Inexact graph matching for entity recognition in OCRed documents

Nihel Kooli, Abdel Belaid

To cite this version:

Nihel Kooli, Abdel Belaid. Inexact graph matching for entity recognition in OCRed documents. ICPR, Dec 2016, Mexico, Mexico. pp.4071 - 4076, 10.1109/ICPR.2016.7900271 . hal-01515412

HAL Id: hal-01515412

https://hal.inria.fr/hal-01515412

Submitted on 27 Apr 2017

Inexact Graph Matching for Entity Recognition in OCRed Documents

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

Abstract—**This paper proposes an entity recognition system in image documents recognized by OCR. The system is based on a graph matching technique and is guided by a database describing the entities in its records. The input of the system is a document which is labeled by the entity attributes. A first grouping of those labels based on a function score leads to a selected set of candidate entities. The entity labels which are locally close are modeled by a structure graph. This graph is matched with model graphs learned for this purpose. The graph matching technique relies on a specific cost function that integrates the feature dissimilarities. The matching results are exploited to correct the mislabeling errors and then validate the entity recognition task. The system evaluation on three datasets which treat different kind of entities shows a variation between 88.3% and 95% for recall and 94.3% and 95.7% for precision.**

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

I. Introduction

An entity is a homogeneous group of attributes such as an enterprise in a business form or some meta-data representing the title, the authors and their affiliation in a scientific paper. It is characterized with its attributes. Entity recognition in documents is the task of identifying the entity attributes. It adds a wealth of knowledge to help us understand the document content. Recognizing entities is an important task in enterprises and constitutes an essential component of data integration research community. The task is not easy, especially when dealing with OCRed documents since there are numerous problems to solve. Most prominently, the OCR alters the document structure and introduces errors in the text. Furthermore, attribute locations in the physical structure of documents vary. This requires high-level knowledge to identify them.

In the literature, the entity recognition approaches can be classified into three categories: context oriented approaches, data oriented approaches and structure oriented approaches.

The first ones rely on contextual and linguistic rules for labeling the entity elements in the text. They require predefined rules and are domain and language dependent. An example of these approaches was proposed in [1] to extract events in online newspapers.

Secondly, the data oriented approaches are based domain- and language-dependent. on an annotated corpus and tend to treat the problem as a classification one, or on a predefined database and try to match the entity definition in the database and its representation in the document by searching for the common terms. An example of the first strategy of approaches is described in [2] which uses a Bayesian model to classify entity attributes based on contextual and intrinsic features. In second strategy context, authors in [3] provide EROCS algorithm to identify entities embedded in document segments (few consecutive sentences). EROCS uses a score, defined for an entity with respect to a segment, which considers the frequency of the common terms in the segment and their importance in the database. This work treats web documents, so it considers the text as a sequence of lines and uses strict comparison between terms. A modified version, called M-EROCS, that treats OCRed documents was proposed in our earlier work in [4]. It identifies attributes in contiguous blocks given by the OCR and tolerates content errors in the comparison using the Edit Distance. However, it does not solve the problem of under-segmentation since it assumes that the terms in each block belong to a single entity. Also, it does not assemble non-contiguous parts of the entity.

Finally, the structure oriented approaches propose to benefit from the physical and/or logical structure of the document for the recognition. For example, the authors in [5], [6] use graphs to model the document layout for logical labeling. This model is proposed at document level and represents spatial relationships between blocks segmented by OCR. However, this approach is dependent on document class. A recent approach, proposed in [7], aims to identify table rows content using a set of text field patterns selected by a user. This approach relies on a graph isomorphism technique between a pattern graph and a model graph. The graphs represent the field names and the relative position between them. This approach treats only linear structure and is dependent on the user given patterns.

The proposed entity recognition approach, called G-ELSE (Graph matching for Entity Local Structure Extraction), is a combination of the three types of approaches described previously. It lies in the context of entity recognition guided by a predefined database describing each entity attributes by its records. Entity attributes are labeled in the document using their syntactic information in the database, as detailed in [8]. The entity labels which are locally close are represented by a structure graph which is then matched with a learned structure model. The matching results are then used to correct the mislabeling and validate the entity recognition.

