Entity Local Structure Graph Matching for Mislabeling Correction

Nihel Kooli, Abdel Belaïd, Aurélie Joseph, Vincent Poulain D 'Andecy

Entity Local Structure Graph Matching for Mislabeling Correction

Nihel Kooli and Abdel Belaïd
LORIA - Université de Lorraine
Campus scientifique - BP 239
54506 Vandoeuvre-lès-Nancy, France
Email: {nihel.kooli,abdel.belaid}@loria.fr

Aurélie Joseph and Vincent Poulain D'Andecy
ITESOFT Groupe - YOOZ
Parc d'Andron, Le Séquoïa
30470 Aimargues, France
Email: {aurelie.joseph,vincent.poulaindandecy}@yooz.fr

***Abstract*—This paper proposes an entity local structure comparison approach based on inexact subgraph matching. The comparison results are used for mislabeling correction in the local structure. The latter represents a set of entity attribute labels which are physically close in a document image. It is modeled by an attributed graph describing the content and presentation features of the labels by the nodes and the geometrical features by the arcs. A local structure graph is matched with a structure model which represents a set of local structure model graphs. The structure model is initially built using a set of well chosen local structures based on a graph clustering algorithm and is then incrementally updated. The subgraph matching adopts a specific cost function that integrates the feature dissimilarities. The matched model graph is used to extract the missed labels, prune the extraneous ones and correct the erroneous label fields in the local structure. The evaluation of the structure comparison approach on 525 local structures extracted from 200 business documents achieves about 90% for recall and 95% for precision. The mislabeling correction rates in these local structures vary between 73% and 100%.**

***Keywords*—entity local structure; subgraph matching; mislabeling correction; structure model; graph clustering;**

I. Introduction

An entity is defined as an existing or real thing, such as a person, a location, an organization, etc. It is contrasted with its attributes. For example, a person has the attributes: name, date of birth, social security number, etc. Extracting entities in documents is equivalent to identifying their attributes. This work lies in the context of entity extraction in document images guided by a predefined database, as reported in [1]. This database describes the entity attributes by its records and informs about their semantics by its fields (columns). The entity attributes are labeled in the document, as proposed in our earlier work [2], using dictionaries extracted from noun fields in the database and regular expression built from fields having a standard from. However, mislabeling may frequently occur as missed labels, extraneous labels or erroneous label fields. The impact is a final wrong detection of an entity. For example, if the social security number is wrongly detected, the person may be not detected or worst confused in the database with an other person having the same name. Missed labels are caused by OCR errors or non-standardized representations between the document and the database (for example, term permutation, abbreviations, typing errors, etc.). Erroneous labels are caused by a confusing syntax of some attributes, for example an amount number labeled by mistake as a zip-code. Finally, the extraneous labels represent noise.

In this paper, we propose to correct the mislabeling by employing the entity label arrangement in the document. We assume that attributes (labels within the page) which are physically close in the document page are more probably related to the same entity than the same ones faraway. These labels build a local structure which is modeled by a labeled graph, called entity local structure graph. In this graph, the nodes represent the content features (syntactic and semantic) and the presentation features of the labels and the arcs represent the geometric relationships between them. This graph is matched with a structure model, which is composed of a set of entity local structure graphs.

Labeled graphs are data structures that offer a significant modeling of complex entities. The labeled nodes describe the entity elements and the labeled arcs represent the relationships between them. Some previous works propose to employ graph modeling for semantic labeling in document images. For example, authors in [3] and [4] propose to model the document layout by graphs for logical labeling and document classification in journals and conference proceedings. This work is based on a model which is learned for each document class. The model represents spatial relationships between the blocks segmented by the OCR. However, this approach is dependent on the document class. The same authors propose in [5] to add the document content study to its layout modeling in order to enhance the labeling in English business letters. However, this approach does not deal with the noisy content given by the OCR.

