Semantic Label and Structure Model based Approach for Entity Recognition in Database Context

Nihel Kooli, Abdel Belaïd

Semantic Label and Structure Model based Approach for Entity Recognition in Database Context

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

***Abstract*—This paper proposes an entity recognition approach in scanned documents referring to their description in database records. First, using the database record values, the corresponding document fields are labeled. Second, entities are identified by their labels and ranked using a TF/IDF based score. For each entity, local labels are grouped into a graph. This graph is matched with a graph model (structure model) which represents geometric structures of local entity labels using a specific cost function. This model is trained on a set of well chosen entities semi-automatically annotated. At the end, a correction step allows us to complete the eventual entity mislabeling using geometrical relationships between labels. The evaluation on 200 business documents containing 500 entities reaches about 93% for recall and 97% for precision.**

I. Introduction

Entity Recognition (ER) is defined in [1] as the process of identifying and locating a term or a phrase in a textual document referring a particular entity such as a person, a place, an organization, etc. We are interested in ER in OCRed documents. The data contained in documents are often registered in databases constituted by experts. An entity contained in a database is described by a series of attributes or fields: text values whose semantics and types are informed by the columns of the database. We propose to detect entities in OCRed documents from their description in the database.
To achieve this goal, some difficulties should be overcome such as non-standardized representations of the entity in the document and in the database like abbreviations, incorrect or missing punctuation and permuted terms, in addition to the altered structure and possible OCR errors.

In the context of matching entity representation in documents with their description in a database, authors in [2] propose EROCS algorithm to identify entities embedded in document segments (few consecutive sentences). It uses a score, defined for an entity with respect to a segment, that considers the frequency of the common terms in the segment and their importance in the database. This work is related to textual documents. It uses strict comparison between terms and considers the text as a sequence of lines. We proposed, in an earlier work, a modified version, called M_EROCS [3], that treats scanned documents. M_EROCS identifies entity terms 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 all the terms in each block belong to only one entity. Also, it does not assemble non-contiguous parts of the entity. Finally, it does not take advantage of the logical and physical structure of the entity representation in the document which is the purpose of this paper. Some previous works, such as [4] and [5], propose to model the document layout for logical labeling and page classification. These models are proposed at document level and represent spatial relations between blocks segmented by the OCR. Other approaches, like [6], [7] and [8], learn a local layout structure from a training document and reuse it to extract fields in test documents. The weakness of such approaches is that they require user intervention for tagging semantic fields. This paper proposes a solution, based on automatic field labeling in the document, that matches labels with their corresponding entities in the database. This solution is reinforced by a structure model which represents geometrical relations between local entity labels in the document.

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

II. Proposed approach

Fig. 1 presents the global schema of the proposed approach. Firstly, fields are labeled in the document using an entity model generated from the database. Once labeled, the document is used to filter candidate entities contained in the entity model. A matching module is then applied to match between labels and candidate entities. At the end, a correction module based on entity structure modeling is integrated.

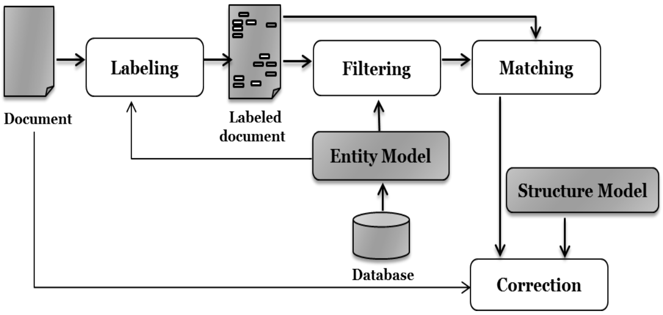


Fig. 1. Global schema of the proposed system.

A document image is physically defined as a hierarchy of zones containing lines and lines containing terms. Similarly, a database is composed of records containing fields and fields containing terms. To make possible the matching, we propose to define a structure model that represents the entities in the document and an entity model that describes the entities in the database. A structure model represents the physical and logical structures of entities in the document. An entity in the structure model can be represented by one or more structures where a structure is defined by lines arranging entity labels. An entity model describes logical definition of entities in the database. It represents field values of the entities. The entity model is the result of a pre-processing step of entity resolution which is performed in the database. The entity resolution step, described in our earlier work [3], aims to eliminate record redundancy in the database.

