Knowl Inf Syst (2016) 49:197–241
DOI 10.1007/s10115-016-0921-4

REGULAR PAPER

Roller: a novel approach to Web information extraction

Patricia Jiménez[1] · **Rafael Corchuelo**[1]

Received: 4 May 2015 / Revised: 11 November 2015 / Accepted: 3 February 2016 /
Published online: 10 March 2016

Abstract The research regarding Web information extraction focuses on learning rules to extract some selected information from Web documents. Many proposals are ad hoc and cannot benefit from the advances in machine learning; furthermore, they are likely to fade away as the Web evolves, and their intrinsic assumptions are not satisfied. Some authors have explored transforming Web documents into relational data and then using techniques that got inspiration from inductive logic programming. In theory, such proposals should be easier to adapt as the Web evolves because they build on catalogues of features that can be adapted without changing the proposals themselves. Unfortunately, they are difficult to scale as the number of documents or features increases. In the general field of machine learning, there are propositio-relational proposals that attempt to provide effective and efficient means to learn from relational data using propositional techniques, but they have seldom been explored regarding Web information extraction. In this article, we present a new proposal called Roller: it relies on a search procedure that uses a dynamic flattening technique to explore the context of the nodes that provide the information to be extracted; it is configured with an open catalogue of features, so that it can adapt to the evolution of the Web; it also requires a base learner and a rule scorer, which helps it benefit from the continuous advances in machine learning. Our experiments confirm that it outperforms other state-of-the-art proposals in terms of effectiveness and that it is very competitive in terms of efficiency; we have also confirmed that our conclusions are solid from a statistical point of view.

Keywords Web information extraction · Knowledge and data engineering · Software information systems · Propositio-relational learning · Dynamic flattening

✉ Patricia Jiménez
patriciajimenez@us.es

Rafael Corchuelo
corchu@us.es

[1] ETSI Informática, University of Sevilla, Avda. Reina Mercedes, s/n, 41012 Sevilla, Spain

1 Introduction

The Web is an enormous, growing source of information that is provided by a variety of websites. Most of them are intended for human consumption, and many of them do not provide any means to extract their information automatically, which makes it difficult to feed automated business processes [5,46]. This has triggered significant interest in automating Web information extraction [11,52,56,62].

Our focus is on supervised rule-based information extractors that work on semi-structured Web documents. They rely on a generic algorithm that executes extraction rules that are learnt from annotated documents that are provided by a user. The documents are written in HTML, and the information of interest is buried into formatting tags that specify how to render it in list, tabular, or other such regular formats. The documents must be annotated, which means that a user must have labelled the pieces of text or the nodes to be extracted; a machine learner is then applied to learn a rule that generalises the annotations and is expected to work as accurately as possible on new unseen documents.

Most existing proposals build on ad hoc machine-learning techniques that were specifically tailored to the problem of extracting Web information [1,2,10,12–14,32,34,38,44,49,50,53, 55,58,61]. This makes it difficult to adapt them as the Web evolves because the features of the documents on which they rely and the techniques used to analyse them are built-in. This, in turn, implies that they cannot easily benefit from the many advances in machine learning. This has made Web information extraction quite an active research field for years.

A few authors have explored representing documents as relational data [8,9,20,24,35,59], that is, as collections of vectors that represent the attributive features of the pieces of text or the nodes to be extracted, e.g., HTML tag, rendering coordinates, or ratio of letters, plus a number of relational features that help establish a neighbourhood around each vector, e.g., right token, left sibling, or parent node. They all have devised proposals that got inspiration from inductive logic programming, that is, they learn first-order rules that constraint the attributive features of the information to be extracted as well as the attributive features of the context that can be reached by means of relational features. These proposals can learn very expressive rules that are very effective [37], but the learning process is usually costly and degrades as the number of examples or features increases [7,8,23,24,48].

In the general field of machine learning, there are a number of so-called propositio-relational proposals [40] that can deal with relational data using existing propositional techniques, that is, techniques that were originally designed to work on attributive features only. Unfortunately, such proposals have seldom been explored regarding learning information extraction rules; the only exception is the work by Sleiman and Corchuelo [57], who introduced an approach that combines automata and neural networks.

In this article, we introduce Roller, which is a proposal to learn web information extraction rules. Our contributions to the field are the following: we have devised a new propositio-relational technique that relies on a search procedure that uses a dynamic flattening technique to explore the context of the nodes that provide the information to be extracted; it needs to be configured with an open catalogue of features, which helps it adapt as the Web evolves, plus a base learner and a rule scorer, which helps it leverage the continuous advances in the general field of machine learning. We have conducted an extensive experimental analysis that proves that our proposal outperforms other state-of-the-art proposals regarding effectiveness; regarding efficiency, our results prove that it is comparable to the best ones. The conclusions that we have drawn from our experimental analysis have been confirmed using standard statistical hypothesis tests in the literature.

The rest of the article is organised as follows: Sect. 2 describes the details of our proposal; Sect. 3 reports on how we have configured it so that it can achieve its best results; then, the results of our experimental analysis are presented in Sect. 4; Sect. 5 presents the related work and a detailed comparison with our proposal; Sect. 6 summarises our conclusions. "Appendices 1 and 2" report, respectively, on our experimentation environment and the performance measures that we have used.

2 Description of our proposal

In this section, we describe Roller, which is our proposal to learn rules that can be used to extract information from semi-structured web documents. Such documents are written in HTML and can then be naturally represented as DOM nodes. For the sake of brevity, we use terms document and node to refer to the previous concepts, since there are not any ambiguities.