Labeled graph matching is usually an NP-complete problem and whose objective is to find a way to correspond a graph to another, such that the topology and the node and arc labels are matched. In the domain of pattern recognition, graph matching techniques are generally used to correspond between an observed graph, called candidate graph and a known entity graph, called model graph. There exist two types of graph matching techniques [6] [7] in the literature: exact graph matching and inexact graph matching. Exact graph matching is to find an isomorphism between graphs. Inexact graph matching is the most commonly used and consists on searching for the best match by tolerating the noisy representation and the errors. For this purpose, the most used approach is the Graph Edit Distance (GED) [8]. GED is inspired from the string edit distance and is defined as the weighted sum of the

978-1-5090-1792-8/16 $31.00 © 2016 IEEE
DOI 10.1109/DAS.2016.36

edit operation costs which are needed to transform one graph to the other. However, in the case of labeled graph where the labels are not of nominal type but represent a set of numeral attributes resulting from a feature extraction step, it is generally not possible to find an exact mapping between these labels. To deal with such a problem, a label discretization based solution can be proposed to transform the numerical attributes into nominal labels, but it is clear that the mapping is sensitive to discretization errors. Another solution consists of defining the mapping cost as the sum of distances between the label values and using a cost threshold for the label mapping decision, but it is not easy to find the proper one. This problem was treated in [9] by reformulating the problem in the Integer Linear Program (ILP) formalism and integrating the label mapping cost in the subgraph matching cost. However, this approach tolerates only the label substitution operation and does not deal with the deletion of nodes or arcs.

Our graph matching strategy lies with inexact subgraph matching which consists on searching an inexact matching between a candidate graph and a subgraph of a model graph. This paper treats also labeled graphs where the labels represent a set of numerical attributes (scalar or vectorial). The graph matching problem is solved by the integration of the feature dissimilarities in the cost function. This work is a thorough extension of the work reported in [2] which performs substitution tolerant subgraph matching for the entity structure correction and involves human for the structure model learning. This paper deals with the node and arc deletion in the subgraph matching in order to tolerate the noisy representation of real graphs and extends the label representation features in the local structures. Furthermore, it proposes an unsupervised learning of the structure model based on a graph clustering algorithm.

The remainder of this paper is organized as follows. The entity structure matching approach is detailed in Section II. The experiments on real world data are presented and analyzed in Section III. The paper is concluded in Section IV.

II. Entity structure matching approach

Fig. 1 presents the global schema of the proposed approach. Firstly, a candidate graph is built for each local structure extracted from an input document. Secondly, the candidate graph is compared to the structure model using a subgraph matching method. This model is initially learned from a chosen dataset using a graph clustering algorithm and is then incrementally updated for a document stream. Finally, the geometrical relations in the matched model graph are used to correct the eventual mislabeling in the local structure.

A. Local structure representation

A label in the local structure is defined as:

$$l_i = (c_i, v_i, conf_i) \tag{1}$$

where v_i is the value of the label, c_i is the corresponding field (column) in the database and $conf_i$ is the confidence of labeling l_i given by the n-gram distances [10] which is used to compare between string values in the dictionary and in the document. v_i is represented by a bag of words $v_i = \{t_j\}$. Each input document is defined by $d = \{l_i\}$.

