# Entity Matching in OCRed Documents with Redundant Databases

Nihel Kooli, Abdel Belaïd

# Entity Matching in OCRed Documents with Redundant Databases

Nihel Kooli and Abdel Belaïd
*LORIA Campus, scientifique BP 239 54506, Vandoeuvre-lès-Nancy Cedex, France*

Keywords: Entity Recognition, Entity Resolution, Entity Matching, OCRed Documents, Redundant Databases.

Abstract: This paper presents an entity recognition approach on documents recognized by OCR (Optical Character Recognition). The recognition is formulated as a task of matching entities in a database with their representations in a document. A pre-processing step of entity resolution is performed on the database to provide a better representation of the entities. For this, a statistical model based on record linkage and record merge phases is used. Furthermore, documents recognized by OCR can contain noisy data and altered structure. An adapted method is proposed to retrieve the entities from their structures by tolerating possible OCR errors. A modified version of EROCS is applied to this problem by adapting the notion of segments to blocks provided by the OCR. It handles document segments to match the document to its corresponding entities. For efficiency, a process of data labeling in the document is applied in order to filter the compared entities and segments. The evaluation on business documents shows a significant improvement of matching rates compared to those of EROCS.

## 1 INTRODUCTION

With the growth of industrial data and the multitude of inflow sources in the companies, the processed documents can be of different types (electronic, scanned, etc.) and different classes (purchase orders, invoices, etc.). These documents often contain unstructured textual data. For a better management, it is necessary to automate the task of data processing in documents by identifying and extracting contained information and then structuring it.

The data contained in administrative documents are often predefined in structured catalogs: databases prepared by experts. An entity contained in a catalog is represented by a series of fields: text values whose semantics and types are informed by the headers of the database.

Entity Recognition is the process of identifying and locating a term or phrase in an unstructured textual document referring a particular entity such as a person, a place, an organization, etc. In this context, the problem can be reformulated as a task of matching a mentioned entity in the document with its representation in the database. This task is far from being solved by a simple intersection between words in document and those in database records and it is a more complex one since:

- Documents do not explicitly mention unique identity of entities as defined in the database.
- Terms defining the entity in the document differ from terms defining the entity in the database. This is caused by non-standardized representations like abbreviations, incorrect or missing punctuation and fused or split words.
- The extraction of content from documents recognized by Optical Character Recognition (called OCRed documents in this paper) is a challenging task since we have to deal with OCR errors.
- OCR poorly reproduces the physical and the logical structures of the document.
- Industrial databases are often voluminous, contain poor quality of data such as missing fields, typing errors and non-standardized representation. Also, they contain redundant records which could be represented differently but refer to the same real world entity. Such database is called redundant database in this paper.

  Entities defined in databases can be duplicated since database can be either dynamic, managed by different users, or a result of merging multiple databases. These problems related to the database make the comparison task with document complex. The task of resolving such problems is called entity resolution. An approach that benefits from entity resolution to enhance entity recognition in documents is proposed.

Kooli N. and Belaïd A..
Entity Matching in OCRed Documents with Redundant Databases.
DOI: 10.5220/0005177301650172
In *Proceedings of the International Conference on Pattern Recognition Applications and Methods* (ICPRAM-2015), pages 165-172
ISBN: 978-989-758-076-5

In this paper, we address the idiosyncrasies of entity recognition in an OCRed document that contains noisy data and has a specific structure. We propose a solution that retrieves entities from OCR blocks and identifies their structured representation in the database in spite of the altered content of OCR. Furthermore, we combine the two problems of entity resolution in industrial databases and entity recognition in documents and we show how the entity resolution task can enhance the entity matching task by improving the quality of the database.

The remainder of this paper is organized as follows: Section 2 presents some related research about entity resolution and entity recognition, Section 3 details the proposed approach and Section 4 presents the experiments on real world data.