Our proposal works on a set of documents and an annotation. The documents provide examples of how the information to extract is encoded and the annotation assigns each of their nodes to a slot that classifies the information that it provides. (There is an implicit null slot to which the nodes that do not provide any information to be extracted are assigned by default.) The documents are assumed to provide information on a given topic and to have regularities that help learn the rule.

The main algorithm first computes a number of attributive and relational features on the input documents. Such features are not intrinsic to our proposal; on the contrary, we assume that the user provides a procedure called FEATUREBUILDER to compute them; in other words, our proposal relies on an open catalogue of features that allows it to evolve as the Web evolves. The attributive features are then used to assemble a training set from which a rule is learnt. Neither is the base learner used intrinsic to our proposal; on the contrary, any technique in the literature that can work with multi-class problems using both numeric and categoric features can be plugged into our proposal using a user-provided procedure to which we refer to as BASELEARNER. The initial rule is then evaluated on the previous training set using a user-defined rule scorer to which we refer to as RULESCORER. The main algorithm in Roller loops as long as a perfect rule is not found and the current rule can be expanded to a new rule that provides some score gain. The expansion procedure explores the context of every node, that is, the neighbouring nodes according to the available relational features, and then selects the one whose attributive features help learn a better rule.

Formally speaking, the problem that we address can be formulated as follows:

Requirements (a) a procedure called FEATUREBUILDER to compute a catalogue of features from a set of documents; (b) a procedure called BASELEARNER to learn a rule from a multi-class training set in which features can be both numeric and categoric; (c) a procedure called RULESCORER that returns a score for a rule based on how well it performs in a test set.
Inputs (a) a set of documents D; (b) an annotation A.
Assumptions (a) the documents in D have a regular structure; (b) if a node is not included in annotation A, then it is implicitly assumed to belong to the null slot.
Problem find a rule r that characterises the information to be extracted using some attributive features of a subset of nodes that are related by means of some relational features; that rule must have the best possible score.

Regarding the requirements, we have performed quite an exhaustive experimentation from which we have drawn the following conclusions: a good catalogue of features must include the following attributive features: the standard W3C HTML features, the standard W3C rendering features, and user-defined features to characterise the contents of the information to be extracted; it must also include the standard W3C DOM features to fetch the neighbours of every node; furthermore, we have found that JRip and Kappa seem to be best combination of base leaner and rule scorer. The previous recommendations can be used as a default to configure Roller. Consult Sect. 3 for further details on our catalogue of features and how to select the best combination of base learner and rule scorer.

In the following subsections, we first present the notation and the core concepts that we use, then introduce the main procedures in our proposal, and, finally, describe some ancillary procedures to deal with training sets and feature vectors.

2.1 Notation and core concepts

We use the standard mathematical notation to represent variables, sets, and logical formulae. There are only a few pieces of notation for which there is not a standard in the literature, namely: given elements $x_1, x_2, \ldots, x_n$, then $\langle x_1, x_2, \ldots, x_n \rangle$ denotes a sequence of them; given two sequences s_1 and s_2, we denote their concatenation as $s_1 \oplus s_2$; we denote the tuples of which a map is composed as $\{x \mapsto y\}$; given a map M, we denote its domain as dom M and its range as ran M; maps are applied using the usual functional notation, e.g., $y = M(x)$; given a map M, we denote its inverse as M^{-1}.

Next, we define the core concepts of our proposal, namely: documents, nodes, features, annotations, slots, contexts, bindings, datasets, rules, base learners, and rule scorers.

Definition 1 (*Documents and nodes*) Documents are character strings that adhere to the HTML syntax and can then be represented as DOM nodes [31,63].

Example 1 Figure 1 illustrates a collection with documents $\{d_1, d_2, d_3\}$. We show a partial view of document d_1, which we use as a running example through the rest of this section. The set of nodes includes $\{n_1, n_2, \ldots, n_{15}\}$, plus the children of the head element and the nodes that correspond to documents d_2 and d_3, which are not shown. Please, note that this

