However, as we are not only looking for accuracy, but also for a fast processing, we used Lopresti's method called graph probing [12]. It is a fast and accurate technique to compare graphs by measuring their degree of dissimilarity. In his paper, Lopresti applied his method successfully on document graphs containing simple structures of lines and words. In order to compare labeled and directed graphs, two probes PB1 and PB2 have to be measured:

1. PB1 = the frequency of each vertex in the graph;
2. PB2 = the frequency of each vertex' edge structure.

PB1 and PB2 are measured for the studied graph P_DC (G_1) and the Document Database (D_DB) graph (G_2). The probing distance P between the graphs G_1 and G_2 is then:

$$P(G_1, G_2) = (PB1(G_1) - PB1(G_2)) + (PB2(G_1) - PB2(G_2))$$

It has to be noticed that if $P(G_1, G_2)$=0, G_1 and G_2 are not necessarily isomorphic. However, graph probing gives an approximation of the edit distance. We notice here that other types of distances are being studied.

Graph probing is then used to compare graphs and to retrive the nearest case in D_DB.

5.2 Solution adaptation

When a similar case for P_DC (G_1) is found in D_DB (G_2), the adaptation consists first in finding for each structure in G_1 the corresponding structure in G_2. This is achieved by measuring the edit distance or the probing distance between the graphs of structures between G_1 and G_2. As the documents correspond to the same case (meaning they belong to the same set of documents), the system just copies the information about the nature (alphabetic, numerical) and the position (left on the same line, right on the same line, top on the line above) of each solution and looks for similar information in the current document. For example, if the solution corresponding to a KW "total" in G_2 case has the properties "real number + right", the system will look for a real number on the right of "total" on the same line (HL) in the processed document. If an answer exists, then it is proposed as a solution for this KW.

6 Local Solving

If no similar case exists in D_DB, the system builds a solution based on the structures already processed in others documents and stored in a special database S_DB (Structure cases database).

6.1 KWS solving

The solving procedure acts as the following:

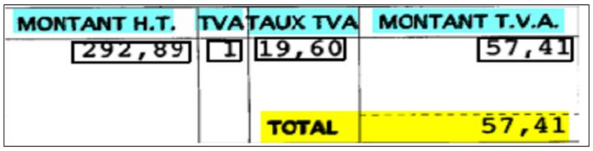
1. For each structure in the document, the nearest structure in the Structure Database (S_DB) is retrieved. P_KWS graph is compared to the KWS cases of S_DB. The solution of the nearest structure is adapted. Graph edit distance is used to find the nearest graphs in S_DB. We have used edit distance between these graphs as we are really looking for graph isomorphism, or at least, sub-graph isomorphism. As S_DB graphs are also small (no more

than 5 vertices per graph in general), it is then better to use a more precise comparison technique than to use a faster but less accurate one like graph probing distance. The cost function used to compute edit distance between graphs has uniform costs for both vertices and edges edit operations as both KW and their relative positions seem to have the same importance in the graph.

2. The nearest structures' solutions are now adapted to the document structures. As the cases in the S_DB have already a correct solution, the adaptation consists in taking the solution of each KW (case of KWS) and trying to find a corresponding solution in the processed document. If a complete solution is found for a structure, then, it can be stored in the S_DB. Otherwise, the following processing has to be done. For KW, some universal knowledge exists and it would be really a waste of time not to take advantage of it. For example, it is usual that the KW "total" is followed by a numerical. This numerical can be a real number or an integer depending on the document but its numerical nature is always valid. Following this logic, a rule basis detailing the general rules associated with key-words was built, in order to complete any partial solution of a KWS. This basis allows completing some missing KWS solutions. It has to be noticed that a rule basis is not able to solve complete cases as it does not take into account the context of the structure, and as the RB knowledge is very general and not related to any concrete case.

   If no solution can be found for a given structure, the system can ask the user to propose one.

The example on figure 8 shows a KWS which nearest KWS in S_DB resolves four out of five KW. By using a rule basis, a complete solution can be found.



MONTANT H.T.	TVA	TAUX TVA	MONTANT T.V.A.
292,89	1	19,60	57,41
		TOTAL	57,41

Fig. 8. A KWS. Only the KW Total is solved by the rule basis.

6.2 PS solving

Each extracted PS (E_PS) is compared with the S_DB cases to retrieve the nearest structure. As PS are represented with strings, their patterns are compared using string edit distance. When a similar PS (C_PS) is found in S_DB, (same pattern, or with a maximum of one transformation), the table columns of E_PS are given the tags of C_PS, unless the rule between C_PS fields can not be applied on EPS. In this case, the system tries to find the rule between EPS fields by trying the rules in other close PS cases (close PS cases with more than one transformation) until a valid rule is found.

If no solution can be found, the user can also here propose one.

A perspective of our work is to use the table headers in order to interpret the table columns. In this way, PS cases could also be considered as KWS cases as we will use the KW found in the headers for the interpretation.

7 Experiments

7.1 The Database

The dataset is composed of 923 documents taken from different clients representing 325 different sets. The test data set is divided in 2 groups :

- the first one contains 100 documents where each one has a similar case in D_DB: this will help us testing the global solving
- the second one contains 823 documents for which no associated case exists in D_DB. Hence, local solving will be applied on these documents.