## 2 RELATED WORK

**Entity Resolution.** To make matching decision between records, some researchers proposed deterministic approaches which are based on a preliminary rule definition (Lee et al., 2000) and probabilistic approaches that use statistics to assign a probabilistic weighting to record pairs. Fellegi and Senter proposed in (Fellegi and Sunter, 1969) a statistical model of linkage between records based on probabilistic definitions. This unsupervised machine learning method views the problem as a task of defining a feature vector for record pairs into three classes: links, non-links and possible links (undecided case requiring, for example, human review). This suggested model has been, by and large, adopted by subsequent researchers.

Some works, such as (Bilenko et al., 2003), are based on attribute comparisons in records using similarity distances such as edit-based distances (example: the edit-distance, Jaro-Winkler), token-based distances and hybrid ones (example: Soft-tf-idf). The authors in (Cohen et al., 2003) studied the comparison between these different measures on record linkage results.

In large databases, record pairs comparison is expensive since it consists of a cartesian product of records. (Bilenko, 2006) proposed a step of blocking which consists of separating the database into blocks that contain approximately similar records. This separation could be based on a selection of keys such as regrouping records that share the same zip-code or the same first three characters of the name.

**Entity Recognition.** In the literature, several studies have addressed the problem of entity recognition in unstructured documents. These studies could be classified into three categories: context-oriented approaches, data-intensive approaches and mixed approaches.

- Context-oriented approaches are based on contextual rules and require an explicit linguistic and grammatical recognition of text using syntactic and eventual semantic labeling. We mention, for example, (Hashemi et al., 2003) that introduced an entity recognition technique for extracting names, titles and their associations. The problem with these approaches is their non-generality since they are dependent on the language and the domain of text. In addition, the task of defining contextual rules is complex in time and resources.
- For data-intensive approaches, entity recognition techniques do not require any explicit grammatical knowledge of the document but obtain their information from an annotated corpus (for example a knowledge database). These approaches treat the problem as a classification task and apply machine learning models to solve it. Different supervised learning methods were used for entity recognition, such as (Zhang et al., 2010) that uses Support Vector Machine (SVMs) for recognizing investigator names in articles.
- Mixed entity recognition methods try to combine the advantages of the machine learning and the rule based approaches. For example, (Laishram and Kaur, 2013) combines Conditional Random Field (CRF) with rule definition that helps to identify features for some particular entities to improve the classification by the CRF.

In the context of data-intensive approach, the corpus could be a structured database that represents the entities. For matching entities in documents with their representations in a database, (Chakaravarthy et al., 2006) proposed an algorithm, called EROCS, that identifies entities embedded in document segments. 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, concerns electronic textual documents. It has deficiencies in the case of OCRed documents caused by their altered structure and content. Indeed, it treats strict comparison between terms and considers the text as a sequence of lines.

The task of disambiguation in the case of more than one record returned for an entity referenced in the document has been treated by (Wu et al., 2007) that exploited relationships between candidate entities for different fragments (a fragment is defined as entity features extracted from the document using seman-

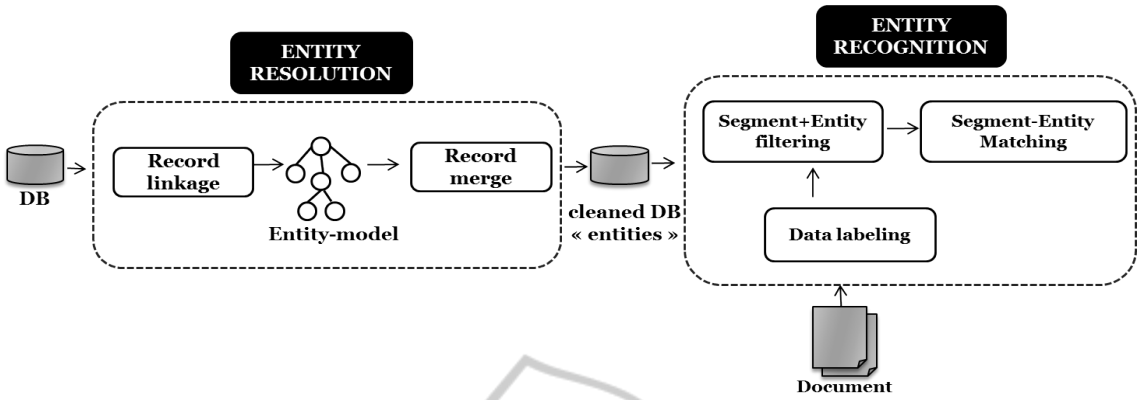


Figure 1: Global schema of the proposed system.

tic patterns) of the same document for identifying the identity of fragments. In contrast, our approach does not remedy ambiguity cases but it avoids them from the beginning.