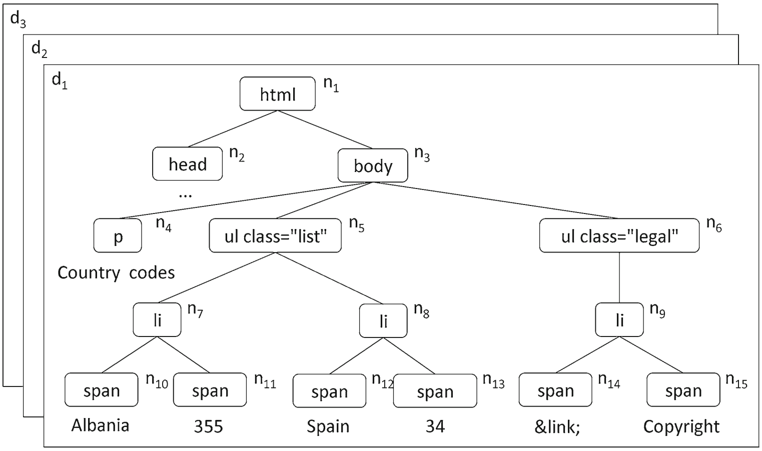


Fig. 1 Sample documents

Table 1 Sample features

Node	Tag	Class	y-pos	x-pos	Len	Is-number	Node	Parent	Left	Right	Child
(a) *Sample attributive features*							(b) *Sample relational features*				
n_1	html	Null	0	0	8	False	n_1	$\{\}$	$\{\}$	$\{\}$	$\{n_2, n_3\}$
n_2	head	Null	0	0	0	False	n_2	$\{n_1\}$	$\{\}$	$\{n_3\}$	$\ldots$
n_3	body	Null	0	0	8	False	n_3	$\{n_1\}$	$\{n_2\}$	$\{\}$	$\{n_4, n_5, n_6\}$
n_4	p	Null	0	0	8	False	n_4	$\{n_3\}$	$\{\}$	$\{n_5\}$	$\{\}$
n_5	ul	List	16	0	4	False	n_5	$\{n_3\}$	$\{n_4\}$	$\{n_6\}$	$\{n_7, n_8\}$
n_6	ul	Legal	48	0	2	False	n_6	$\{n_3\}$	$\{n_5\}$	$\{\}$	$\{n_9\}$
n_7	li	Null	16	0	2	False	n_7	$\{n_5\}$	$\{\}$	$\{n_8\}$	$\{n_{10}, n_{11}\}$
n_8	li	Null	32	0	2	False	n_8	$\{n_5\}$	$\{n_7\}$	$\{\}$	$\{n_{12}, n_{13}\}$
n_9	li	Null	48	0	2	False	n_9	$\{n_6\}$	$\{\}$	$\{\}$	$\{n_{14}, n_{15}\}$
n_{10}	span	Null	16	0	1	False	n_{10}	$\{n_7\}$	$\{\}$	$\{n_{11}\}$	$\{\}$
n_{11}	span	Null	16	100	1	True	n_{11}	$\{n_7\}$	$\{n_{10}\}$	$\{\}$	$\{\}$
n_{12}	span	Null	32	0	1	False	n_{12}	$\{n_8\}$	$\{\}$	$\{n_{13}\}$	$\{\}$
n_{13}	span	Null	32	100	1	True	n_{13}	$\{n_8\}$	$\{n_{12}\}$	$\{\}$	$\{\}$
n_{14}	span	Null	48	0	1	False	n_{14}	$\{n_9\}$	$\{\}$	$\{n_{15}\}$	$\{\}$
n_{15}	span	Null	48	25	1	False	n_{15}	$\{n_9\}$	$\{n_{14}\}$	$\{\}$	$\{\}$

example is fictitious because it is not possible to show an actual collection of documents due to space constraints, but it is enough for illustration purposes.

Definition 2 (*Attributive and relational features*) An attributive feature is a function that maps a node onto a value that represents either an HTML attribute [31], which is specified in the HTML code of a document, a rendering attribute [63], which is computed by a browser, or a user-defined attribute. A relational feature is a function that maps a node onto a set of nodes with which the former is related by means of a neighbouring relationship.

Example 2 Table 1 illustrates some of the features of the nodes of which the documents in Figure 1 are composed. $node$ represents the node being examined; tag and $class$ represent its HTML tag and its CSS class, respectively; y-pos and x-pos represent the ordinate and the abscissa of the corresponding rendering box, respectively; len and is-$number$ represent the number of tokens in the text that is associated with the node and whether it is a number or not, respectively.

Definition 3 (*Annotations and slots*) An annotation is a function that maps a node onto a slot. A slot is a label that provides a meaning to the nodes with which it is associated. There is a special slot called $null$ that indicates that a node does not provide any information to be extracted. The nodes that belong to the $null$ slot are referred to as negative examples and the others as positive examples.

Example 3 Table 2 presents the annotation that corresponds to document d_1 in Fig. 1. The set of slots is $\{Record, country, code, null\}$, where $Record$ labels the records to be extracted, which are composed of a country name that is denoted as $country$ and a phone code that is denoted as $code$.

Table 2 Sample annotation

Node	Slot	Node	Slot	Node	Slot
n_1	Null	n_6	Null	n_{11}	Code
n_2	Null	n_7	Record	n_{12}	Country
n_3	Null	n_8	Record	n_{13}	Code
n_4	Null	n_9	Null	n_{14}	Null
n_5	Null	n_{10}	Country	n_{15}	Null

Definition 4 (*Contexts and bindings*) A context is a sequence of tuples of the form (t, rf, s), where t denotes a target variable, rf denotes a relational feature, and s denotes a source variable. If s and rf are not $null$, then it binds t to the result of applying relational feature rf to s, that is, $t = rf(s)$; if both rf and s are $null$, then it is an initial context tuple that indicates that t is bound to the set of all of the nodes in the input documents. Simply put, a context is a symbolic representation of a binding; the binding itself is a map in which the variables in a context are bound to their corresponding nodes.

Example 4 Regarding the documents in Fig. 1, context $\langle(node_0, null, null), (node_1, parent, node_0)\rangle$ sets variable $node_0$ to the nodes in the input documents and then variable $node_1$ to their parents. The corresponding binding is the following: $\{node_0 \mapsto \{n_1, n_2, \ldots, n_{15}, \ldots\}, node_1 \mapsto \{n_1, n_3, n_5, n_6, n_7, n_8, n_9, \ldots\}\}$.

Definition 5 (*Datasets and rules*) A dataset is a function that maps every node in the input documents onto a vector with its attributive features and the attributive features of some neighbours (possibly none); such neighbours are fetched by means of relational features, and they are introduced by means of a context. The datasets that are used to learn rules are referred to as training sets and the datasets that are used to score rules are referred to as test sets. A rule is a function that maps a vector onto a slot that is expected to provide its correct meaning.