Our graph matching strategy consists of an inexact graph matching technique which looks for the best match by tolerating noisy representation and errors. Several approaches have been proposed in the literature for graph matching in document analysis as described in [9]–[11]. The most commonly used approach is the Graph Edit Distance (GED) as reported in [12]. GED is defined as the weighted sum of the edit operation costs 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, it is generally not possible to find an exact mapping between those labels. To deal with such problems, label discretization based solution can be used to transform the numerical attributes into nominal labels, but such mapping is very 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 threshold. This problem was treated in [13] by reformulating the problem in the Integer Linear Program (ILP) formalism and integrating the label mapping cost into the graph matching cost. However, this approach only takes into account the label substitution operation and does not treat the deletion of nodes or arcs.

This work is a thorough extension of the work reported in [4] which performs substitution tolerant graph matching for the entity structure correction and involves human for the structure model learning. The mislabeling correction proposal was detailed in our previous work [14]. In this paper, we add the node and arc insertion and deletion in the graph matching in order to tolerate the noisy representation of our real graphs. We also propose an unsupervised learning of the structure model based on a clustering algorithm. Moreover, we integrate probability estimation of the labels belonging to the entities based on their structure for making the recognition decision. We also extend the experiments by validating the system on two other datasets.

The remainder of this paper is organized as follows. The entity recognition approach is detailed in Section II. Then, experiments on three real world datasets are presented in Section III. Section IV concludes.

II. Proposed approach: G-ELSE

Fig. 1 presents the global schema of G-ELSE. Firstly, the document labels are grouped to select candidate entities from the database. Secondly, the labels of the same entity which are physically close are used to build a local structure. The latter is modeled by an attributed graph, called local structure graph. This graph is matched to a structure model. The structure model is itself composed of a set of local structures called model graphs. The geometrical relations in the matched model graph are then used to correct the eventual mislabeling in the local structure. The structure model is initially learned from a chosen examples of the studied dataset based on a graph clustering algorithm. It is then progressively updated during the recognition process. Finally the entity candidates are validated or excluded.

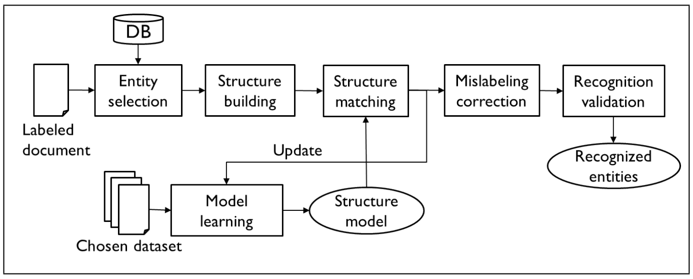


Fig. 1. Global schema of G-ELSE

A. Entity selection

A label is defined as: $l_i = (c_i, conf_i, v_i, sim_i)$, where c_i is the corresponding column (field) in the database, $conf_i$ is the confidence of labeling l_i as a field c_i, $v_i = \{t_j\}$ is the value of the label represented by a bag of words and sim_i is the Levenshtein distance between v_i and $e.c_i$ where $e.c_i$ is the attribute of the entity e corresponding to the field c_i.
Each input document is represented by a set of labels $d = \{l_i\}$.

After labeling, labels are grouped into candidate entities. The grouping is made possible using a score defined in (1).

$$score(e,d) = \sum_{l_i \in F(e,d)} P(l_i \in e|e).conf_i.sim_i \qquad (1)$$

where e is an entity, $P(l_i \in e|e)$ is the probability that the label l_i belongs to a known entity e (initialized to 1 at this phase), $F(e,d)$ is the set of labels that belong to d and contained as well in the entity e, i.e. $l_i \in F(e,d) \equiv (l_i \in d$ and $v_i \simeq e.c_i)$ where $v_i \simeq e.c_i$ means that v_i and $e.c_i$ are considered similar according to the string distance of value sim_i.

The set of labels d is then matched with an entity e_m when: $e_m = \arg\max_{e \in E} score(e,d)$. A rejection threshold T is defined for $score(e_m,d)$ to deduce the set of candidate entities for the document. This threshold is empirically fixed.

B. Structure building

An entity local structure is modeled by an attributed graph $G = (N, A, \mu, \xi)$ where N is a finite set of nodes corresponding to field labels, $A \subseteq N \times N$ is a finite set of arcs representing the 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 $a_{ij} \in A$ linking the nodes n_i and n_j is represented by $n_i n_j$.

