Unsupervised Information Extraction with the ONDUX Tool

André Porto Eli Cortez Altigran S. da Silva Edleno S. de Moura

[1]Departamento de Ciência da Computação
Universidade Federal do Amazonas (UFAM)
Manaus – AM– Brazil

{andreporto,eccv,alti,edleno}@dcc.ufam.edu.br

Abstract. *We present a tool that implements ONDUX, a probabilistic method for extracting information from unstructured data records using text segmentation. ONDUX relies on data available in a pre-existing data source on a given domain to associate segments from an input text with attributes of this domain. Unlike other approaches, ONDUX adopts effective approximate matching strategies instead of explicit learning strategies. This assigns to ONDUX a high degree of flexibility with less user interventions. The tool implements the functionalities of the method, allowing users to perform the extraction process and follow its steps by means of a graphical user interface. We describe our tool, discuss its technical details, and illustrate its main features by means of a case study.*

1. Introduction

The abundance of on-line sources of textual documents containing implicit semi-structured data records in the form of continuous text, such as product descriptions, bibliographic citations, postal addresses, classified ads, etc., has attracted a number of research efforts towards automatically extracting data of interest by segmenting the text containing them [Agichtein and Ganti 2004, C. Zhao et. al. 2008, E. Cortez et. al. 2010, Mansuri and Sarawagi 2006]. This interest is motivated by the necessity of having these data stored in some structured format as relational databases or XML, so that it can be further queried, processed and analyzed.

We present in this demo a tool that implements an unsupervised information extraction method called ONDUX (On-Demand Unsupervised Information Extraction) [E. Cortez et. al. 2010] for extracting information from unstructured data records. ONDUX relies on a Knowledge Base composed by sets of attribute values, taken from pre-existing data sources, to associate segments in the input text with attributes of a given domain. Different from previous work [Agichtein and Ganti 2004, V. Borkar et. al. 2001, Mansuri and Sarawagi 2006], there is not an explicit learning process in this step. Instead, the method uses simple generic approximate matching functions to compute a score measuring the likelihood of text segments to occur as a typical value of an attribute.

In sum, the main features of the ONDUX Tool are: (1) an effective implementation of an unsupervised probabilistic approach for information extraction by text segmentation; (2) a friendly graphical user interface that allows non expert users to easily carry out information extraction tasks; and (3) visualization facilities that allow users to follow and understand all the steps involved in the extraction process.

In the following, we first overview the ONDUX method in Section 2 and then describe the ONDUX Tool in Section 3. Next, in Section 4, we present a case study of

a typical information extraction task carried out using the ONDUX Tool, and outline the demonstration we prepared in Section 5. Finally, in Section 6 we present our conclusions.

$$K = \{\langle Neighborhood, O_{Neighborhood}\rangle, \langle Street, O_{Street}\rangle, \langle Bathrooms, O_{Bathrooms}, Phone, O_{Phone}\rangle\}$$
$$O_{Neighborhood} = \{\text{“Regent Square”,“Milenight Park”}\}$$
$$O_{Street} = \{\text{“Regent St.”,“Morewood Ave.”,“Square Ave. Park”}\}$$
$$O_{Bathrooms} = \{\text{“Two Bathrooms”,“5 Bathrooms”}\}$$
$$O_{Phone} = \{\text{“(323) 462-6252”,“171 289-7527”}\}$$

Figure 1. A simple example of a Knowledge Base.

2. ONDUX Overview

Informally, the IETS problem consists in segmenting an implicit semi-structured data record in a way such that each segment s receives a label ℓ corresponding to an attribute a_ℓ, where s represents a value in the domain of a_ℓ. Similar to previous approaches [Agichtein and Ganti 2004, C. Zhao et. al. 2008], in ONDUX, the attributes used to label segments in the input text come from a pre-existing data source from each domain (e.g. addresses, bibliographic data, etc.), which is called a *Knowledge Base* or *KB*.

A Knowledge Base is a set of pairs $K = \{\langle a_1, O_1\rangle, \ldots, \langle a_n, O_n\rangle\}$ in which each a_i is a distinct attribute, and O_i is a set of strings $\{o_{i,1}, \ldots, o_{i,n_i}\}$ called *occurrences*. Intuitively, O_i is a set of strings representing plausible or typical values for attribute a_i.

Given a data source on a certain domain which includes values associated with fields or attributes, building a Knowledge Base is a simple process that consists in creating pairs of attributes and sets of occurrences. Example of possible data sources are: databases, reference tables, ontologies, etc. Figure 1 presents a simple example of a Knowledge Base with only four attributes: *Neighborhood*, *Street*, *Bathrooms*, and *Phone*.