7.2 Measures

The results are described thanks to three different measures:

$$R = \frac{|right\ solutions\ in\ all\ documents|}{|desired\ solutions\ in\ all\ documents|}.$$

$$R\_KWS = \frac{|right\ KW\ solutions\ inall\ documents|}{|desired\ KW\ solutions\ in\ all\ documents|}.$$

$$R\_PS = \frac{|right\ PS\ fields\ in\ all\ documents|}{|desired\ PS\ fields\ in\ all\ documents|}.$$

A right solution corresponds to a KW's solution or to a field in a PS that has been correctly extracted and interpreted. S_DB contained initially 300 structures. Only 20 of the tested structures have a complete similar case in S_DB. The remaining cases in S_DB are taken from several other documents which are not related to the tested documents. We have chosen to test our system in this way to show its ability to find a solution for a given problem even if it has never been studied before.

7.3 Results

The results are given in table 1.

In global solving, the missing 14.78% correspond to 3.72% system errors and 11.06% OCR errors. In local solving, for KWS, errors are due to:

- 12.15% system errors (bad solution, no solution found, confusion with other solutions),

Table 1. Results of CBRDIA for global solving and local solving

	R	R_KWS	R_PS
Global Solving	85.22%	79.77%	90.10%
Local Solving	74.90%	75.25%	74.80%

- 12.59% of OCR errors.

In local solving, for PS, errors are due to:

- bad detection of table lines (11.32%) (missing lines, no detection of table),
- OCR and segmentation errors (12.67%) (e.g : 12.T instead of 12.7. Two fields are fused together which implies a wrong tag),
- bad solution (1.21%) (fields are given bad solutions).

OCR errors are not just due to the software performance, but they also depend on:

- the quality of documents. In our dataset, we had about 11% of documents of very poor quality (this can be caused by the original quality of the document, or by a bad scanning)
- noisy information such as missing characters

The OCR used in our system is a professional one used by ITESOFT.

The obtained results are very satisfying from an industrial point of view. This system will be used soon in an industrial environment.

A special case of KWS was also tested (addresses). We have tested our system for KWS solving on 30 documents containing addresses. We obtained 78.33% (118/150) of good results (150 being the number of processed KW in these address bocks). We can notice that this special case exists not only in invoice documents, but also in any other administrative documents.

The difference between the results of global solving and those of local solving can be explained as the following:

- In global solving, the system is processing a similar document: it has the knowledge of what it is looking for in the document. The only sources of error can be:
  - a bad tagging (words, fields)
  - a bad PS extraction,
  - a missing KWS or a missing KW,
  - OCR errors.
- However, in local solving, in addition to all the previous sources of errors, we can also add the bad extracted solutions. This can happen when the processed structures have no very close structures in S_DB. This can deteriorate the quality of the proposed solutions.

8 Conclusion and future works

A CBR approach for invoice document analysis and interpretation was proposed in this paper. CBRDIA produces good results even if the documents have never been processed by the system before. This work is still under study in several ways. We are studying the improvement of problem elaboration especially in PS extraction. We are also focusing on S_DB and D_DB indexing in order to reduce the solving time. Finally, solutions' quality is the next step in our work. Enriching S_DB and D_DB requires having high quality solutions; otherwise, a lot of noisy cases can reduce the solving process efficiency. These studies should allow CBRDIA to have better results.

References

1. J. Yuan, L, Xu, and C.Y. Suen. Form items extraction by model matching. In *International Conference on Document Analysis and Recognition*, France, 1991.
2. H. Sako, M.Seki, N. Furukawa, H.Ikeda, and A.Imaizumi. Form reading based on form-type identification and form-data recognition. In *International Conference on Document Analysis and Recognition*, Scotland, 2003.
3. Francesca Cesarini, Enrico Francesconi, Marco Gori, and Giovanni Soda. Analysis and understanding of multi-class invoices. *IJDAR*, 6(2):102–114, 2003.
4. Yolande Belad and Abdel Belad. Morphological tagging approach in document analysis of invoices. In *ICPR*, pages 469–472, 2004.
5. Rosina Weber, Kevin Ashley, and Stefanie Bruninghaus. Textual case-based reasoning. *The Knowledge Engineering Review*, 20(3):255–260, 2006.
6. Petra Perner, Alec Hlot, and Michael Richter. Image processing in case-based reasoning. *The Knowledge Engineering Review*, 20(3):311–314, 2006.
7. Colleen Cunningham, Rosina Weber, Jason M. Proctor, Caleb Fowler, and Michael Murphy. Investigating graphs in textual case-based reasoning. In *ECCBR*, pages 573–586, 2004.
8. William Cheetham and Joe Shultz. Using ensembles of binary case-based reasoners. In *ICCBR*, pages 152–162, 2005.
9. A. Aamodt and E. Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. In *IOS press*, 1994.
10. Adam Schenker, Mark Last, Horst Bunke, and Abraham Kandel. Comparison of distance measures for graph-based clustering of documents. In *GbRPR*, pages 202–213, 2003.
11. H.Bunke, P. Foggia, C. Guidobaldi, and M. Vento. Graph clustering using the weighted minimum common supergraph. In *GbRPR*, pages 235–246, 2003.
12. Daniel P. Lopresti and Gordon T. Wilfong. A fast technique for comparing graph representations with applications to performance evaluation. *IJDAR*, 6(4):219–229, 2003.