A node n_i corresponding to a label l_i is defined in (2).

$$n_i = (c_i, conf_i, nt_i, nl_i, p_i) \qquad (2)$$

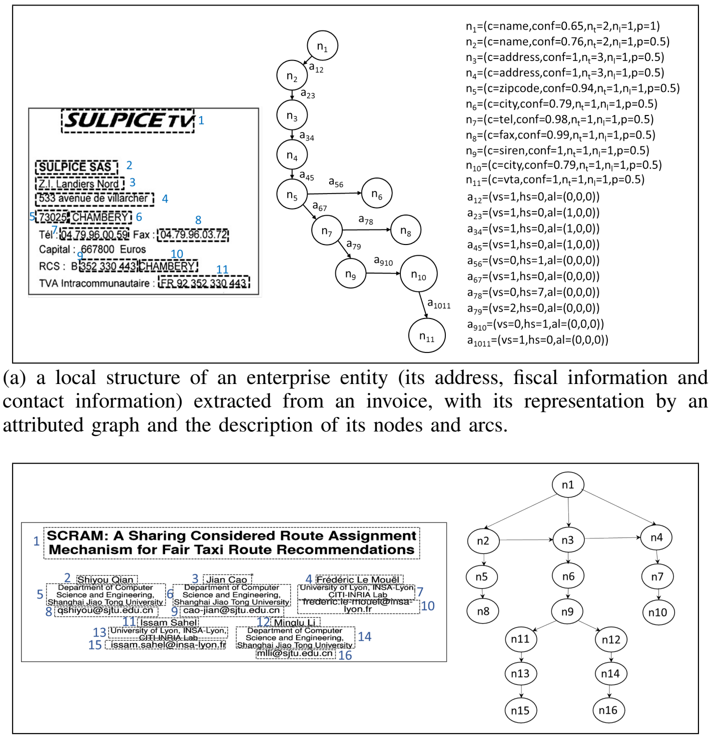
where nt_i is the number of terms, nl_i is the number of lines and p_i is the normalized font size according to the average font in document (small: 0, medium: $\frac{1}{2}$, large: 1).

For an arc a_{ij}, we define a feature vector describing the spatial relationships between the labels in (3).

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

where vs_{ij} (vertical separation) is the number of lines that separate the labels corresponding to n_i and n_j. hs_{ij} (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 values to inform about the relative vertical position (”above” or ”below”) or the relative direction (”on the right” or ”on the left”). $al_{ij} = (rJust_{ij}, lJust_{ij}, cent_{ij})$ is a vector of three binary values informing about line alignment (”right aligned” or ”left aligned” or ”centered”). Slight variation (≤ 20 pixels) between the line boundaries is tolerated for the alignment.

Fig. 2 shows examples of local structures extracted from real world document images. Fig. 2 (a) shows a local structure of an enterprise entity with its representation by an attributed graph and the description of its nodes and arcs. Fig. 2 (b) shows a local structure of a meta-data of a scientific paper entity with its representation by an attributed graph. For simplicity, we do not represent all the arcs in the graphs.