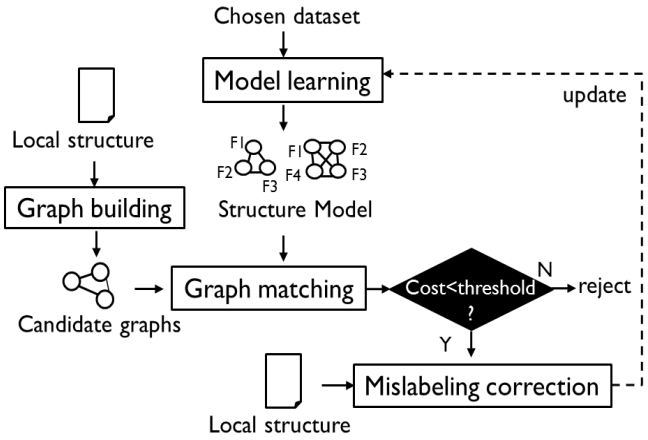


Fig. 1. Global schema of the proposed system

An entity local structure in the document is modeled by a directed complete graph $G = (N, A, \mu, \xi)$ where N is a finite set of nodes that correspond to semantic labels, $A \subseteq N \times N$ is a finite set of arcs that represent geometrical relations between the nodes, $\mu : N \rightarrow L_N$ and $\xi : A \rightarrow L_A$ are two functions assigning a label to a node and an arc respectively. L_N and L_A are discrete sets of labels for the nodes and the arcs respectively. Each arc $a_{ij} \in A$ relating the nodes n_i and n_j is represented by $n_i n_j$. A node n_i that corresponds to a label l_i is defined by:

$$n_i = (c_i, conf_i, nt_i, nl_i, p_i) \tag{2}$$

where nt_i is the number of label terms, nl_i is the number of label lines and p_i is the normalized font size according to the average font of the document (small: 0, medium: $\frac{1}{2}$, large: 1). For an arc a_{ij} relating the nodes n_i and n_j, we define a feature vector describing the spatial relation between these nodes as:

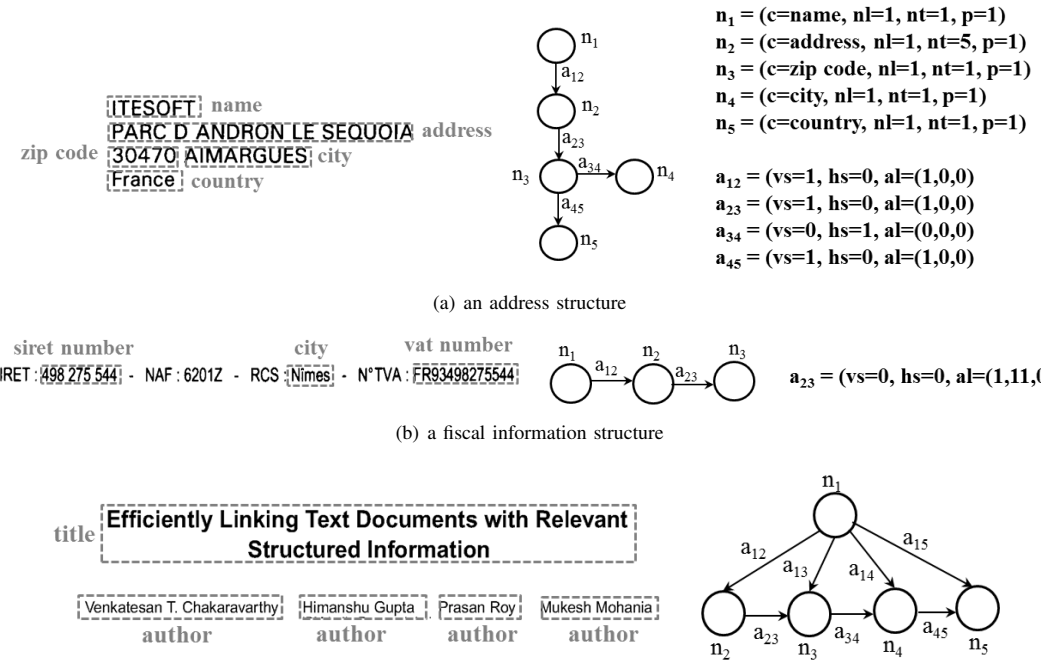
$$a_{ij} = (vs_{ij}, hs_{ij}, al_{ij}) \tag{3}$$

where vs (vertical separation) is the number of lines that separate the labels corresponding to n_i and n_j. hs (horizontal separation) is the distance, in number of characters, that separates the bounding boxes of the labels corresponding to n_i and n_j. vs_{ij} and hs_{ij} are signed to inform about the relative vertical position (above, below) or the relative direction (on the right, on the left). $al_{ij} = (rJust_{ij}, lJust_{ij}, cent_{ij})$ is a vector of three binary values which inform about line alignment (right aligned, left aligned, centered text). Slight variation (lower than 20 pixels) between the line boundaries is tolerated for the alignment.