Some works treat the problem of Information Retrieval (IR) in noisy text in the context of OCR. (Taghva et al., 2006) proposes studies that prove the degradation of the information extraction process caused by altered characters in OCR text. For OCR error correction, (Pereda and Taghva, 2011) uses approximate string matching to remedy some character misrecognition.

# 3 PROPOSED APPROACH

Figure 1 presents the global schema of the proposed approach. It is a succession of two main modules. The entity resolution module takes as input a database, proposes to link its contained records and constructs an entity model for records referring the same real word entity. It then uses this model to produce as output a cleaned database composed of entities by merging linked records. The entity recognition module proposes to label data in the document. These labels are used to filter segments in the document and entities in the database. Then, this module tries to find the entities in the document segments.

## 3.1 Entity Resolution

The module of entity resolution is composed of two main sub-tasks: record linkage and record merge described in the following.

### 3.1.1 Record Linkage

Let $E$ be a structured database that contains $n$ records $\{r_i\}$ and $m$ columns $\{c_j\}$. Each record $r_i$ is composed of $m$ attributes corresponding to each column. The attribute of $r_i$ that corresponds to the column $c_j$ is noted $r_i.c_j$. We propose to group the records in $E$ into $q$ blocks $\{B_k\}$ using regrouping keys. Each block contains records that may refer the same entity. The number of records in $B_k$ is defined as $|B_k|$.

Let $dist(a,b)$ be the similarity distance between two field values $a$ and $b$. These latter are decided to be similar if their distance exceeds a threshold $T1$. For the linkage decision, the statistical model given by (Fellegi and Sunter, 1969) is used. It estimates matching $M(p)$ and unmatching probabilities $U(p)$ for each field of index $p$. We propose to compute a ratio ($ratio$) between each pair of records and decide to link them if the value of $ratio$ exceeds the threshold $T2$ (see Algorithm 1). $T1$ and $T2$ are empirically defined.

The output of the record linkage process is represented as an entity model where each entity is represented by a tree of three levels: the root node represents the entity reference, the nodes of second level represent the entity fields where each field node is related to nodes of the third level that correspond to different field values of the linked records.

### 3.1.2 Record Merge

Record merge considers linked records in the entity model. It returns a new record that provides better information.

For identical records, this step of fusion will remove duplicates. For complementary records where each record provides additional attributes, we propose to gather all values in a merge resulting record. For example, the following linked records *r1* and *r2* are merged into *r3*.
*r1 = [name: Xerox, zip-code: 92202]*
*r2 = [name: Xerox, phone: 0825082081]*
*r3 = [name: Xerox, zip-code: 92202, phone: 0825082081]*

```
input  : E // the database
         T1, T2: int // the thresholds
output: E' // linked database
for k ← 1 to q do
  for i ← 1 to |B_k|- 1 do
    for j ← i+1 to |B_k| do
      Ratio = 0;
      for p ← 1 to m do
        d = dist(r_i.c_p, r_j.c_p);
        if d ≥ T1 then
          Ratio+ = log(M(p)/U(p)) * d;
        else
          Ratio+ = log((1−M(p))/(1−U(p)));
        end
      end
      if Ratio ≥ T2 then
        E' = link(E, r_i, r_j)
      end
    end
  end
end
```

**Algorithm 1:** The record linkage algorithm.

A record is said dominated by another when the second record contains more information that enables it to give a higher quality description of the underlying entity. In the case of domination between records, the merge step will keep the record that better describes the entity. In the previous example, we keep *r1* which is dominated by *r3* since it describes the entity with more attributes. For records with differences in some attributes caused by various descriptions of the entity or changes of its characteristics, different representations are retained in the resulting record. For example, for the linked records *r3* and *r4*, different phone numbers of the entity are retained in the record *r5* for future use.