SULPICE TV
1
SULPICE SAS
2
Z.I. Landiers Nord
3
533 avenue de villarcher
4
5
73025
CHAMBERY
6
Tél :
7
04.79.96.00.59
Fax :
8
04.79.96.03.72
Capital : 667800 Euros
RCS : B
9
352 330 443
10
CHAMBERY
TVA Intracommunautaire :
11
FR 92 352 330 443
n1
n2
n3
n4
n5
n6
n7
n8
n9
n10
n11
a12
a23
a34
a45
a56
a67
a78
a79
a910
a1011
n1=(c=name,conf=0.65,nt=2,nl=1,p=1)
n2=(c=name,conf=0.76,nt=2,nl=1,p=0.5)
n3=(c=address,conf=1,nt=3,nl=1,p=0.5)
n4=(c=address,conf=1,nt=3,nl=1,p=0.5)
n5=(c=zipcode,conf=0.94,nt=1,nl=1,p=0.5)
n6=(c=city,conf=0.79,nt=1,nl=1,p=0.5)
n7=(c=tel,conf=0.98,nt=1,nl=1,p=0.5)
n8=(c=fax,conf=0.99,nt=1,nl=1,p=0.5)
n9=(c=siren,conf=1,nt=1,nl=1,p=0.5)
n10=(c=city,conf=0.79,nt=1,nl=1,p=0.5)
n11=(c=vta,conf=1,nt=1,nl=1,p=0.5)
a12=(vs=1,hs=0,al=(0,0,0))
a23=(vs=1,hs=0,al=(1,0,0))
a34=(vs=1,hs=0,al=(1,0,0))
a45=(vs=1,hs=0,al=(1,0,0))
a56=(vs=0,hs=1,al=(0,0,0))
a67=(vs=1,hs=0,al=(0,0,0))
a78=(vs=0,hs=7,al=(0,0,0))
a79=(vs=2,hs=0,al=(0,0,0))
a910=(vs=0,hs=1,al=(0,0,0))
a1011=(vs=1,hs=0,al=(0,0,0))
1
SCRAM: A Sharing Considered Route Assignment Mechanism for Fair Taxi Route Recommendations
2
Shiyou Qian
3
Jian Cao
4
Frédéric Le Mouël
5
Department of Computer Science and Engineering, Shanghai Jiao Tong University
6
Department of Computer Science and Engineering, Shanghai Jiao Tong University
7
University of Lyon, INSA-Lyon CITI-INRIA Lab
8
qshiyou@sjtu.edu.cn
9
cao-jian@sjtu.edu.cn
10
frederic.le-mouel@insa-lyon.fr
11
Issam Sahel
12
Minglu Li
13
University of Lyon, INSA-Lyon, CITI-INRIA Lab
14
Department of Computer Science and Engineering, Shanghai Jiao Tong University
15
issam.sahel@insa-lyon.fr
16
mlli@sjtu.edu.cn
n1
n2
n3
n4
n5
n6
n7
n8
n9
n10
n11
n12
n13
n14
n15
n16

(a) a local structure of an enterprise entity (its address, fiscal information and contact information) extracted from an invoice, with its representation by an attributed graph and the description of its nodes and arcs.

(b) a local structure of a meta-data of a scientific article entity (its title, authors and affiliations), extracted from a scientific paper with its representation by an attributed graph

Fig. 2. Local structure examples

C. Structure matching

The local structure graph is compared with the structure model. This can be formulated as a problem of inexact graph matching between a candidate graph and a model graph in the structure model. For matching, the arc and node label distortions (i.e. variations in their attributes) are considered. Also, extraneous and missed arcs or nodes in the candidate graph are taken into account. In order to solve this problem which is generally NP-hard, we use the branch and bound algorithm, [15], 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 syntactic/semantic and physical heuristics. Some heuristics are generic, such as:

- a label which corresponds to an alphabetic field (for example, name, city, title, etc.) can not be mapped to the one that corresponds to numeric or alphanumeric field (for example, zip code, phone number, date, etc.);
- paths that match more than N labels of uncommon fields are eliminated, where N is empirically fixed (for example, $N = 3$ for enterprise entities).

We also employ some dataset specific heuristics, such as :

- paths present on the right of a postal code other than a city are eliminated;
- an author can not be present above a title in a bibliography reference header structure.

Let $\delta : G \to M \cup \{\epsilon\}$ be a graph mapping function from a candidate graph $G = (N_G, A_G, \mu_G, \xi_G)$ to 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 δ from G to M is shown in (4).

$$C(G, M, \delta) = \frac{\alpha}{|N_G|} \sum_{n \in N_G} C_N(n, \delta(n)) + \frac{1-\alpha}{|A_G|} \sum_{n \in N_G} \sum_{n' \in N_G} C_A(nn', \delta(n)\delta(n')) \quad (4)$$

where $\alpha \in [0, 1]$ and $C_N : N_G \times N_M \to R^+$ and $C_A : A_G \times A_M \to R^+$ are the mapping cost functions for the nodes and arcs respectively. Let Δ be the set of the possible mapping functions from G to M. The matching cost is defined in (5).