Fig. 2 shows examples of local structures extracted from real world document images with their representations by attributed graphs (for simplicity, we do not represent all the arcs in the graphs). Fig. 2 (a) and Fig. 2 (b) are extracted from administrative documents. They model an address and a fiscal information structures respectively. Fig. 2 (c) represents a bibliography header structure extracted from a scientific paper.

B. Structure comparison

This can be formulated as a problem of inexact subgraph matching between a candidate graph and a model graph in



(a) an address structure

(b) a fiscal information structure

(c) a bibliography reference header structure

Fig. 2. Examples of local structures

the structure model. The following distortions are considered: arc and node label distortions (variation in their attributes) and extraneous arcs or nodes in the candidate graph. It is not necessary to deal with missing arcs or nodes since we search for a subgraph matching between a candidate graph and a model graph. In order to solve this problem which is generally NP-hard, we use the Branch and bound algorithm [11] which proposes a tree search of the node mapping with backtracking using heuristics. This algorithm is easily adapted to our context since it takes into account the node and arc attributes to provide the matching heuristics. In fact, we employ semantic heuristics (for example, a node labeled by a zip code can not be mapped to a node labeled by a name in an address structure) and physical heuristics (for example, a node labeled by an author can not be placed above a node labeled by a title in a bibliography reference header structure).

Let $t : G \rightarrow M \cup \{\epsilon\}$ be a graph mapping function from a candidate graph $G = (N_G, A_G, \mu_G, \xi_G)$ to a subgraph of a model graph $M = (N_M, A_M, \mu_M, \xi_M)$. To allow the node deletion, it is possible to map a node in G to ϵ. The graph matching cost for the mapping t from G to M is defined as:

$$C(G, M, t) = \frac{\alpha}{|N_G|} \sum_{n \in N_G} C_N(n, t(n)) + \frac{1-\alpha}{|A_G|} \sum_{n \in G} \sum_{n' \in G} C_A(nn', t(n)t(n')) \quad (4)$$

where $\alpha \in [0, 1]$ and $C_N : N_G \times N_M \rightarrow R^+$ and $C_A : A_G \times A_M \rightarrow R^+$ are the mapping cost functions for the nodes and the arcs respectively. Let Δ be the set of the possible mapping functions from G to a subgraph of M. The matching cost is then defined as:

$$C(G, M) = \min_{t \in \Delta} C(G, M, t) \quad (5)$$

Let $F_N = \{nt, nl, p\}$ and $F_A = \{vs, hs, al\}$ be the sets of node and arc features respectively. The node mapping cost function from $n = (c, conf, nt, nl, p)$ to $n' = (c', conf', nt', nl', p')$ is then defined as:

$$C_N(n, n') = \begin{cases} \lambda_{n'}(1 - conf.conf') & \text{if } c = c'; \\ \lambda_{n'} \sum_{f \in F_N} \lambda_f d_f(n, n') & \text{else.} \end{cases} \quad (6)$$

The arc mapping cost function from $a = (vs, hs, al)$ to $a' = (vs', hs', al')$ is defined as:

$$C_A(a, a') = \lambda_{a'} \sum_{f \in F_A} \lambda_f d_f(a, a') \quad (7)$$

where $\lambda_{n'}, \lambda_{a'} \in [0, 1]$ are the weight factors for nodes and arcs in the model graph and $\lambda_f \in [0, 1]$ depends on the feature relevance. Dissimilarities of scalar features are defined as:

$$d_f(a, a') = |f^N - f'^N| \quad \forall f \in F_N \cup F_A \setminus \{al\} \quad (8)$$

f^N is the normalized value of f defined by:

$$f^N = \frac{f - \max(f)}{\max(f) - \min(f)} \quad (9)$$

where $\max(f)$ and $\min(f)$ represent the maximum and the minimum values respectively and are dataset dependent.
The alignment dissimilarity is defined as:

$$d_{al}(a, a') = \begin{cases} 0 & \text{if } al \times al' \neq 0; \\ 1 & \text{else.} \end{cases} \quad (10)$$