Example 5 Table 3a shows a sample dataset in which the context involves the nodes in the input documents and their parents. Columns $node_0$ and $node_1$ present the corresponding bindings. Table 3b shows a sample rule that was learnt from the previous dataset; it is of the form $\langle r_1, r_2, \ldots, r_n\rangle$ $(n \geq 1)$, where each component r_i is of the form $c_{i,1} \wedge c_{i,2} \wedge \cdots \wedge c_{i,k_i} \Rightarrow slot = s$ $(i = 1 \,..\, n, k_i \geq 0)$; each $c_{i,j}$ is a simple condition of the form $n.f\ \theta\ v$, where n denotes a target variable in a context, f denotes an attributive feature, θ is a comparator, and v is a value. Given a node to classify, it is first transformed into its corresponding vector, and then the components of the rule are applied in sequence; the last component assigns a default slot to nodes that cannot be better classified by the previous components.

Definition 6 (*Base learners and rule scorers*) A base learner is a procedure that takes a training set and an annotation as input and returns a rule. A rule scorer is a function that takes a rule and a test dataset and returns a value that indicates how good the rule is at predicting the correct slot for each node in the dataset. The scores must be normalised in range $[0.00 \,..\, 1.00]$ so that the closer to the lower bound, the worse, and the closer to the upper bound, the better.

Example 6 Regarding the rule in Table 3b, a rule scorer might score it at 0.94 when it is evaluated on the dataset in Table 3a. Note that it is not generally possible to assess a score in isolation unless it is 0.00 or 1.00; that is, a score of 0.94 does not mean that the rule works well in 94 % of the examples to which it is applied or something like that; it simply means that a rule that scores at, say, 0.90 is worse and a rule that scores at, say, 0.98 is better.

Table 3 Sample dataset and rule

Node$_0$	Node$_0$						Node$_1$	Node$_1$ = parent(node)					
	Tag	Class	y-pos	x-pos	Len	Is-number		Tag	Class	y-pos	x-pos	Len	Is-number
(a) *Sample dataset*													
n_1	html	Null	0	0	8	False	Null	null	Null	Null	Null	Null	Null
n_2	Head	Null	0	0	0	False	n_1	html	Null	0	0	8	False
n_3	body	Null	0	0	8	False	n_1	html	Null	0	0	8	False
n_4	p	Null	0	0	8	False	n_3	body	Null	0	0	8	False
n_5	ul	List	16	0	8	False	n_3	body	Null	0	0	8	False
n_6	ul	Legal	48	0	8	False	n_3	body	Null	0	0	8	False
n_7	li	Null	16	0	2	False	n_5	ul	List	16	0	8	False
n_8	li	Null	32	0	2	False	n_5	ul	List	16	0	8	False
n_9	li	Null	48	0	2	False	n_6	ul	Legal	48	0	8	False
n_{10}	span	Null	16	0	1	False	n_7	li	Null	16	0	2	False
n_{11}	span	Null	16	100	1	True	n_7	li	Null	16	0	2	False
n_{12}	span	Null	32	0	1	False	n_8	li	Null	32	0	2	False
n_{13}	span	Null	32	100	1	True	n_8	li	Null	32	0	2	False
n_{14}	span	Null	48	0	1	False	n_9	li	Null	48	0	2	False
n_{15}	span	Null	48	25	1	False	n_9	li	Null	48	0	2	False

(b) Sample rule

$\langle$

$node_0.x\text{-}pos \geq 100 \Rightarrow slot = code,$

$node_0.tag = span \wedge node_1.y\text{-}pos \geq 16 \wedge node_1.y\text{-}pos \leq 32 \Rightarrow slot = country,$

$node_1.y\text{-}pos \leq 0 \Rightarrow slot = null,$

$node_1.y\text{-}pos \geq 48 \Rightarrow slot = null,$

$\Rightarrow slot = Record$

$\rangle$

```
$Roller(D, A)$
    – Step 1: compute features.
    $(AF, RF) = \textsc{featureBuilder}(D)$
    – Step 2: learn an initial rule.
    $c = (node_0, null, null)$
    $C = \langle c \rangle$
    $B = \{node_0 \mapsto \mathrm{dom}\ A\}$
    $TS = createTrainingSet(c, AF, RF, A)$
    $r = \textsc{baseLearner}(TS, A)$
    – Step 3: find an expansion.
    $keepSearching = (\textsc{ruleScorer}(r, TS) \neq 1.00)$
    while $keepSearching$ do
        $(C, B, TS, r') = findExpansion(C, B, TS, r, AF, RF, A)$
        $keepSearching = (r' \neq r \wedge \textsc{ruleScorer}(r', TS) \neq 1.00)$
        $r = r'$
    end
return $(C, r)$
```

Fig. 2 Roller's main procedure

2.2 Main procedures

Figure 2 presents the main procedure in Roller. It works on a set of documents *D* and an annotation *A*; it returns a tuple (C, r), where *C* is a context and *r* is a rule. It consists of three steps that run in sequence, namely:

- The first step consists in computing the attributive and the relational features of the nodes of which the input documents are composed. This is performed by a user-provided procedure called FEATUREBUILDER, which works on a set of documents *D* and returns a tuple (AF, RF), where *AF* denotes the set of attributive features and *RF* the set of relational features that it has computed. It is quite a simple procedure from a conceptual point of view: it loads the input documents, parses them into DOM trees, iterates over the resulting nodes, and computes the features that are provided in a catalogue. Note, however, that it is a little more involved from a technology point of view since it requires to interact with DOM-specific, browser-specific, and user-defined APIs to compute the HTML, the rendering, and the user-defined features, respectively. Such technology details are out of the scope of this article, in which our focus is on presenting the proposal, not on delving into the technology intricacies to implement it. Working on tuples of the form (AF, RF, A), where *AF* is a set of attributive features, *RF* is a set of relational features, and *A* is an annotation, is very common in our proposal; for the sake of brevity, we refer to such tuples as input configurations.
- The second step consists in learning an initial rule building, exclusively, on the attributive features of the nodes in the input documents. To do so, we have to create an initial context of the form $C = \langle c \rangle$, where $c = (node_0, null, null)$. The corresponding binding *B* simply maps variable $node_0$ onto the set of all nodes in the input documents, which can be very easily computed from the domain of the input annotation *A*. Then, a training set *TS* is created; it maps every node bound to $node_0$ in the initial context tuple *c* onto a vector that represents its attributive features. The base learner is finally invoked on training set *TS* and the input annotation *A* in order to learn a rule *r*. Working on tuples of the form (C, B, TS, r), where *C* denotes a context, *B* its corresponding binding, *TS* is a training

```
$findExpansion(C, B, TS, r, AF, RF, A)$
    – Step 1: initialise rule configuration.
    $C^* = C;\ B^* = B;\ TS^* = TS;\ r^* = r$
    $g^* = 0.00$
    – Step 2: explore candidate expansions.
    for each $(c, rf) \in C \times RF$ as long as $\textsc{ruleScorer}(r^*, TS^*) \neq 1.00$ do
        $c' = $ (a new variable, $rf$, target of $c$)
        if $\neg redundant(c', C)$ then
            – Step 2.1: expand rule configuration.
            $C' = C \oplus \langle c' \rangle$
            $B' = B \cup \bigcup \{rf(n) \mid n \in B(\text{source of } c')\}$
            $TS' = expandTrainingSet(TS, c', B', AF, RF, A)$
            $r' = \textsc{baseLearner}(TS', A)$
            – Step 2.2: save the expanded rule configuration.
            $g' = \textsc{ruleScorer}(r', TS') - \textsc{ruleScorer}(r, TS)$
            if $g' > g^*$ then
                $C^* = C';\ B^* = B';\ TS^* = TS';\ r^* = r'$
                $g^* = g'$
            end
        end
    end
return $(C^*, B^*, TS^*, r^*)$
```

Fig. 3 Procedure to find an expansion

set for context C, and r a rule that was learnt from that training set is very common in our proposal; for the sake of brevity, we refer to such tuples as rule configurations.

- The third step consists in finding an expansion, which is a term that we use to refer to a rule configuration that results from exploring some neighbours of the nodes in the context of the best rule configuration found so far. Ideally, such expansion should provide a rule that scores better than the current rule. To achieve such a goal, we combine the attributive features of the nodes in the current context with the attributive features of the nodes that are explored in the expansion. If an expansion that achieves a better score is found, this step is repeated again; otherwise, it stops and the procedure returns the best rule found and its associated context.

The procedure to find an expansion is presented in Fig. 3. It works on the rule configuration (C, B, TS, r) that corresponds to the best rule found so far and an input configuration (AF, RF, A); it returns a rule configuration (C^*, B^*, TS^*, r^*) such that r^* improves or equals the score achieved by r. It consists of the following steps:

- The first step initialises a rule configuration of the form (C^*, B^*, TS^*, r^*) to the input rule configuration; the procedure searches for candidate expansions and stores the best one that it finds in this starred rule configuration. The criterion used to determine whether an expanded configuration is better than another is based on the score achieved by the corresponding rule; we use variable g^* to save the score gain that is achieved when the current rule is replaced by the best expansion found so far. It is initialised to 0.00 because the first configuration coincides with the input configuration.
- The second step is a loop that explores candidate expansions. It iterates over the set of pairs (c, rf) of the Cartesian product of the context tuples in C and the relational features in RF, as long as the best expansion found is not perfect. For each such pair, a new context

tuple of the form (x, rf, y) is created, where x denotes a new variable that is not used in context C and y denotes the target variable in context tuple c; simply put, the new context tuple binds a new variable to the result of applying relational feature rf to the nodes that are currently bound to the target of context tuple c. This allows to explore the neighbourhood of every node in the current context C. Note, however, that only context tuples that are not redundant with regard to the current context must be explored. Such context tuples are then used to create a new rule configuration (C', B', TS', r'). The score gain of r' with respect to the input rule r is then computed; if it is greater than the score gain of the best expansion found so far, then it means that the new expansion must be saved since it has resulted in a better rule. This step iterates until a perfect rule is found or the whole Cartesian product is explored; in both cases, the best rule configuration found is returned.

The check for redundancy is implemented by means of predicate $redundant$, which given a context tuple (t, rf, s) and a context C holds as long as there is a context tuple (t', rf', s') in C such that $s = s'$ and $rf = rf'$ or $s' = t$ and $rf' = rf^{-1}$. The first condition is trivial since it amounts to saying that context tuples (t, rf, s) and (t', rf, s), where $t \neq t'$, are redundant because they bind the same nodes to different variables, which does not help explore new neighbours. The second condition is a little more involved; it amounts to saying that context tuples (t, rf, s) and (t', rf^{-1}, t), where $t \neq t'$, are redundant because the second one binds t' to the same nodes that are bound to s. Formally speaking, rf is the inverse of rf' if $\forall n_1 \cdot rf_1(n_1) = N \Rightarrow \{n_1\} = \bigcup\{rf_2(n_2) \mid n_2 \in N\}$.