A. *Document labeling*

Fields contained in the entity model are labeled in the document using regular expressions built for columns of standard format or dictionaries created for noun columns. For OCR error tolerance, the n-gram similarity [9] is used to compare between fields in the dictionary and in the document. A label is defined as $l_i = (c_i, v_i)$ where v_i is the value of the label and c_i is the corresponding column in the database. v_i is represented by a bag of words $v_i = \{t_j\}$. A pre-step of label values standardization is performed. For example, punctuation are removed and phones are represented in a common form.

B. *Entity filtering*

Since the comparison of the document labels with all the entities in the entity model is complex, we propose to keep only the entities that may match the document. This consists of filtering entities using the labels in the document. Indeed, we keep candidate entities that have at least one field value that corresponds to one label value of the same column.

C. *Entity matching*

1) Matching score: For any document, we define a set of labels $d = \{l_i\}$. Let E be an entity model that contains n entities and m columns $\{c_k\}$. Each entity e in E has fields $\{e.c_k\}$ for the m columns. If e is an entity that matches the document, then each value $e.c_p$ in e corresponds to some value v_q of a label l_q contained in d.
Let $F(e,d)$ be the set of labels that belong to d and contained as well in the entity e, i.e.

$$l_i \in F(e,d) \equiv l_i \in d \text{ and } v_i \simeq e.c_i$$

where $v_i \simeq e.c_i$ means that v_i and $e.c_i$ are considered similar according to a similarity distance. The score of an entity e with respect to the set of labels d is defined as:

$$score(e,d) = \sum_{l_i \in F(e,d)} \sum_{t_j \in v_i} tf(l_i,d).idf(t_j,c_i).conf_i$$

where $tf(l_i,d)$ is the frequency of the label in the document, $idf(t_j,c_i)$ is the importance of the term t_j in the column c_i of the database and $conf_i$ is the confidence of labeling l_i given by the n-gram distance.
The set of labels d in the document is matched with an entity e_m when:

$$e_m = \arg\max_{e \in E} score(e,d)$$

We define a rejection threshold T for $score(e_m,s)$ to reduce the number of false positive entities. This threshold is empirically fixed.

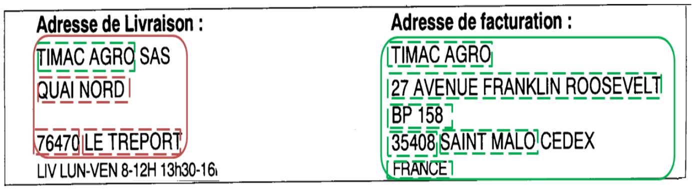
2) Matching algorithm: Algorithm 1 recursively matches a set of labels d with a set of candidate entities $setE$ based on the score maximization. d is initialized to the set of labels in the document. It is updated for each matched entity by removing the used labels in the matching. The recursive algorithm is stopped when d or $setE$ are empty.

```
input : $E$ // entities in the entity model
        $d$ // labels in the document
output: $matchE$ // matched entities
begin
    $matchE = \emptyset$;
    $setE = \texttt{filter}(E,d)$;
    while ($setE \neq \emptyset$ & $d \neq \emptyset$) do
        $setE = \{e \in setE \mid score(e,d) \geq T\}$;
        $e_{max} = \arg\max_{e \in setE} score(e,d)$;
        $d = \{l \in d \mid v \notin e_{max}.c\}$;
        $setE = setE \backslash \{e_{max}\}$;
        $matchE =$
        $\texttt{addEntity}(matchE, e_{max}, score(e_{max},d))$;
    end
    return ($matchE$);
end
```

Algorithm 1: Matching algorithm

3) Algorithm deficiencies: This algorithm considers only term values in the matching between document labels and entity fields. Hence, it may fail in the case of confusing labels that are shared between different entities in the document. Also, the algorithm is sensitive to mislabeling errors.

a) Confusing labels: The proposed algorithm does not overcome the confusion between labels that are repeated in the document and referring to different entities. Fig. 2 presents an example of a shared label (*TIMAC AGRO*) between two entities. This label is miss-associated to the first matched entity (framed in green color) which leads to a matching failure for the second entity (framed in red color). This failure is due to a low value of the matching score.