Comparing a candidate graph to all model graphs in the structure model is time consuming. We propose then to

filter these graphs using semantic heuristics (for example, the number of nodes having a common label field) and structural heuristics (for example, the number of lines in the local structure or the logical order of the labels in a page line or column). Given a model graph dataset $S_M = \{M_1, ..., M_n\}$ and one candidate graph G, the structure comparison is equivalent to the selection of the model graph M_{match}, where:

$$M_{match} = \arg\min_{M_i \in S_M} C(G, M_i) \quad (11)$$

The model M_{match} is retained if the matching cost does not exceed an empirically fixed threshold.

C. Mislabeling correction

A matched model graph with a candidate is used to correct the three types of mislabeling (missed labels, erroneous label fields and extraneous labels in the local structure modeled by this candidate graph). The extraneous nodes in the model graph may correspond to missed labels in the local structure. The geometrical relations provided by the arcs related to these extraneous nodes are used to localize the missed labels in the document. To validate these label correspondence to the candidate entity in the database, their values are compared to the entity attributes by being less strict in the string distances. Furthermore, the substituted node labels in the model graph are used to correct the label fields in the local structure. Finally, the labels corresponding to deleted nodes in the candidate graph are pruned in the local structure.

Fig. 3 shows an example of mislabeling correction in a structure model. In Fig. 3 (a), the value "3091Z" was labeled as a zip code due to a confusion made by the OCR of the character "Z" with the character "7". Furthermore, the mail was not labeled due to OCR errors and the fax was labeled as a phone since they have the same syntax. The candidate graph built from this local structure is shown in Fig. 3 (b). The matched model graph with this candidate graph using inexact subgraph matching is shown in Fig. 3 (c). This model graph is used to extract the missed mail, to correct the erroneous phone and to prune the extraneous zip code in the local structure represented in Fig. 3 (a).

D. Structure model learning

The structure model is initially learned via the incremental graph clustering algorithm detailed in Algorithm 1. The latter is executed on a dataset chosen to be representative of the corpus. It is an improvement of the Leaders algorithm [12] by adding the incremental update of the centroid to be more representative of the cluster members. This algorithm is adapted to our case since it is a simple incremental clustering algorithm that requires one dataset scan. Besides, experiments on our corpus show that the performances are not affected by the data stream order. During the document stream processing, the structure model is incrementally updated. In fact, each model graph matched with a treated candidate graph is recomputed by considering this new candidate graph.

Let us define now the distance and the centroid mentioned in the algorithm: the distance is the graph matching cost function and the centroid is the group representative graph. The representative graph structure is built using the concept of "Weighted Minimum Common Supergraph (WMCS)" proposed in [13]. The representative graph attributes are learned by fusing the attributes of the member graphs in the cluster. For distances, the sample average is computed. For alignment, the dominant value is used. The representative graph weight factors are set up inversely proportional to the deviation of the attributes in the samples. That is to say, the larger the sample variation of an attribute is, the less discriminant it is and so the lower its weight factor is.

Algorithm 1 Graph clustering

```
select a graph g from the dataset, declare it as an initial centroid and assign it to a new cluster c
for each g ∈ dataset do
    find the nearest centroid
    if distance with the nearest centroid < threshold then
        add p to the member list of c
        recompute the centroid of c
    else
        declare g as a centroid and assign it to a new cluster
    end if
end for
```

III. Experiments

We work on a real-world industrial project in direct collaboration with the ITESOFT[1] company. For tests, we use a dataset of 525 local structures extracted from 200 printed documents. These documents represent industrial invoices or purchase orders. The local structures are categorized into 4 types: postal address structures, contact information structures, fiscal information structures and mixed structures which are a combination of two or more types of structures. For the entity attribute labeling in the documents, a company intern tool called FullText is used. The dictionaries and the regular expressions are built from a database which is composed of a table containing about 230000 records, where each one denotes an entity. This database registers information about enterprises (industrial suppliers and clients) such as their names, addresses, contact numbers, etc.