*r4 = [name: Xerox, zip-code: 92202, phone: 0825082082]*

*r5 = [name: Xerox, zip-code: 92202, phone: <0825082081; 0825082082>]*

## 3.2 Entity Recognition

Entity recognition consists of matching entities in the document with referring to the database. It includes three main steps : data labeling, segment and entity filtering and the matching described in the following.

### 3.2.1 Data Labeling

Data labeling consists of identifying some entity components in the document in order to localize their containing segments.

Dictionaries and regular expressions, defined from instances in the database, are used for this labeling.

### 3.2.2 Segment and Entity Filtering

Comparing all terms in each segment with all terms of each entity is a costly task. We propose, then, to localize segments that may refer an entity in the database and limit the search to only these segments. This consists of filtering segments in the document focusing only on segments that contain some labeled data. Furthermore, the labeled data is used to filter entities in the database and keep only the records that have at least one field value that corresponds to one label value of the same type. This step of filtering will improve the efficiency of the matching.

### 3.2.3 Entity-document Matching Model

For entity matching in the document with the database, we are inspired by EROCS algorithm (Chakaravarthy et al., 2006) which is described in the following. It perceives the document $d$ as a set of segments $d = \{s_i\}$ where each segment $s_i$ is a collection of terms $s_i = \{t_j\}$. Each record of the structured database $E$ is considered as an entity $e$, having its own terms $e = \{t_l\}$ contained in its attributes. If $e$ is an entity that matches the segment $s_l$, then each term $t_p$ in $s_l$ corresponds to some term $t_q$ of the entity $e$.

The weight of a term $t$ is defined as:

$$w(t) = \begin{cases} \log((N/n(t)) & \text{if } n(t) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $N$ is the total number of distinct entities in the database, and $n(t)$ is the number of distinct entities that contain $t$.

Let $T(e,s)$ be the set of terms that appear in the segment $s$ and contained as well in the entity $e$. The score of an entity $e$ with respect to a segment $s$ is defined by (Chakaravarthy et al., 2006) as:

$$score(e,s) = \sum_{t \in T(e,s)} TF(t,s).w(t) \quad (2)$$

For OCR error tolerance, this approach proposes an adapted score that employs the edit distance as:

$$score(e,s) = \sum_{t_1 \in close(\theta, T(s), T(e))} TF(t_1,s).w(t_2) \quad (3)$$

where $close(\theta, T(s), T(e))$ is the set of terms $t_1$ in s such that there is some $t_2$ in e with $dist(t1,t2) \leq \theta$. $dist(t1,t2)$ is defined as the edit distance between $t_1$ and $t_2$. It computes the minimum number of edit characters (insert, delete, substitute) required to convert the string $t_1$ to the string $t_2$ with $|t_1| \leq |t_2|$.