Adresse de Livraison :
TIMAC AGRO SAS
QUAI NORD
76470 LE TREPORT
LIV LUN-VEN 8-12H 13h30-16

Adresse de facturation :
TIMAC AGRO
27 AVENUE FRANKLIN ROOSEVELT
BP 158
35408 SAINT MALO CEDEX
FRANCE

Fig. 2. An example of matching failure due to a shared field between two entities in the same document.

b) Missing labels: The matching algorithm considers only the labels and so disregards the remaining text in the document. Hence, mislabeling problems, due to OCR error or non-standardized values between the document and the database, are not recoverable.

D. *Structure correction*

A correction module is proposed for mislabeling and label confusion problems. The structure model is used for the correction. It represents the arrangement of local entity labels in the document. The geometric configuration of a structure in the model is represented by a graph.

For an incomplete candidate entity due to mislabeling errors, an attributed sub-graph of the recognized labels is generated (see example in Fig. 3 (b)). The sub-graph is then matched to a graph structure in the model (see Fig. 3 (c)). The structure is then used to localize missing labels in the document (see the label framed in red color in Fig. 3 (a)). To complete the entity, nodes are spotted in the document using the spatial relations provided by the arcs $\{a_{ij}\}$ in the structural graph.

For a confusing label, the structure in the model is recalled to verify its attachment to the candidate entity. Similarly, a sub-graph is generated for reliable labels. The corresponding structure confirms the location of the confusing label.

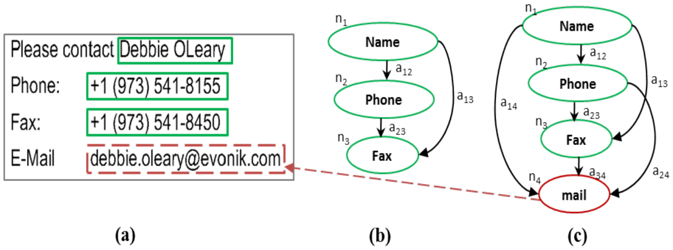


Fig. 3. (a) Entity representation in the document. (b) The sub-graph of recognized labels. (c) The structure of the entity.

1) Structure graph: An entity local structure in the document is modeled by a complete directed graph $G = (N, A)$ where N is a finite set of nodes that corresponds to semantic labels and $A \subseteq N \times N$ is a finite set of arcs that represent relations between the nodes. A node n_i that corresponds to a label l_i is defined by:

$$n_i = (c_i, conf_i)$$

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:

$$a_{ij} = (vs, hs, al)$$

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 and hs are signed to inform about the relative vertical position (above, below) or the relative direction (on the right, on the left). $al = (rJust, lJust, cent)$ is a vector of three binary values which inform about line alignment (right align, left align, centered text). Slight variation (lower than 20 pixels) between the line boundaries is tolerated for the alignment.

2) Graph matching: The idea is to retrieve the structure in the model that includes the sub-graph of a candidate entity.

For node mapping, we consider an exact match between label columns. The cost function for mapping a node n to a node n' with labels l and l' respectively is then defined as:

$$c_N(n, n') = \begin{cases} 1 - conf.conf' & \text{if } c = c'; \\ 1 & \text{else.} \end{cases}$$

For arc mapping, we define a cost function for two arcs $a = (vs, hs, al)$ and $a' = (vs', hs', al')$ as:

$$c_A(a, a') = \tfrac{1}{3} \textstyle\sum_f \lambda_f . d_{feature_f}(a, a') \quad \lambda_f \in [0, 1]$$

where weight λ_f is varied according to feature relevance. Feature dissimilarity measures are defined as:

$$d_{vs}(a, a') = |vs^N - {vs'}^N| \; ; \quad d_{hs}(a, a') = |hs^N - {hs'}^N| \; ;$$

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

$feature_f^N$ is the normalized value of $feature_f$ defined by:

$$feature_f^N = \frac{feature_f - \max feature_f}{\max feature_f - \min feature_f}$$

where $\max feature_f$ and $\min feature_f$ are dataset dependent.

The match cost for mapping a candidate graph $G = (N, A)$ to a structure graph $S = (N', A')$ in the model M is defined as:

$$C(G, S) = \frac{1}{2|N|} \sum_{n \in N} \lambda_{n'} c_N(n, n') + \frac{1}{2|A|} \sum_{a \in A} \lambda_{a'} c_A(a, a')$$

where $\lambda_{n'}, \lambda_{a'} \in [0, 1]$ are the weight factors for nodes and arcs in the structure graph. Given one structure and one candidate graph, we simply search for the best node mapping. Given several structures and one candidate graph, the matching is equivalent to the selection of the structure S_m, where:

$$S_m = \arg\min_{S_i \in M} C(G, S_i)$$

Graph matching is an interesting problem and is generally NP-hard. Branch and bound search using some heuristics or optimization techniques [10] constitutes a good alternative to solve this problem in practice. We apply a simplified version of the branch and bound algorithm to find the first one to one mapping between nodes in the candidate entity graph and the structure graph. It is simplified by heuristics which promote the exact matching between the nodes. Model graphs processing is simplified by a filtering on the set of nodes since we foster exact matching between the set of nodes.