First, we evaluate the local structure graph matching approach on our corpus. Then, we highlight its benefits for the mislabeling correction. Finally, we evaluate the model learning module based on the graph clustering algorithm.

A. Graph matching results

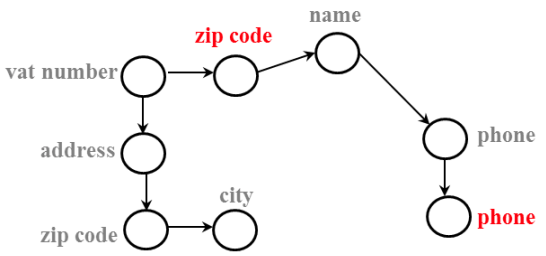
For the graph matching experiments, we use a structure model composed of 41 model graphs (the experiments on the model learning will be presented in section III-C) and a dataset of 525 graph candidates built from the local structure samples. For evaluation, we use a ground truth table that links each candidate graph with its model graph. This ground truth was manually prepared by an expert.

A relevant model graph for a candidate graph is defined as a graph that contains a subgraph corresponding to this candidate graph. The precision and recall are then defined as:

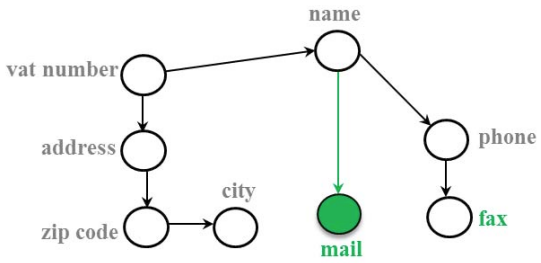
$$recall = \frac{\#\ RMG}{\#\ RG} \quad precision = \frac{\#\ RMG}{\#\ MG} \quad (12)$$

[1] http://www.itesoft.com

(a) An example of local structure with mislabeling



(b) The corresponding candidate graph



(c) The matched model graph

Fig. 3. Mislabeling correction in a local structure using subgraph matching

where RG, MG and RMG represent the relevant graphs, the matched graphs and the relevant matched graphs respectively.

Two strategies are used for the evaluation: always match a candidate graph with the best model graph (top 1) or match it with the best model graph only if their matching cost is below a threshold. The results are illustrated in Table I. It shows that the graph matching algorithm performs well for all structure types. Besides, it shows that the acceptance threshold strategy gives better precision (95.78% compared to 92.57%) and f-measure (93.26% compared to 92.57%) than the best match strategy even if the recall (90.86% compared to 92.57%) is worse. This is explicated by the considerable reduction of the false positives due to the rejection by the threshold. In order to limit the wrong corrections of the local structures, the acceptance threshold strategy is adopted for the remaining experiments.

The graph matching method is compared with existent methods in Table II. Results show that our method outperforms the GED + label discretization and the GED + decision threshold ones using either the substitution tolerant subgraph matching or the substitution and deletion tolerant subgraph matching due to the integration of the feature dissimilarities in the matching cost function. Furthermore, it gives better precision but worse recall than the optimization based method in the case of substitution tolerant subgraph matching. This is explained by the employed heuristics, in the branch and bound algorithm, which discard some "wrong" matching solutions.

B. Mislabeling correction results

The three types of mislabeling correction, using the structure matching, are evaluated separately on a set of labels of variant fields. Missed labels detection in the local structures is evaluated on a set of missed labels using a manually prepared ground truth. TABLE III presents the obtained results. It shows that the identification of all column labels gets high recall (varied between 73.75% and 96%) and precision (varied between 84.28% and 100%) rates. The erroneous label fields correction and the extraneous labels pruning are similarly evaluated and results are presented in TABLE IV and TABLE V respectively. Results are promising for both recall and precision rates. In fact, the recall (precision) rates are varied between 75% and 96% (90.91% and 100%) for erroneous label fields correction and between 77.14% and 97.5% (91.53% and 100%) for extraneous labels pruning. Errors cases are essentially caused by a wrong graph matching solution or a deficiency in verifying the similarity between the detected missed label value and the entity attribute value in the database.