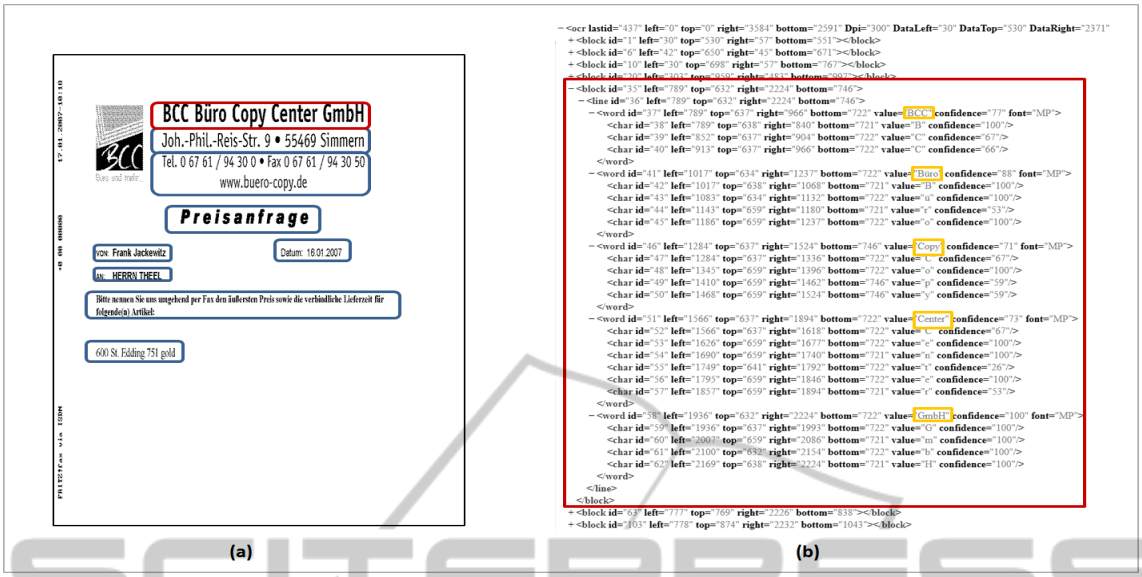


Figure 2: (a) An example of image document. (b) The OCR result of the block surrounded in red color.

EROCS proposes to match each segment $s$ in the document with an entity $e$ in the database where:

$$e_{max} = \arg\max_{e \in E} score(e, s)$$

To limit false positive (FP) matching cases, this approach defines a threshold of rejection $T$ as: if $score(e_{max}, s) > T$ then we decide to match the segment $s$ with the entity $e_{max}$.

### 3.2.4 OCRed Documents Model for Matching

An OCRed document is represented by a hierarchy of blocks containing lines and lines containing words. Figure 2 (a) presents an example of a document where blocks obtained from OCR are surrounded. Figure 2 (b) presents the OCR result of its block surrounded in red color in the document.

A segment in the matching model detailed in Section 3.2.3 corresponds, firstly, to a block in the OCR. However, an entity can be split into more than one segment of the OCR as shown in Figure 3. Indeed, the entity surrounded of green color is split into four segments. We propose then to increasingly construct a segment by merging consecutive blocks.

Furthermore, OCR does not restitute the physical and the logical structure of the original image document. An entity can therefore be split into non-consecutive blocks. For example, in Figure 3, the entity is split into blocks 4,6,8 and 10. We use the block coordinates in the OCR to merge contiguous blocks into one segment.

For matching, Algorithm 2 details the proposed algorithm. This algorithm merges increasingly contiguous segments based on an iterative score computation.

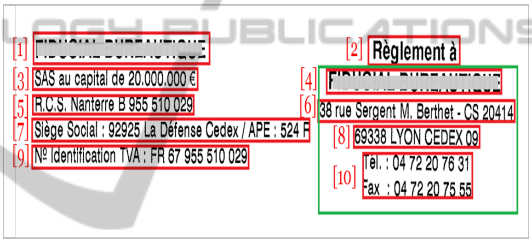


Figure 3: An example of entity structure altered by OCR.

It identifies the set of segments that contain some labeled data (*segSet*). For each identified segment ($s$), it enlarges the search area by adding each time the closest segment while the matching score is increased during $\lambda$ consecutive merges. Once the merge is stopped, the last $\lambda$ added segments are removed from the resulting segment. $\lambda$ is experimentally set to 2. The entity that corresponds to the resulting segment is retained if its score exceeds a threshold $T$.