The first step in an extraction process carried out with ONDUX is called *Blocking*. In this step, the input string is roughly segmented into units called *blocks*. Blocks are simply sequences of terms (words) that are likely to form a value of an attribute. This step is based on the co-occurrence of terms in a same attribute value according to the Knowledge Base. Thus, although terms in a block must all belong to a same value, a single attribute value may have terms split among two or more blocks.

Next, in a *Matching* step, blocks are matched against known attribute values, which are available in the Knowledge Base, using a small set of specific approximate matching functions. By the end of the matching step, each block is *pre-labeled* with the name of the attribute for which the best match was found.

In ONDUX, the Blocking and Matching steps alone are enough to correctly label the large majority of the segments in the input string. Indeed, experiments with different domains show that blocks are correctly labeled after these steps in more than 80% of the cases [E. Cortez et. al. 2010]. However, there are cases where the Matching steps may fail. First, mismatches can happen when distinct attributes have domains with a large intersection. For instance, when extracting from scientific paper headings, values from attributes Title and Keywords usually have terms in common. Second, unmatched blocks may occur when no matches are found for them on the Knowledge Base.

To deal with such problems, ONDUX deploys a third step called *Reinforcement* in which the pre-labeling resulting from the Matching step is reinforced by taking into

consideration the positioning and the sequencing of labeled blocks in the input texts. To accomplish this, first, a probabilistic HMM-like graph model called PSM(Positioning and Sequencing Model) is built. This model captures (i) the probability of a block labeled with ℓ appear in position p in the input text, and (ii) the probability of a block labeled with ℓ appear before a block labeled with m in the input text. Next, these probabilities are used to reinforce the pre-labeling resulting from the Labeling step, assigning labels to unmatched blocks and changing labels for blocks found to be mismatched so far.

One important point to highlight regarding ONDUX is that PSM is built without manual training, using the pre-labeling resulting from the Matching step. This implies that the model is learned *on-demand* from the input text, without requiring an *a priori* training over it.

3. The ONDUX Tool

In this section we present the main features of the ONDUX Tool. We first describe its architecture and then discuss the main aspects of its graphical user interface.

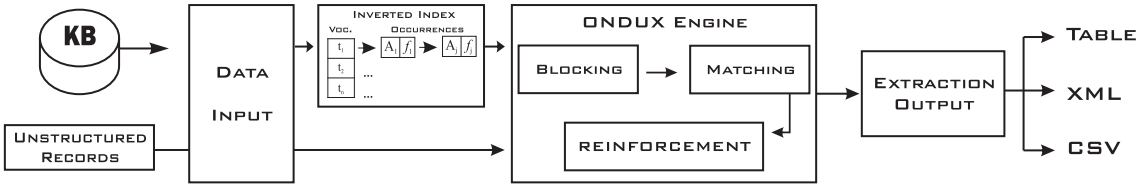


Figure 2. The architecture of the ONDUX Tool.

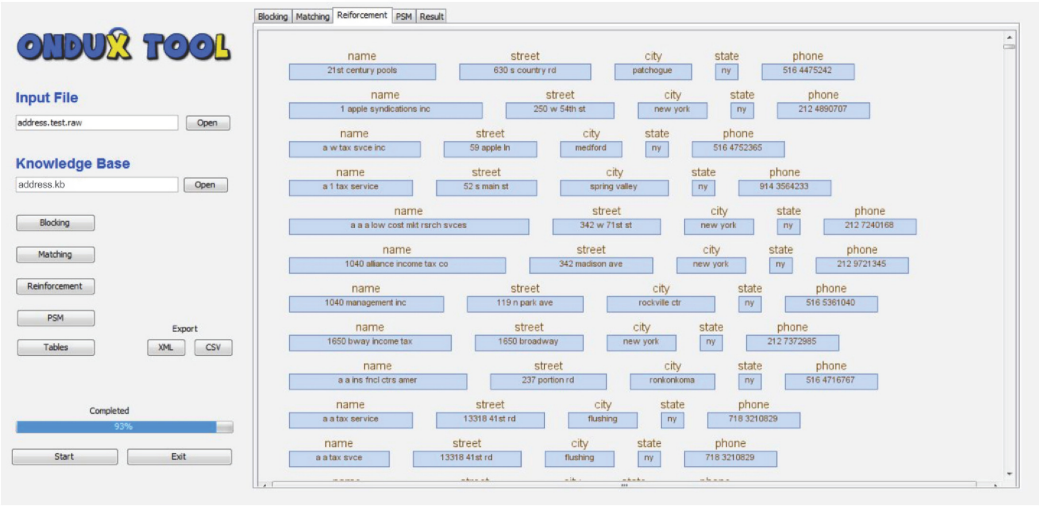


Figure 3. A Screen shot of the ONDUX Tool.