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

Let $F_N = \{nt, nl, p\}$ and $F_A = \{vs, hs, al\}$ be two sets of node and arc features respectively. The node mapping cost function from $n_1 = (c_1, conf_1, nt_1, nl_1, p_1) \in N_G$ to $n_2 = (c_2, conf_2, nt_2, nl_2, p_2) \in N_M$ is shown in (6).

$$C_N(n_1, n_2) = \begin{cases} \lambda_{n_2}(1 - conf_1.conf_2) & \text{if } c_1 = c_2; \\ \lambda_{n_2} \sum_{f \in F_N} \lambda_f d_f(f_1, f_2) & \text{else.} \end{cases} \quad (6)$$

The arc mapping cost function from $a_1 = (vs_1, hs_1, al_1) \in A_G$ to $a_2 = (vs_2, hs_2, al_2) \in A_M$ is defined in (7).

$$C_A(a_1, a_2) = \lambda_{a_2} \sum_{f \in F_A} \lambda_f d_f(f_1, f_2) \quad (7)$$

where $\lambda_{n_2}, \lambda_{a_2} \in [0, 1]$ are the weight factors for n_2, a_2, $\lambda_f \in [0, 1]$ depends on the feature relevance and f_1, f_2 are the values corresponding to the feature f for n_1, n_2 or a_1, a_2. Dissimilarities of scalar features are defined in (8).

$$d_f(f_1, f_2) = |f_1^N - f_2^N| \quad \forall f \in F_N \cup F_A \setminus \{al\} \quad (8)$$

f_1^N resp. f_2^N is the normalized value of f_1 resp. f_2. f_1^N (equivalently f_2^N) is defined in (9).

$$f_1^N = \frac{f_1 - \max(f_1)}{\max(f_1) - \min(f_1)} \quad (9)$$

where $\max(f_1)$ and $\min(f_1)$ represent the maximum and the minimum values respectively and are dataset dependent. The alignment dissimilarity is defined in (10).

$$d_{al}(al_1, al_2) = \begin{cases} 0 & \text{if } al_1 \times al_2 \neq 0; \\ 1 & \text{else.} \end{cases} \quad (10)$$

Comparing a candidate graph to all model graphs in the structure model is time consuming. We therefore propose to filter these graphs using semantic and structural heuristics:

- the number of lines in the local structure;
- the number of nodes having a common label field;
- the logical order of the labels in a page line or column.

Given a set of model graphs $S_M = \{M_1, ..., M_n\}$ and one candidate graph G, the matching is equivalent to the selection of the model graph M_{match}, as in (11).

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

The selected model graph is retained if its graph matching cost is below an empirically fixed threshold.

D. Mislabeling correction

A matched model graph with a candidate graph is used to correct the following three types of mislabeling in the local structure: missed labels, erroneous label fields and extraneous labels. These errors are corrected using the deletion, substitution and insertion operations as follows:

- 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 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 than in the labeling step.
- The substituted node labels in the model graph are used to correct the label fields in the local structure.
- the labels corresponding to deleted nodes in the candidate graph are pruned in the local structure.

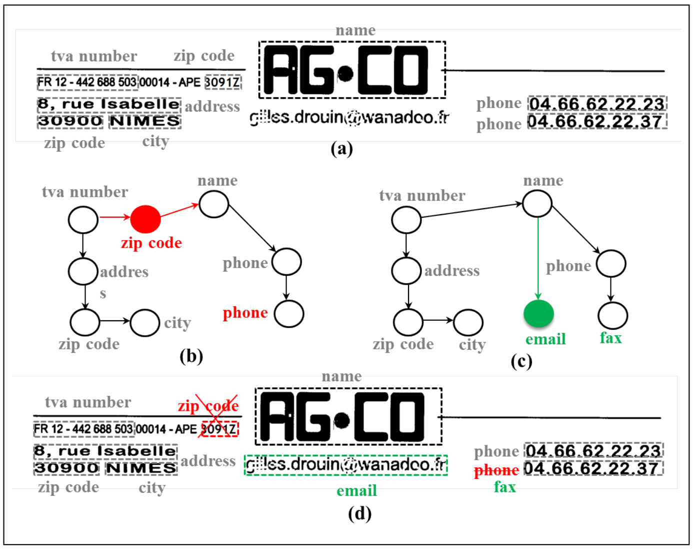


Fig. 3. An example showing mislabeling correction in an enterprise local structure, (a) the erroneous structure, (b) the corresponding candidate graph, (c) the matched model graph, (d) the corrected structure

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 number was labeled as a phone number 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 graph matching is shown in Fig. 3 (c). This model graph is used to extract the missed mail, to correct the erroneous phone number and to prune the extraneous zip code in the local structure as showed in Fig. 3 (d).