## 4 EXPERIMENTS

For the experiments, we use a database containing a table of suppliers composed of 229345 records. It carries information about suppliers such as their names, addresses and phone numbers. The supplier is the entity of interest. We consider a dataset of 500 image documents and their OCR results. These documents represent industrial invoices or purchase orders. For evaluation, we use a ground truth data containing a table that relates each document with its contained entity identifiers in the database. A document can be

```
input  : E // the database
         D // the document
output: matchE = ∅ // matched entities
D′= dataLabeling (E,D);
segSet = segFilter (D′);
foreach s in segSet do
    E′ = entityFilter (E,D′,s);
    score = max_{e∈E′} score(e,s);
    emax = arg max_{e∈E′} score(e,s);
    i=0;
    while i < λ do
        s= addClosestSeg (s);
        scoreNew = max_{e∈E′} score(e,s);
        emax = arg max_{e∈E′} score(e,s);
        i++;
        if scoreNew ≥ score then
            i = 0;
            score = scoreNew;
        end
    end
    s= removeClosestSeg (s,λ);
    if score ≥ T then
        matchE =
        addEntity(matchE, emax, score);
    end
end
```

**Algorithm 2:** The matching algorithm.

related to one supplier's identifier or more (in the case of an entity that can reference a client in the document and is present in the DB as a supplier of an other document). This table was manually prepared by an industrial expert.

## 4.1 Entity Resolution

For computing similarity between field attribute pairs, we propose to use *Soft-tf-idf* measure combined with *Jaro-Winkler* measure since we have multiterm attributes with possible typing errors. The choice is based on comparative study of similarity measures proposed in (Cohen et al., 2003) with a fixed $threshold = 0.8$ for *Jaro-Winkler* distance.

The matching probability is defined as: $Mprobability = 1 - error\_rate$ where $error\_rate$ is defined as the percentage that a pair of matched records do not agree on the field. It is estimated by a preliminary review of labeled data. The unmatching probability for each field is approximated with the frequency of its distinct values. It is defined as: $Uprobability = 1/\#distinct\_values$. Figure 4 presents the evolution of unmatched record pairs frequency varying the value of coupling ratio. This Figure shows the three zones of the probabilistic model: the non-matching zone, the matching zone and the undecided zone. One may note the presence of FP (False Positives) pairs in the zone of matching and FN (False Negatives) in the zone of non-matching.

Table 1: Record linkage for varying key values in blocking.

| | Blocks number | Compared pairs number | Linked pairs number |
|---|---|---|---|
| Without block | — | 2643850 | 11750 |
| Key1 | 953 | 48772 | 11599 |
| Key2 | 498 | 104182 | 11746 |
| Key3 | 1114 | 92411 | 11745 |

A matching pair is defined as a pair that refers the same entity in reality. Precision and Recall are then defined as:

$$Recall = \frac{\#correctly\ linked\ pairs}{\#matching\ pairs}$$

$$Precision = \frac{\#correctly\ linked\ pairs}{\#linked\ pairs}$$

Figure 5 presents the evolution of Precision, Recall and F-measure for varying the coupling ratio threshold. The curves show that setting the threshold value at 13 maximizes the value of F-measure. This value is retained in the rest of the study.

For blocking evaluations, we define three different keys (*key1*: the supplier's name, *key2*: the first term of the name and *key3*: concatenation of the first term of the name with the zip-code ). Table 1 presents the results for different key values obtained by entity resolution in a snippet of 2300 records of the *Suppliers* database.

This Table shows that the blocking produces a significant decrease in the number of compared pairs of records compared to that of record linkage without blocking. However, restricting the choice could influence the quality of results as for *key1* where one may notice that the low number of comparisons relatively to other keys causes a reduction in the number of linked pairs. One may notice that *key3* decreases the number of compared records compared to that of *key2* while retaining nearly the same number of linked records. *key3* is considered as the key value in the rest of the study.

After a merging step, results show a decrease of about 30% in the number of records in *Suppliers* table.

## 4.2 Entity Recognition

OCRed documents are parsed to extract segments with their contained words. Also, a vector of terms

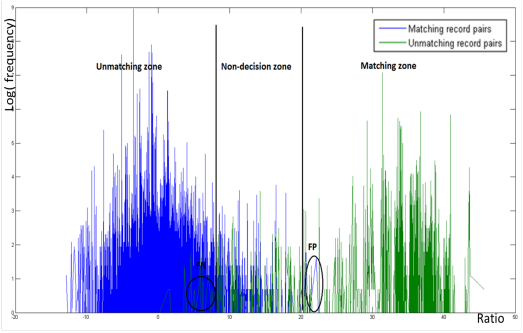


Figure 4: Ratio coupling.