Example 7 Assume that Roller is executed on the input documents and the annotation that are sketched in Fig. 1 and Table 2, respectively. It first uses the user-provided FEATUREBUILDER procedure to compute the sets of attributive and relational features that are sketched in Table 1. These features and the annotation are used to create an initial training set that corresponds to context tuple $(node_0, null, null)$, which is sketched in Table 3a. Note that the previous figure sketches a training set that corresponds to two context tuples, namely, the initial context tuple $(node_0, null, null)$ and another context tuple that explores the parents of the nodes that are bound to $node_0$, that is, $(node_1, parent, node_0)$. The initial training set corresponds to the part of the figure that refers to the initial context tuple. If we apply a base learner to learn a rule from this training set, then we might get the following rule:

$$\begin{array}{l}\langle\\ \quad node_0.tag = span \wedge node_0.x\text{-}pos \leq 0 \Rightarrow slot = country,\\ \quad node_0.len \geq 8 \Rightarrow slot = null,\\ \quad node_0.y\text{-}pos \geq 48 \Rightarrow slot = null,\\ \quad node_0.tag = span \Rightarrow slot = code,\\ \quad \Rightarrow Slot = Record\\ \rangle,\end{array}$$

which is scored at 0.90. That means that it is reasonably good at classifying each node in the input documents into the appropriate slot, but it is not perfect. Thus, it makes sense to explore the neighbouring nodes in order to find out if there is one whose attributive features can contribute to producing a better rule. Since the context currently has the initial context tuple $(node_0, null, null)$ only and the relational features are $parent$, $left$, $right$, and $child$, then the procedure to find an expansion has to explore the following additional contexts:

$$\begin{gathered}\langle (node_0, null, null), (node_1, parent, node_0)\rangle,\\ \langle (node_0, null, null), (node_1, left, node_0)\rangle,\\ \langle (node_0, null, null), (node_1, right, node_0)\rangle,\\ \langle (node_0, null, null), (node_1, child, node_0)\rangle.\end{gathered}$$

Exploring the first context amounts to creating a new training set in which the attributive features of each node are combined with the attributive features of their parents. In this case, the resulting training set is sketched in Table 3a. If the base learner is applied to this training set, then we might get the following new rule:

$$\begin{array}{l}\langle\\ \quad node_0.x\text{-}pos \geq 100 \Rightarrow slot = code,\\ \quad node_0.tag = span \wedge node_1.y\text{-}pos \geq 16 \wedge node_1.y\text{-}pos \leq 32 \Rightarrow slot = country,\\ \quad node_1.y\text{-}pos \leq 0 \Rightarrow slot = null,\\ \quad node_1.y\text{-}pos \geq 48 \Rightarrow slot = null,\\ \quad \Rightarrow slot = Record\\ \rangle,\end{array}$$

which is scored at 0.94. Exploring the remaining context tuples results in similar rules, none of which is scored better. That means that we now have to explore the following contexts:

$$\begin{gathered}\langle (node_0, null, null), (node_1, parent, node_0), (node_2, left, node_0)\rangle,\\ \langle (node_0, null, null), (node_1, parent, node_0), (node_2, right, node_0)\rangle,\\ \langle (node_0, null, null), (node_1, parent, node_0), (node_2, child, node_0)\rangle,\\ \langle (node_0, null, null), (node_1, parent, node_0), (node_2, parent, node_1)\rangle,\\ \langle (node_0, null, null), (node_1, parent, node_0), (node_2, left, node_1)\rangle,\\ \langle (node_0, null, null), (node_1, parent, node_0), (node_2, right, node_1)\rangle.\end{gathered}$$

Note that there are two contexts that need not be explored, namely: context $\langle (node_0, null, null), (node_1, parent, node_0), (node_2, parent, node_0)\rangle$ is not explored because it does not provide any additional data to the training set and would result in the same rule; context $\langle (node_0, null, null), (node_1, parent, node_0), (node_2, child, node_1)\rangle$ is not explored because context tuples $(node_1, parent, node_0)$ and $(node_2, child, node_1)$ are redundant because relational feature $child$ is the inverse of relational feature $parent$, so exploring it would result in the same rule because $node_2 = node_0$; that is, both $node_2$ and $node_0$ would be bound to exactly the same nodes. Note, however, that the new contexts are allowed to include context tuples that have been explored previously; for instance, a context of the form $\langle (node_0, null, null), (node_1, parent, node_0), (node_2, left, node_0)\rangle$ explores the left sibling of every node again, but in a different context: previously, we explored the nodes and their left siblings and now we explore the nodes, their parents, and their left siblings together.

In this case, the context that results in the best rule is $\langle (node_0, null, null), (node_1, parent, node_0), (node_2, parent, node_1)\rangle$, namely:

$$\begin{array}{l}\langle\\ \quad node_0.tag = span \wedge node_0.x\text{-}pos \geq 100 \Rightarrow slot = code,\\ \quad node_0.tag = span \wedge node_0.x\text{-}pos \leq 0 \wedge node_2.class = list \Rightarrow slot = country,\\ \quad node_0.tag = li \wedge node_0.y\text{-}pos \leq 32 \Rightarrow slot = Record,\\ \quad \Rightarrow slot = null\\ \rangle.\end{array}$$

This rule is scored at 1.00, which means that it is a perfect rule, that is, it assigns every node in the training set to the correct slot. So the search for a rule finishes here. Note that

```
$createTrainingSet(c, AF, RF, A)$
    $TS = \emptyset$
    $N = \operatorname{dom} A$
    for $n \in N$ do
        $v = computeVector(n, AF, c)$
        $TS = TS \cup \{n \mapsto \{v\}\}$
    end
return $TS$
```

(a) Creating training sets.

```
$expandTrainingSet(TS, c, B, AF, RF, A)$
    $TS' = \emptyset$
    for $\{n \mapsto V\} \in TS$ do
        $V' = expandVectors($
                $n, V, c, B, AF, RF, A)$
        $TS' = TS' \cup \{n \mapsto V'\}$
    end
return $TS'$
```

(b) Expanding training sets.