C. Model learning results

For the clustering, a dataset of 290 local structure graphs extracted from 87 documents is used. The number of obtained clusters (model graphs) is 41. To evaluate the clustering algorithm, the Dunn index D is used. It is defined as:

$$D = \frac{\min_{1 \leq i < j \leq n} d(i,j)}{\max_{1 \leq k \leq n} d'(k)} \tag{13}$$

where $d(i,j)$ is the inter-cluster distance between the clusters i and j (the distance between the centroids) and $d(k)$ is the intra-cluster distance of the cluster k (the maximal distance between any element pair in the cluster k).

The Dunn index is evaluated for three variant stream orders of processed graphs in the clustering algorithm. The obtained values are around 0.74 for the three stream orders with the same number of obtained clusters (41 clusters). These results prove that our corpus is independent of the data stream order.

Table VI presents a comparison of the clustering results between the proposed algorithm and the Leaders one. Its shows that the proposed algorithm outperforms the Leaders one with a Dunn index of 0.74 compared to 0.71. This justifies our choice consisting of the incremental update of each cluster centroid.

TABLE I. STRUCTURE GRAPH MATCHING RESULTS

Structure	Best Match			Acceptance threshold		
	Recall (%)	Precision (%)	F-measure (%)	Recall (%)	Precision (%)	F-measure (%)
Postal addresses (135 graphs)	94.07	94.07	94.07	92.59	96.15	94.34
Contact information (100 graphs)	88.00	88.00	88.00	86.00	90.53	88.21
Fiscal information (175 graphs)	95.43	95.43	95.43	94.86	98.81	96.79
Mixed (115 graphs)	90.43	90.43	90.43	86.96	95.24	90.91
Total (525 graphs)	92.57	92.57	92.57	**90.86**	**95.78**	**93.26**

TABLE II. GRAPH MATCHING COMPARISON RESULTS

	Method	Recall (%)	Precision (%)	F-measure (%)
Substitution tolerant subraph matching	GED + Label discretization	86.10	94.76	90.22
	GED + Decision threshold	86.67	95.39	90.82
	Optimizaton [9]	88.00	95.45	91.58
	Our earlier work [2]	87.81	96.04	91.74
Substitution and deletion tolerant subraph matching	GED + Label discretization	87.81	94.66	91.11
	GED + Decision threshold	88.57	94.90	91.63
	Our method	**90.86**	**95.78**	**93.26**

TABLE III. MISSED LABEL IDENTIFICATION RATES

Label field	Number of labels			Recall (%)	Precision (%)
	Missed	Found	Correct		
Name	80	70	59	73.75	84.28
Address	80	73	65	81.25	89.04
Zip code	100	96	96	96	100
City	100	91	89	89	97.80
Country	50	45	41	82	91.11
Phone	70	62	60	85.71	96.77
Fax	70	60	60	85.71	100
mail	50	44	44	88	100
Vat number	50	42	39	78	92.86
Siret number	50	41	37	74	90.24

TABLE IV. ERRONEOUS LABEL FIELDS CORRECTION RATES

Label field	Number of labels			Recall (%)	Precision (%)
	Erroneous	Substituted	Correct		
Zip code	100	96	96	96	100
Name	100	81	75	75	92.59
City	100	91	88	88	96.70
Phone	50	44	40	80	90.91
Fax	50	47	45	90	95.74