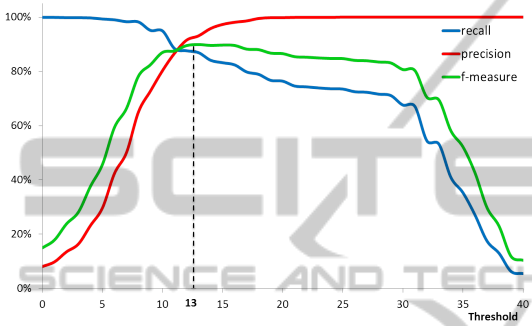


Figure 5: Precision, Recall and F-measure of record linkage for varying threshold of ratio.

is constructed for each entity in the database and its corresponding weight as defined in eq (1).

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

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

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

Figure 6 shows the evolution of Precision, Recall and F-measure of entity matching in documents with varying the threshold $T$ defined in Section 3.2.3. It shows that setting the threshold value at 17 maximize the value of F-measure. This value is retained to evaluate the matching results in the rest of the study.

Entity recognition process is evaluated for the main proposed improvement on the original version of EROCS. M_EROCS$_1$ is defined as the first modified version with OCR error tolerance that consists of integrating edit distance in the matching score as shown in eq (3). M_EROCS$_2$ concerns the integration of the filtering step detailed in Section 3.2.2. M_EROCS$_3$ concerns the application of the Entity Resolution process.

Table 2 presents the obtained results. It shows that the Recall of matching increased from 67.58% for EROCS to 71.05% for M_EROCS$_1$ but the Run-

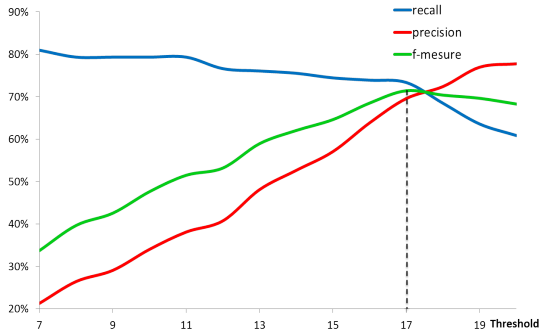


Figure 6: Precision, Recall and F-measure of entity matching for varying threshold of score.

time increased slightly due to the edit distance comparisons. Also, it shows a significant decrease in the run time for M_EROCS$_2$ (from 70.8 to 6.2 sec per document). Furthermore, the Recall and the Precision were improved respectively with 2.31 and 14.81 points for M_EROCS$_3$. The Runtime was also reduced by about 32% which is in relation to the reduction of about 30% of record number.

The increase of matching rates for M_EROCS$_3$ is due to the step of merging records that completes missing information in the matched entity and the increase of some term weights in this entity thanks to redundancy reduction. In addition, entity resolution solves some ambiguity cases due to the reduction of several records, referring the same entity and maximizing the score, into only one record. It also reduces the algorithm complexity and the execution time thanks to the reduction of the number of compared entities in database for each segment in the document.

The error cases of matching are explained. In about 5% of cases, failure is caused by the fact that entity is not well represented in the document. For example, the supplier of document could be represented only by a logo. In about 5% of cases, failure is caused by the bad quality of scanned documents that produces grave OCR errors. In about 4% of cases, it is caused by non-standardized representations of entity terms in the document and the database. In about 7% of cases, it is caused by incomplete entities in database where some fields are missing even after entity resolution step. Finally, in about 6%, failure is caused by non contiguity of entity components.