Fig. 4 Procedures to deal with training sets

the resulting rule takes into account nodes $node_0$ and $node_2$ only; $node_1$ was used just to reach $node_2$, but its attributive features do not provide any classification power in this example.

2.3 Working with training sets

Training sets associate nodes with the vectors that describe their attributive features within a given context. Figure 4 presents the two ancillary procedures that we propose to deal with training sets.

The first procedure is *createTrainingSet*, which works on an initial context tuple c and an input configuration (AF, RF, A); it returns a training set in which every node in the domain of the annotation is mapped onto a singleton that provides its representation as a vector. Initially, every node is associated with a unique vector, but if the training set is expanded using a multi-valued relational feature, then the initial vectors need to be combined with the vectors that correspond to several neighbours. This is the reason why training sets associate nodes with sets of vectors.

The second procedure is *expandTrainingSet*, which works on a training set TS, a context tuple c, a binding B, and an input configuration (AF, RF, A); it returns a training set in which every node in TS is mapped onto a set of expanded vectors that represent the attributive features that are already present in training set TS plus the attributive features that correspond to the nodes bound in B by context tuple c.

Example 8 Table 3a illustrates a training set that is created from the attributive features in Table 1a. The initial training sets consists of the vectors that correspond to context tuple $(node_0, null, null)$; the same figure illustrates how this training set is expanded to take into account the features of the parents of every node.

2.4 Working with vectors

Vectors represent the attributive features of a subset of nodes in the input documents in a format that is suitable to learn a rule using a propositional base learner. We need two ancillary procedures to deal with them, which are presented in Fig. 5.

The first procedure is *computeVectors*. It works on a node n, a set of attributive features AF, and a context tuple c. It computes a vector that is implemented as a map in which each attributive feature is associated with its corresponding value on node n regarding context c.

```
$computeVector(n, AF, c)$
    $v = \emptyset$
    for $af \in AF$ do
        $v = v \cup \{(c, af) \mapsto af(n)\}$
    end
return $v$
```

(a) Computing vectors.

```
$expandVectors(n, V, c, B, AF, RF, A)$
    $N = B(\text{target of } c)$
    $rf = \text{relation of } c$
    $V' = \emptyset$
    for $(m, v) \in N \times V$ such that $m \in rf(n)$ do
        $w = computeVector(m, AF, c)$
        $V' = V' \cup \{v \cup w\}$
    end
return $V'$
```

(b) Expanding vectors.

Fig. 5 Procedures to deal with vectors

The second procedure is *expandVectors*. It works on a node n, a set of vectors V that is associated with n in a given training set, a context tuple c, a binding B, and an input configuration (AF, RF, A). It first computes the set of nodes N that correspond to the target of context tuple c and, after getting the relational feature in c and initialising the result V' to the empty set, it iterates over the set of pairs (m, v) in which m denotes a node in N and v is one of the vectors in V; note that only pairs in which $m \in rf(n)$ are considered, that is, pairs in which node m is a neighbour of node n regarding relational feature rf. For every such pair, we first compute the vector w that corresponds to m using the set of attributive features AF and the context tuple c; that vector is then merged with vector v, that is, vector v is expanded with the attributive features of node m.

Example 9 Let us examine node n_{10} in the document in Fig. 1 and the initial context tuple $c = (node_0, null, null)$. Recall that Table 1a reports on the attributive features of the nodes in our running example. The vector that is associated with this node is the following: $v = \{(c, tag) \mapsto span, (c, class) \mapsto null, (c, y\text{-}pos) \mapsto 16, (c, x\text{-}pos) \mapsto 0, (c, len) \mapsto 1, (c, is\text{-}number) \mapsto false\}$. If this vector is expanded with context tuple $c' = (node_1, parent, node_0)$, then it becomes $v' = \{(c, tag) \mapsto span, (c, class) \mapsto null, (c, y\text{-}pos) \mapsto 16, (c, x\text{-}pos) \mapsto 0, (c, len) \mapsto 1, (c, is\text{-}number) \mapsto false, (c', tag) \mapsto li, (c', class) \mapsto null, (c', y\text{-}pos) \mapsto 16, (c', x\text{-}pos) \mapsto 0, (c', len) \mapsto 2, (c', is\text{-}number) \mapsto false\}$.

3 Configuring Roller

Roller has three variation points, namely: procedure FEATUREBUILDER, which computes a catalogue of features, BASERLEARNER, which learns a rule from a propositional training set, and RULESCORER, which assesses how good a rule is. There are several alternatives to implement these procedures; the decision must, obviously, be made building on an experimental study that proves that the chosen combination of alternatives is very good.

The details regarding our experimentation environment and the performance measures that we used are described in "Appendices 1 and 2", respectively. In the following subsections, we first present a method that helped us make a decision regarding which the best combination of alternatives was; then, we report on the feature builder, the base learner, and the rule scorers that we examined; finally, we report on the results of the experimental analysis that we carried out to find the best combination of alternatives.

3.1 A ranking method

Since we have to compare many alternatives regarding the variation points in Roller, we need a method to rank them building on a number of performance measures that we denote as M. Without loss of generality, we can assume that M can be partitioned as $M = A \uplus B$, where A denotes a set of performance measures whose value must be maximised and B denotes a set of performance measures whose value must be minimised.