3) Model learning: A structure graph in the model is learned from a set of training entity graphs from the studied corpus. Well chosen entities are semi-automatically annotated using the results of the entity matching step. Matched entities with success (True Positives) are automatically annotated and then revised manually to verify label attachment to each entity local structure. A graph is then created for each visually distinct local structure. The attributes of graph samples are fused to get the attributes of the structure graph. For distances, the sample average is computed. For alignment, the dominant value is used. 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.

4) Matching verification: Values of the identified labels, in the previous step, are matched with values in a dictionary built from the candidate entity fields. To overcome non-standardized values and OCR errors, we propose a modified Jaccard distance that combines token-based and edit-based distances as:

$$Jaccard_{edit}(U, V) = \frac{|U \cap V|}{|U \cup V|}$$

$U \cap V$ is the set of words $u \in U$ where there is some $v \in V$ such that $edit\_distance(U, V) < \theta$. $U \cup V$ is the set of words $u \in U$, $v \in V$ where for each u and v we have $edit\_distance(u, v) > \theta$. θ is a threshold for the edit distance.

III. Experiments

We work on a real-world industrial project in direct collaboration with the ITESOFT[1] company. For tests, we use a database that contains a table composed of 229345 records. It registers information about enterprises (industrial suppliers and clients) such as their names, addresses and contact numbers. We consider also a dataset of 200 printed documents. They contain 500 entities. These documents represent industrial invoices or purchase orders. For evaluation, we use a ground truth table that links each document with its contained entities identifiers in the database. This table was manually prepared by an industrial expert.

A. Matching method evaluation

Fields are labeled in the document using a company intern tool called FullText. We evaluate the labeling using an annotated corpus of about 100 documents. Mislabeling errors are about 11%. They are due to OCR errors (in 43.33% of cases), non-standardization of values (in 50% of cases) or fields spanned over several lines (in 6.66% of cases). TABLE I presents a sample of mislabeled fields caused by non-standardization of values, such as abbreviation and term permutation, or some altered characters by the OCR.

TABLE I. A sample of mislabeled fields in the document

Label column	**Value in database**	**Value in document**	**Causes**
Address	**AV** DE L'EUROPE	**AVENUE** DE L'EUROPE	abbreviation
Name	**TPS** DIGOINNAIS	**TRANSPORTS** DIGOINNAIS	abbreviation
Name	TRANSPORTS ALAIN CASSIER	CASSIER ALAIN TRANSPORTS	term permutation
Phone	0**1**45**6**23**0**78	0**i**45**b**23**O**78	OCR errors
Zip-code	3()6/	3**0**6**7**	OCR errors

The filtering step widely reduces the candidate entities used for the matching. Only $4,57\%$ of the total number of entities are considered as candidate ones on an average.

A relevant entity for a document is defined as an entity present in the document and that refers to a record in the database. Precision and Recall are defined as:

$$Recall = \frac{\#\ relevant\ matched\ entities}{\#\ relevant\ entities}$$

$$Precision = \frac{\#\ relevant\ matched\ entities}{\#\ matched\ entities}$$

Fig. 4 shows the evolution of Precision, Recall and F-measure of entity matching in documents with varying the threshold T defined in Section II-C1. It shows that setting the threshold value at 15 maximizes the value of F-measure $(91, 94\%)$. This value is retained to evaluate the matching results in the remaining experiments. The corresponding matching rates are 88.88% for Precision and 95.23% for Recall.

The sources of error are investigated. In about 7% of cases, failure is due to mislabeling errors. In about 3% of cases, it is due to the problem of confusing labels (explained in Section II-C3a). Finally, in 2% of cases, failure is due to the problem of incomplete entities (missing fields in the record). Errors due to mislabeling will be reduced by the structure correction.