E. Recognition validation

Once the entity local structures are corrected, the recognition of such entities is validated similarly to the entity selection module, where the score in (1) is redefined to consider the structure matching results. In fact, $P(l_i \in e|e)$ becomes dependent on the label membership to a local structure of e, denoted $S(e)$, and the graph matching cost between the candidate graph G that represents this local structure and its corresponding model graph M in the structure model. This is expressed by (12).

$$P(l_i \in e|e) = \begin{cases} 1 - C(G, M) & \text{if } l_i \in S(e); \\ 1 - P_s(c_i) & \text{else.} \end{cases} \quad (12)$$

where $P_s(c_i)$ is the probability of any label having a field c_i to belong to a local structure. This probability is statistically computed over the corpus.

In the following, we define the process of learning the structure model employed in the structure matching module.

F. Structure model learning

The structure model is learned using the incremental graph clustering algorithm detailed in Algorithm 1. The proposed algorithm is executed on a dataset chosen to be representative of the corpus. It is an improvement of the Leaders algorithm, proposed in [16], 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 independent of the data stream order.

The distance mentioned in the algorithm is the graph matching cost function. The centroid is the representative graph of the cluster members. The structure of this graph is built using the concept of “Weighted Minimum Common Supergraph (WMCS)” proposed in [17]. The attributes of the representative graph are learned by fusing the attributes of the members. For distances, the sample average is computed. For alignment, the dominant value is used. The weight factors of the representative graph 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.

During the document stream processing, the structure model is updated similarly to the initial learning. Indeed, if the candidate graph corresponds to an existent model graph, the latter is recomputed by considering this new candidate graph. Otherwise, an embryo model graph is initialized by the candidate graph. It becomes a mature graph when its occurrence exceeds an threshold (empirically fixed to 10).

III. Experiments

For experiments, we used three datasets where the two first datasets are provided by the ITESOFT[1] company and the third

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

TABLE I. ENTITY RECOGNITION COMPARISON RESULTS

Method	Enterprise dataset				Material dataset				Scientific paper dataset			
	R (%)	P (%)	F (%)	Runtime (s/doc)	R (%)	P (%)	F (%)	Runtime (s/doc)	R (%)	P (%)	F (%)	Runtime (s/doc)
EROCS [3]	67.68	54.10	60.14	69	76.19	77.42	76.80	37	67.46	77.27	72.03	42
M-EROCS [8]	73.38	69.68	71.48	4.4	79.37	78.74	79.05	3.5	71.42	90.00	79.64	3
Entity selection	86.69	56.86	68.67	0.84	77.62	74.31	75.93	0.85	80.51	70.41	75.13	0.90
G-ELSE	**95.06**	**94.70**	**94.88**	1.4	**92.06**	**94.30**	**93.17**	1.2	**88.49**	**95.30**	**91.77**	1.2

TABLE II. CLUSTERING COMPARISON RESULTS

Method	Enterprise dataset			Material dataset			Scientific paper dataset		
	Best threshold	# Cluster	Dunn	Best threshold	# Cluster	Dunn	Best threshold	# Cluster	Dunn
Leaders [16]	0.1	42	0.71	0.15	10	0.74	0.11	31	0.78
Our method	0.1	41	0.74	0.15	11	0.77	0.12	28	0.80

```
Algorithm 1 Graph dataset clustering
Require: graph dataset : $G$, threshold : $T$
Ensure: cluster set: $CL$
$CL = \emptyset$    ▷ a cluster $c \in CL$ is defined by its centroid $c.centoid$
select a graph $g_1 \in G$
$CL = \{c_1\}$ where $c_1.centoid = g_1$
for each $g_i \in G$; $i = 2 \to |G|$ do
    $c_{min} = \arg\min_{c \in CL} C(g_i, c.centroid)$    ▷ $C(g_i, c.centroid)$ is the graph matching cost between $g_i$ and $c.centroid$
    if $C(g_i, c_{min}.centroid) < T$ then
        recompute $c_{min}.centroid$
    else
        $CL = CL \cup \{c\}$ where $c.centoid = g_i$
    end if
end for
```

TABLE III. GRAPH MATCHING RESULTS IN ENTERPRISE DATASET

Structure	R (%)	P (%)	F (%)
Postal addresses (135 graphs)	92.59	96.15	94.34
Contact information (100 graphs)	86.00	90.53	88.21
Fiscal information (175 graphs)	94.86	98.81	96.79
Mixed (115 graphs)	86.96	95.24	90.91
Total (525 graphs)	90.86	95.78	93.26