3.1. Architecture

Figure 2 illustrates the architecture of the ONDUX Tool. It consists of 3 main components: *Data Input*, *ONDUX Engine* and *Extraction Output*, which are detailed in the following.

The *Data Input* component is responsible for reading and processing two required input files: (1) a structured file containing the occurrences that compose a Knowledge Base and (2) a text file containing the unstructured records to be extracted.

```
<kb>
    <name> 21st & Century Pools </name>
    <name> Microsoft S.A. </name>
    <street> 630 S Country Rd </street>
    <street> Kennedy Avenue </street>
    <city> New York </city>
    <city> Orlando </city>
    <phone> (516) 447-5242</phone>
    <phone> (55) 92 331-7917</phone>
</kb>
```

Figure 4. Example of a Knowledge Base file.

The Knowledge Base file must follow a simple XML-based format, which is illustrated in Figure 4. In this figure, each line represents an occurrence that composes the Knowledge Base. The XML tags correspond to attribute names and the values between the tags correspond to attribute values. In this example, the Knowledge Base contains occurrences of the attributes Name, Street,City and Phone.

Besides the tasks of reading and processing the input files, the *Data Input* component builds data structures necessary to the execution of the ONDUX method. In particular, as depicted in Figure 2, an inverted index is built for processing the Knowledge Base.

The inverted index stores important information about the occurrences of each attribute. It contains a vocabulary structure that holds the distinct terms available in the Knowledge Base. Each entry of this vocabulary contains an occurrence list that stores information about the frequency of each term in a given attribute. This structure is crucial for computing the generic approximate matching functions used in the ONDUX method (See Section 2).

The *ONDUX Engine* component implements the 3 main steps of the ONDUX method: Blocking, Matching and Reinforcement (see Section 2). In the ONDUX Tool, this step rely on the output of the *Data Input* component to perform their tasks. The extraction process follows the execution sequence illustrated in Figure 2, thus, a given step can be executed only when the previous step is over.

Finally, the *Extraction Output* component is responsible for presenting the extraction result to the user and exporting it into several formats. This component takes the output of the *ONDUX Engine* component and creates views of the extraction result. As Figure 2 illustrates, the extraction results can be exported into different formats: tables, XML and CSV.

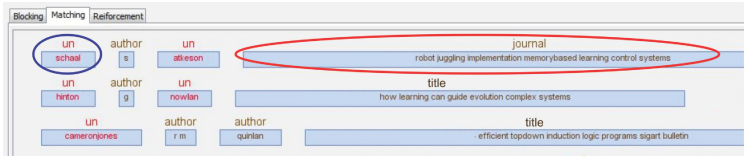
3.2. Graphical User Interface

In the ONDUX Tool, the operation of the graphical user interface (GUI) is very intuitive and simple. Figure 3 presents a screen shot of the GUI. It includes boxes for loading a file containing the Knowledge Base and the input file containing unstructured records. The GUI also features buttons for executing each step of the ONDUX method, that is, Blocking, Matching and Reinforcement. Partial results from the extraction process are presented on the screen to the user through tabs.

The Blocking tab presents the blocks resulting from the blocking step. The Matching tab presents the blocks generated in the previous step associated to labels corresponding to attributes, or identified as unmatched. Finally, the Reinforcement tab shows the final extraction result. As illustrated in Figure 3, in this last step, all blocks are associated to an attribute.

An additional tab, PSM, graphically illustrates the positioning and sequencing model (PSM) built for the current extraction process. The last tab, Result, presents the extraction result in a tabular format. Finally, the XML and CSV buttons allow the user to export the extraction result in these formats.

(a)



(b)

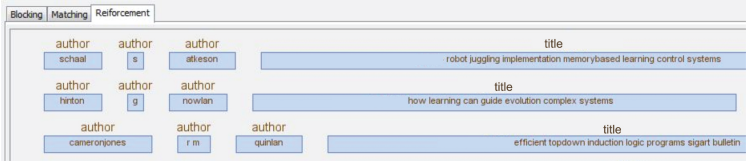


Figure 5. Matching (a) and Reinforcement (b) steps in the ONDUX Tool.

4. Case Study

In this section we present a case study in which we use the ONDUX Tool to perform an extraction process over the CORA dataset. Cora is a public dataset that contains unstructured bibliographic references. These references contain several attributes values like: author names, publication titles, page numbers, etc.