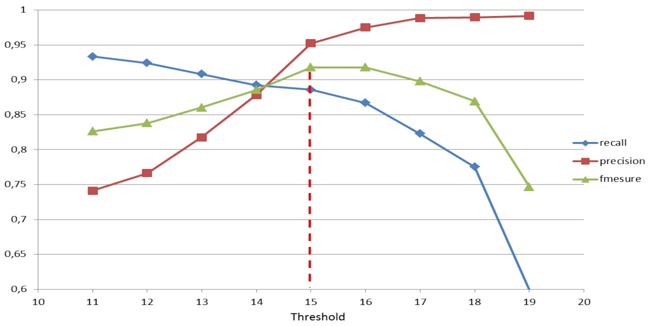


Fig. 4. Precision, Recall and F-measure of entity matching for varying the score threshold.

B. Structure correction evaluation

For structure model learning, we use 10% of entity dataset which corresponds to 50 entities contained in 32 documents.

Missing label detection in incomplete entities is evaluated using the successfully matched entities. Each time, we eliminate labels of a column type (Name, Address, Zip-code, ...) and try to detect them using the structure model. TABLE II presents the obtained results. Its shows that the identification of all column labels gets high Recall and Precision rates.

TABLE II. Label identification rates

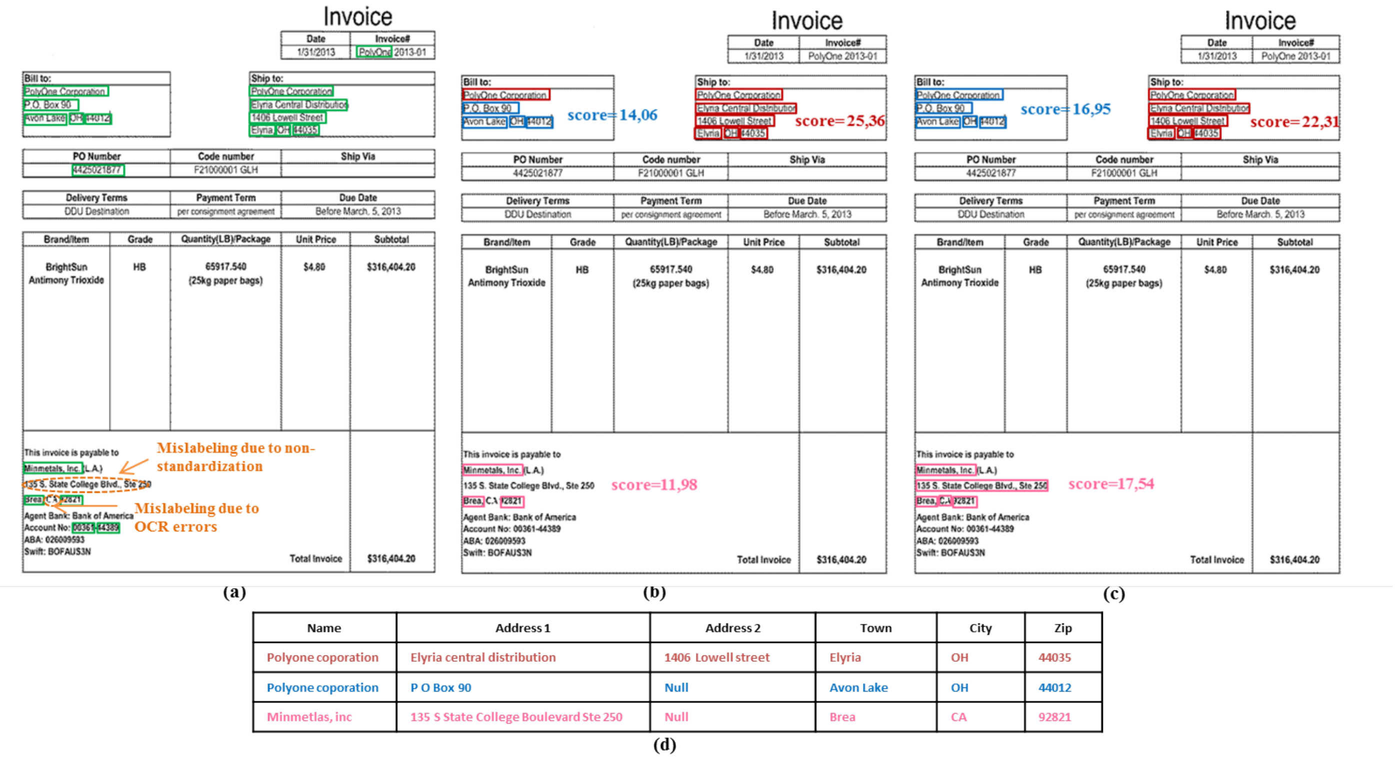
Label column	Number of labels			Recall (%)	Precision (%)
	Missing	Found	Correct		
Name	80	65	57	71.25	87.69
Address	80	71	64	80.00	90.14
Zip-code	100	89	89	89.00	100
City	100	86	84	84.00	97.67
Phone	50	44	35	70.00	79.55
Fax	50	40	32	64.00	80.00
Vat number	50	39	31	62.00	79.49