TABLE IV. GRAPH MATCHING RESULTS IN MATERIAL DATASET

Structure	R (%)	P (%)	F (%)
Table row (320 graphs)	89.69	95.67	92.58

TABLE V. GRAPH MATCHING RESULTS IN SCIENTIFIC PAPER DATASET

Structure	R (%)	P (%)	F (%)
Reference information (50 graphs)	96.00	90.56	93.20
Edition information (40 graphs)	90.00	90.00	90.00
Author information(40 graphs)	82.50	82.93	82.71
Mixed (70 graphs)	85.71	92.3	88.88
Total (200 graphs)	88.50	89.53	89.01

one is extracted from public sites[2]:

1) Enterprise dataset:
   - 278 documents: invoices and purchase orders;
   - a database of 230000 records, described by fields: name, address, phone, mail, tva, etc.;
   - 526 entities: enterprises (clients + suppliers).
2) Material dataset:
   - 200 documents: invoices;
   - a database of 86600 records, described by fields: description, serial, amount, price, etc.;
   - 630 entities: materials.
3) Scientific paper dataset:
   - 252 documents: first pages of journals, conferences, thesis manuscripts, posters;
   - a database of 415500 records, described by fields: title, authors, affiliations, journal, etc.;
   - 252 entities: meta-data of scientific papers.

For evaluation, we used ground truth tables that link documents with their contained entity identifiers in the databases. These tables were manually prepared. For the entity attribute labeling in the documents, a company internal tool called FullText was used.

First, we evaluate the proposed approach by focusing on the ultimate goal: the entity recognition. Next, a finer evaluation is made by considering each of the two main modules of the approach: the model learning and the graph matching.

A. *Entity recognition evaluation*

For evaluation, recall (R), precision (P) and f-measure (F) are computed as in (13).

$$R = \frac{\#\ RME}{\#\ RE} \quad P = \frac{\#\ RME}{\#\ ME} \quad F = \frac{2.P.R}{P+R} \tag{13}$$

where RE represent the relevant entities, such as a relevant entity for a document is defined as an entity present in the document and referring to some record in the database, ME represent the matched entities and RME represent the relevant matched entities.

The entity recognition results are shown in Table I. This table shows that the entity selection process gives low recall, precision and f-measure caused by the association of labels to the entities without considering their structural information in the document page. These percentages are significantly improved by the mislabeling correction processes. The causes of entity recognition errors are enumerated in Tables VI.

TABLE VI. CAUSES OF ENTITY RECOGNITION ERRORS

Corpus Causes	Enterprises	Materials	Scientific papers
Structure segmentation errors (%)	1.52	0.95	2.78
Structure matching errors (%)	0.38	0.48	0.79
Unrecoverable OCR errors (%)	1.90	1.90	2.38
Non-standardization (%)	0.19	4.29	3.57
Incomplete entities (%)	0.95	0.32	1.98

We compared G-ELSE to two other works in the state of the art. Table I presents the obtained results. It shows a significant increase in precision and recall compared to the

[2] https://hal.archives-ouvertes.fr/;http://www.istex.fr/;http://dblp.uni-trier.de/

evaluation of EROCS and M-EROCS methods on our datasets. Furthermore, we notice an important decrease in the run time due to the labeling and the entity filtering steps.

B. Model learning evaluation

For the clustering, we used:

1) Enterprise: 290 graphs taken from 87 documents.
2) Material: 210 graphs taken from 45 documents.
3) Scientific paper: 200 graphs taken from 70 documents.

To evaluate the clustering algorithm, the Dunn index D was used. It is defined in (14).

$$D = \frac{\min_{1 \leq i < j \leq n} d(i,j)}{\max_{1 \leq k \leq n} d'(k)} \tag{14}$$

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).

Table II 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 for all datasets. This is due to our choice which consists of using the incremental update technique for each cluster centroid.

C. Graph matching evaluation

For the graph matching experiments, we used:

1) Enterprise: 41 model graphs, 525 candidate graphs.
2) Material: 11 model graphs, 320 candidate graphs.
3) Scientific paper: 28 model graphs, 200 candidate graphs.