Figures 5 (a) and (b) present screen shots of the GUI when executing this extraction process. Figure 5 (a) shows the result of the Matching step, where almost all blocks were associated to an attribute. The figure also shows cases of blocks that were wrongly labeled and blocks that received the label “un”, meaning that these blocks were left unmatched.

The result of the Reinforcement step is depicted in Figure 5 (b). Now, all blocks are associated to an attribute (i.e., there is no unmatched blocks), and, as illustrated, blocks that were wrongly labeled in the Matching step are now correctly labeled.

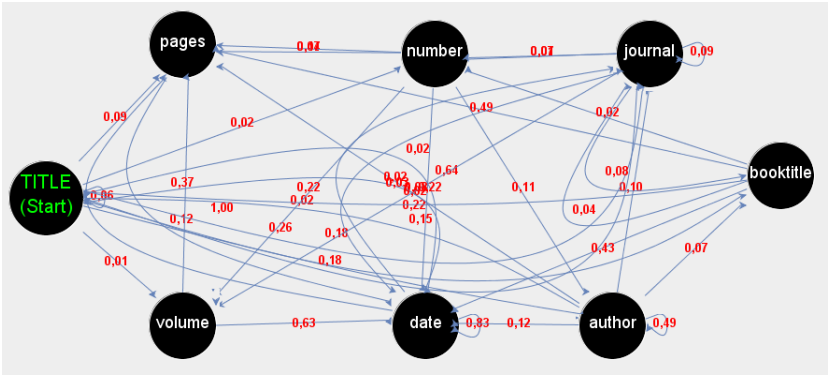


Figure 6. Graphical illustration of the Positioning and Sequencing Model (PSM1).

As explained in Section 2, the Reinforcement step relies on the Positioning and Sequencing Model (PSM). Figure 6 shows a graphical visualization of the PSM generated by the tool for this extraction task. As already mentioned, this visualization is available on the PSM tab in the tool. In the graph shown, each vertex represents an attribute and the edges represent transition probabilities.

5. Demo Outline

During the demo session, we will demonstrate the features of the ONDUX Tool using different datasets from distinct domains. These datasets are listed in Table 1. At the demonstration site, we will give attendees the opportunity to experience the ONDUX Tool and show them how to use it to perform real case information extraction tasks. Furthermore, attendees will have the opportunity to explore different perspectives of the ONDUX method. The demo will run on a notebook with all data stored locally.

Dataset	Domain	Text Inputs	# Attrib.
BigBook	Postal Addresses	2000	5
CORA	Bibliographic References	500	3 to 7
Web Ads	Classified Ads	500	5 to 18
Recipes	Cooking Recipes	500	3
Products	Product Offers	10000	3

Table 1. Datasets to be used in the demonstration.

6. Conclusions

In this demo we presented a tool that implements ONDUX, an unsupervised probabilistic method for information extraction by text segmentation. ONDUX relies on pre-existing data, more specifically, on sets of attributes values from pre-existing data sources, to associate segments in the input string with a given attribute. The ONDUX Tool allows non expert users to easily perform information extraction tasks and export the extraction result in different formats.

Acknowledgements

This work is partially supported by INWeb (MCT/CNPq grant 57.3871/2008-6), by project MinGroup (CNPq grant 575553/2008-1), by UOL Bolsa Pesquisa program (grant 20110212103900), and by the authors' individual grants and scholarships from CNPq, CAPES and FAPEAM.

References

Agichtein, E. and Ganti, V. (2004). Mining Reference Tables for Automatic Text Segmentation. In *Proc. 10th ACM SIGKDD Intl. Conf. on Knowl. Discov. and Data Mining*, pages 20–29.

C. Zhao et. al. (2008). Exploiting Structured Reference Data for Unsupervised Text Segmentation with Conditional Random Fields. In *Proc. SIAM Intl. Conf. on Data Mining*, pages 420–431.

E. Cortez et. al. (2010). ONDUX: On-Demand Unsupervised Learning for Information Extraction. In *Proc. ACM SIGMOD Intl. Conf. on Management of Data*, pages 807–818.

Mansuri, I. R. and Sarawagi, S. (2006). Integrating Unstructured Data into Relational Databases. In *Proc. 22nd Intl. Conf. on Data Engineering*, page 29.

V. Borkar et. al. (2001). Automatic Segmentation of Text into Structured Records. In *Proc. ACM SIGMOD Intl. Conf. on Management of Data*, pages 175–186.