Fig. 5 shows results of ER in an invoice. Entities of interest are described in a sample of the database in Fig. 5 (d). In Fig. 5 (a), we see examples of mislabeling problems due to OCR errors or non-standardization in value representation. Fig. 5 (b) shows two examples of false negative entities which have scores below the threshold 15. The one in blue is due to the confusing label (*Polyone Corporation*) associated by mistake to the red entity. The one in pink is due to mislabeling errors. In Fig. 5 (c) the matching is corrected by the structure model. Missing labels are identified. Label confusion is solved and the matching score is increased.

Matching results are evaluated after the correction using the structure model. Recall and Precision reaches 93.37% and 97.50% respectively.

The proposed approach is compared with two works in the state of the art. TABLE III presents the obtained results. It shows a significant increase in Precision and Recall compared to the evaluation of EROCS and M_EROCS methods on our corpus. Furthermore, we see an important decrease in the run

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


Mislabeling due to non-standardization
Mislabeling due to OCR errors
score=14,06
score=25,36
score=11,98
score=16,95
score=22,31
score=17,54
(a)
(b)
(c)
(d)

Name	Address 1	Address 2	Town	City	Zip
Polyone coporation	Elyria central distribution	1406 Lowell street	Elyria	OH	44035
Polyone coporation	P O Box 90	Null	Avon Lake	OH	44012
Minmetlas, inc	135 S State College Boulevard Ste 250	Null	Brea	CA	92821

Fig. 5. Example showing entity recognition results. (a) Field labeling results. (b) Matching results. (c) Matching correction results. (d) Searched entities as described in the database.

time due to the labeling and the filtering steps. However, it increases slightly with the integration of the correction module.

TABLE III. ENTITY MATCHING RATES COMPARISON

	Recall (%)	Precision (%)	Fmeasure (%)	Runtime (sec/doc)
EROCS [2]	67.58	54.09	60.09	69.5
M_EROCS [3]	73.36	69.58	71.43	4.4
Matching method	88.88	95.23	91, 94	0.7
Matching method + Structure correction	93.37	97.50	95.39	1

IV. CONCLUSION AND FUTURE WORK

This paper proposes an approach of entity matching in documents with their description in a database. The structural modeling of semantic labels has proved to be effective in reducing false negative entities. The results on a dataset of 200 documents are promising and achieve about 93% for recall and 97% for precision.

Our future work is to introduce adaptive learning of features and weights in the structure model based on the matching feedback. Furthermore, we plan to enhance the verification step by combining different similarity measures and using an OCR correction model based on character shape classification. Another perspective is the use of other corpus, limited in this study to enterprise entities, in order to integrate more physical and logical structures of the document and to exploit them in the entity search.

V. ACKNOWLEDGMENT

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

REFERENCES

[1] Osesina, I. and Talburt, J.: A Data-Intensive Approach to Named Entity Recognition Combining Contextual and Intrinsic Indicators, International Journal of Business Intelligence Research, 2012, 3(1): pp. 55–71.

[2] Chakaravarthy, V.T., Gupta H., Roy P. and Mohania, M.: Efficiently Linking Text Documents with Relevant Structured Information, in International Conference on Very Large Data Bases, 2006, pp. 667–678.

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

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

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

[6] Rusinol, M., Benkhelfallah, T. and D'Andecy, V.P.: Field Extraction from Administrative Documents by Incremental Structural Templates, in International Conference on Document Analysis and Recognition, 2013, pp. 1100–1104.

[7] Peanho, C., Stagni, H. and Correa da Silva, F.: Semantic information extraction from images of complex documents, in: Applied Intelligence, 37(5): pp. 543-557, 2012.

[8] Ishitani, Y.: Model-based information extraction method tolerant of OCR errors for document images, in International Conference on Document Analysis and Recognition, 2001, pp. 908-915.

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

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