A relevant model graph for a candidate graph is defined as a graph that corresponds to this candidate graph. The recall (R), precision (P) and f-measure (F) are then defined in (15).

$$R = \frac{\#\ RMG}{\#\ RG} \quad P = \frac{\#\ RMG}{\#\ MG} \quad F = \frac{2.P.R}{P+R} \tag{15}$$

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

For evaluation, we used ground truth tables that link each candidate graph with its model graph. These tables were manually prepared. The results on the three datasets are illustrated in Table III, Table IV and Table V. They show that the proposed method performs well for all structure types. This is explained by the employed heuristics, in the branch and bound algorithm, which guides the matching solutions retrieval.

IV. Conclusion and future work

This paper proposes an entity recognition approach in documents recognized by OCR driven by a structure model. The inexact graph matching technique between the candidate graphs (built for local entity labels) and the model graphs (in the structure model) has proved to be effective in improving the recognition process. The results on three datasets treating different kinds of entities are promising. Our future work is to study the local structure position in the physical or/and logical document structure to guide the search. Moreover, we plan to investigate several string matching combinations for the attribute matching in order to enhance the recognition.

V. Acknowledgment

The authors would like to thank their collaborator ITESOFT for providing real word data for tests.

References

[1] R. Hashemi, C. W. Ford, T. Vamprooyan, and J. Talburt, "Extraction of features with unstructured representation from html documents," in *IADIS International Conference on WWW/Internet*, 2002, pp. 47–53.

[2] O. O. Isaac and T. John R., "A data-intensive approach to named entity recognition combining contextual and intrinsic indicators," *International Journal of Business Intelligence Research*, vol. 3, no. 1, pp. 55–71, 2012.

[3] V. T. Chakaravarthy, H. Gupta, P. Roy, and M. Mohania, "Efficiently linking text documents with relevant structured information," in *Proceedings of the International Conference on Very Large Data Bases*, 2006, pp. 667–678.

[4] N. Kooli and A. Belaïd, "Semantic label and structure model based approach for entity recognition in database context," in *Proceedings of the International Conference on Document Analysis and Recognition*, 2015, pp. 301–305.

[5] J. Liang and D. S. Doermann, "Logical labeling of document images using layout graph matching with adaptive learning," in *Proceedings of Document Analysis Systems*, vol. 2423, 2002, pp. 224–235.

[6] ——, "Content features for logical document labeling," in *Proceedings of Document Recognition and Retrieval*, 2003, pp. 189–196.

[7] K. C. Santosh, "g-dice: graph mining-based document information content exploitation," *International Journal on Document Analysis and Recognition*, vol. 18, no. 4, pp. 337–355, 2015.

[8] N. Kooli and A. Belaïd, "Entity matching in ocred documents with redundant databases," in *Proceedings of the International Conference on Pattern Recognition Applications and Methods*, 2015, pp. 165–172.

[9] 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*, vol. 18, no. 3, pp. 265–298, 2004.

[10] E. R. Hancock and R. C. Wilson, "Pattern analysis with graphs: Parallel work at bern and york," *Pattern Recognition Letters*, vol. 33, no. 7, pp. 833–841, 2012.

[11] 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*, vol. 28, no. 1, 2014.

[12] A. Sanfeliu and K. Fu, "A distance measure between attributed relational graphs for pattern recognition," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 13, no. 3, pp. 353–362, 1983.

[13] P. L. Bodic, P. Héroux, 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*, vol. 45, no. 12, pp. 4214–4224, 2012.

[14] N. Kooli, A. Belaid, A. Joseph, and V. P. D'Andecy, "Entity local structure graph matching for mislabeling correction," in *Document Analysis Systems*, 2016, pp. 257–262.

[15] A. Wong, M. You, and S. Chan, "An algorithm for graph optimal monomorphism," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20, no. 3, pp. 628–638, 1990.

[16] P. Vijaya, M. N. Murty, and D. Subramanian, "Leaders-subleaders: An efficient hierarchical clustering algorithm for large data sets," *Pattern Recognition Letters*, vol. 25, no. 4, pp. 505–513, 2004.

[17] H. Bunke, P. Foggia, C. Guidobaldi, and M. Vento, "Graph clustering using the weighted minimum common supergraph," in *Proceedings of the IAPR International Conference on Graph Based Representations in Pattern Recognition*, 2003, pp. 235–246.