TABLE V. EXTRANEOUS LABELS PRUNING RATES

Label field	Number of labels			Recall (%)	Precision (%)
	Extraneous	Pruned	Correct		
Zip code	80	78	78	97.5	100
Name	100	82	78	78	95.12
City	100	87	85	85	97.70
Phone	70	59	54	77.14	91.53
Siret number	50	46	45	90	97.83

TABLE VI. CLUSTERING COMPARISON RESULTS

Method	Best threshold	# Cluster	Dunn index
Leaders [12]	0.1	42	0.71
Our method	**0.1**	**41**	**0.74**

IV. CONCLUSION AND FUTURE WORK

This paper proposes a structure comparison approach for mislabeling correction in documents using a structure model. The comparison is based on a subgraph matching method between a candidate graph (built for each local structure) and a model graph (in the structure model). The evaluation on a dataset of 525 local structures extracted from 200 documents is promising and achieves about 90% for recall and 95% for precision. Furthermore, the method has proven to be effective in correcting the mislabeling (missed labels, erroneous label fields and extraneous labels) in the local structures.

Our future work is to evaluate our approach on other datasets in order to investigate more local structure types such as table lines and bibliographic references. Another interesting work is to enhance the verification of the identified missed labels by combining different similarity measures and using an OCR correction model based on character shape classification Furthermore, we plan to propose a probabilistic score of entity matching in document images based on an estimation of the belonging of their labeled attributes to the local structures.

REFERENCES

[1] N. Kooli and A. Belaïd, "Entity Matching in OCRed Documents with Structured Databases", in International Conference on Pattern Recognition Applications and Methods, 2015, pp. 165–172.

[2] N. Kooli and A. Belaïd, "Semantic Label and Structure Model based Approach for Entity Recognition in Database Context", in International Conference on Document Analysis and Recognition, 2015, pp. 301–305.

[3] J. Liang and D. S. Doermann, "Logical labeling of document images using layout graph matching with adaptive learning", in Document Analysis System, 2002, pp. 224–235.

[4] J. Liang, D. S. Doermann, M. Y. Ma and J. K. Guo, "Page Classification through Logical Labelling", in International Conference on Pattern Recognition, 2002, pp. 477–480.

[5] J. Liang and D. S. Doermann, "Content features for logical document labeling", in Document Recognition and Retrieval, 2003, pp. 189–196.

[6] D. Conte, P. Foggia, C. Sansone and M. Vento, "Thirty years of graph matching in pattern recognition", International Journal of Pattern Recognition and Artificial Intelligence, 2004, 18(3): pp. 265–298.

[7] P. Foggia, G. Percannella and M. Vento, "Graph Matching and Learning in Pattern Recognition in the last 10 years", International Journal of Pattern Recognition and Artificial Intelligence, 2014, 28(1): pp. 1450001-1–1450001-40.

[8] A. Sanfeliu, KS. Fu, "A distance measure between attributed relational graphs for pattern recognition", IEEE Transactions on Systems Man and Cybernetics, 1983, 13(3): pp. 353-362.

[9] P. Le Bodic, P. Hroux, S. Adam and Y. Lecourtier, "An Integer Linear Program for Substitution Tolerant Subgraph Isomorphism and its Use for Symbol Spotting in Technical Drawings", Pattern Recognition, 2012, 45(12): pp. 4214–4224.

[10] G. Kondrak, "N-Gram Similarity and Distance", in International conference on String Processing and Information Retrieval, 2005, pp. 115–126.

[11] A. K. C. Wong, M. You and S. C. Chan, "An algorithm for graph optimal monomorphism", IEEE Transactions on System Man and Cybernetics, 1990, 20(3): pp. 628–638.

[12] P.A. Vijaya, M. Narasimha Murty and D.K. Subramanian, "Leaders-Subleaders: An efficient hierarchical clustering algorithm for large data sets", in Pattern Recognition Letters, 2004, 25(4): pp. 505–513.

[13] H. Bunke, P. Foggia, C. Guidobaldi and M. Vento, Graph clustering using the weighted minimum common supergraph, in International Conference on Graph based Representations in Pattern Recognition, 2003, pp. 235–246.