## 5 CONCLUSION AND FUTURE WORK

We present a method, called M_EROCS, for entity matching in the database with their representations

Table 2: Entity matching rates.

| | Recall (%) | Precision (%) | Fmeasure (%) | Runtime (sec/doc) |
|---|---|---|---|---|
| **EROCS** (Chakaravarthy et al., 2006) | 67.58 | 54,09 | 60,09 | 69.5 |
| **M_EROCS**$_1$(+OCR error tolerance) | 71,05 | 53,89 | 61,29 | 70.8 |
| **M_EROCS**$_2$(+filtering) | 71,05 | 54,77 | 61,86 | 6.2 |
| **M_EROCS**$_3$ (+entity resolution) | **73,36** | **69,58** | **71,43** | **4,4** |

by segments in OCRed document. The extensions on term matching and segment restructure of EROCS are proven effective for OCRed documents which have altered content and structure. A filtering step based on data labeling reduces the runtime from 70.8 sec to 6.2 sec per document. The pre-processing step of entity resolution on the database improves the matching rates with 2.31 points for the recall and 14.81 points for the precision and it decreases the runtime with about 1.8 seconds by document. The results on a dataset of 500 documents are promising and achieve about 73% for recall and about 70% for precision.

The future work is to solve the problem of non-contiguity of elements composing an entity. In case of incomplete entity, we will choose from distant labeled elements those they complete correctly the entity. The choice will be focused on the elements increasing the matching score. Furthermore, we will plan the use of other datasets, limited in this study to supplier entities, in order to enlarge the field search to all elements (close and distant) with more complex spatial relations. The idea is to integrate the physical and logical structures of the document and to exploit them in the element searching. Another prospect is to apply other methods for OCR matching and correction. A dictionary that maintains spell variations of fields, such as abbreviations and character variations, will be used.

## ACKNOWLEDGEMENT

We would like to thank our collaborator ITESOFT for providing real word data (images, OCRed documents and database) for test.

## REFERENCES

Bilenko, M. (2006). Adaptive blocking: Learning to scale up record linkage. In *Proceedings of the 6th IEEE International Conference on Data Mining*, pages 87–96.

Bilenko, M., Mooney, R., Cohen, W., Ravikumar, P., and Fienberg, S. (2003). Adaptive name matching in information integration. *IEEE Intelligent Systems*, 18(5):16–23.

Chakaravarthy, V. T., Gupta, H., Roy, P., and Mohania, M. (2006). Efficiently linking text documents with relevant structured information. In *International Conference on Very Large Data Bases*, pages 667–678.

Cohen, W. W., Ravikumar, P., and Fienberg, S. E. (2003). A comparison of string distance metrics for name-matching tasks. In *Proceedings of IJCAI-03 Workshop on Information Integration*, pages 73–78.

Fellegi, I. P. and Sunter, A. B. (1969). A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210.

Hashemi, R. R., Ford, C., Bansal, A., Sieloff, S. D., and Talburt, J. R. (2003). Building semantic-rich patterns for extracting features from events of an on-line newspaper. In *Proceedings of the IADIS International Conference WWW/Internet*, pages 627–634.

Laishram, J. and Kaur, D. (2013). Named entity recognition in Manipuri: a hybrid approach. In *The International Conference of the German Society for Computational Linguistics and Language Technology*, volume 8105, pages 104–110.

Lee, M.-L., Ling, T. W., and Low, W. L. (2000). Intelliclean: a knowledge-based intelligent data cleaner. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 290–294.

Pereda, R. and Taghva, K. (2011). Fuzzy information extraction on OCR text. In *ITNG*, pages 543–546.

Taghva, K., Beckley, R., and Coombs, J. S. (2006). The effects of OCR error on the extraction of private information. In *Document Analysis Systems*, pages 348–357.

Wu, N., Talburt, J., Heien, C., Pippenger, N., Chiang, C.-C., Pierce, E., Gulley, E., and Moore, J. (2007). A method for entity identification in open source documents with partially redacted attributes. *J. Comput. Small Coll.*, 22(5):138–144.

Zhang, X., Zou, J., Le, D. X., and Thoma, G. R. (2010). Investigator name recognition from medical journal articles: a comparative study of svm and structural svm. In *Document Analysis Systems*, ACM International Conference Proceeding Series, pages 121–128.