For the method to work well, we first need to normalise the values of the measures so that they all range within the same interval and the interpretation of the lower and the upper bounds is homogeneous. Assume that we are dealing with a performance measure m, that we have gathered a set of values W regarding it, that a denotes the minimum value in set W, and that b denotes the maximum value. If $m \in A$, then we define its set of normalised values as $W' = \{w' \mid \exists w \cdot w \in W \wedge w' = (w - a) \div (b - a)\}$; if $m \in B$, then we define its set of normalised values as $W' = \{w' \mid \exists w \cdot w \in W \wedge w' = 1.00 - (w - a) \div (b - a)\}$. ($a \div b$ equals a/b if $b \neq 0.00$; otherwise, it equals 1.00.) Note that the values in W' range in interval [0.00 .. 1.00], so that the closer a value to the lower bound the worse and the closer to the upper bound the better.

We now have to transform the normalised values of the measures into a rank. We assume that the experimenter provides a map β that assigns a relative weight in range [0.00 .. 1.00] to every measure in M; obviously, the weights must sum up to 1.00 so that they are consistent. The idea behind weighting the performance measures is that the researcher can provide a hint regarding the measures on which he or she is most interested. Note that we need to perform several experiments to rank a number of alternatives, which means that we actually compute a set of values for every measure. To work with that set, we need to compute their mean value, but we also have to take into account their deviation since a high deviation means that the mean value of a variable is not actually representative, that is, it is not good to estimate a rank. To deal with this problem, we have devised the following formula:

$$K^p = \sum_{m' \in M'} \beta(m)\, K^p_{m'}$$

$$K^p_{m'} = \frac{\mathrm{mdr}^p_{m'}}{\mathrm{mdr}^{max}_{m'}}$$

where p denotes an alternative to be compared, M' denotes a set of new measures that are in one-to-one correspondence with the measures in M, but are normalised according to the previous procedure, and $\mathrm{mdr}^p_{m'}$ denotes the mean-to-deviation ratio of alternative p with regard to normalised performance measure m' and $\mathrm{mdr}^{\max}_{m'}$ denotes the maximum mean-to-deviation ratio of performance measure m' across all of the alternatives to compare. This ratio is defined as follows:

$$\mathrm{mdr}^p_m = \begin{cases} \frac{\mu^p_m}{\sigma^p_m}\, \mu^p_m & \text{if } \sigma^p_m \neq 0.00 \\ \mu^p_m & \text{otherwise} \end{cases}$$

where m denotes an arbitrary performance measure, μ^p_m denotes its mean value regarding alternative p, and σ^p_m its standard deviation regarding alternative p. Note that this ratio maps every measure onto a value that weights its mean value with the inverse of the coefficient of variation. Intuitively, the smallest the coefficient of variation, the more representative the mean value; in other words, the smallest the coefficient of variation of a measure, the more can that measure contribute to the rank of an alternative.

Table 4 Partial catalogue of user-defined features

(a) *Some attributive features*

beginsWithNumber
beginsWithParenthesis
beginsWithPunctuation
countOfAlphaNum
countOfBlanks
countOfCapitals
countOfDigits
countOfIntegers
countOfLetters
countOfLowercaseBigrams
countOfLowercaseTokens
countOfLowercaseTrigrams
countOfTokens
countOfTrigrams
countOfUppercaseBigrams
countOfUppercaseTokens
countOfUppercaseTokens
countOfUppercaseTrigrams
endsWithNumber
endsWithParenthesis
endsWithPunctuation
firstBigram
firstToken
hasBlanks
hasBracketedAlphaNum
hasBracketedNumber
hasCurrencySymbol
hasQuestionMark
isAlphaNum
isBlank
isCapitalised
isCurrency
isDate
isEmail
isISBN
isLowerCase
isNumber
isPhone
isUppercase
isURL
isYear
lastBigram
lastToken

(b) *Some relational features*

ancestor
children
firstSibling
lastSibling
leftSibling
parent
rightSibling

Summing up, K^p provides a rank for alternative p in which both the mean value of the performance measures and their deviation are taken into account. We think that this is a good approach since it blurs the contribution of measures that are not stable enough, but emphasises the others. Please, note that it was not our purpose to devise a general-purpose ranking method, but an ad hoc method that has proven to guide our search for a good alternative very well.

Before concluding, we would like to mention that our experimental analysis has revealed that some alternatives fail when they are applied to some datasets. Sometimes, the reason is that they consume too much memory; sometimes, they cannot learn a rule in a reasonable time (we set a deadline of 15 CPU minutes); sometimes, the dataset has some characteristics that make it impossible to execute the base learner on it. That means that we also need to compute a failure ratio for every alternative under consideration, which is defined as follows:

$$FR = \frac{F}{D},$$

where F denotes the number of datasets on which an alternative did not work, and D denotes the number of datasets on which the alternative was run. Intuitively, the closer to 0.00 the better and the closer to 1.00 the worse. We obviously are not willing to accept an alternative whose failure ratio is different from 0.00, since that means that it is not generally applicable.

3.2 Our feature builder

Our feature builder computes the standard HTML features and the standard rendering features of the input documents, as they are defined in the corresponding W3C recommendations [31, 63]. Additionally, it computes some user-defined features; Table 4 shows only the user-defined features that have proven to be useful in our experiments.

The attributive features can be classified according to the prefixes of their names into the following groups: (a) prefix $beginsWith$ identifies some features that check whether the text of a node begins with a token that belongs to a given lexical class, e.g., a number or a punctuation symbol; (b) prefix $countOf$ identifies some features that count the number of tokens in the text of a node that fulfil a given property, e.g., the count of alphanumeric tokens or the count of lowercase tokens; (c) prefix $endsWith$ denotes features that check if the text in a node ends with a token that belongs to a given lexical class, e.g., a number or a punctuation symbol; (d) prefix $first$ denotes features that return a prefix of the text in a node, e.g., the first token or the first two tokens (which are commonly referred to as bigrams); (e) prefix has identifies features that check if there is a subsequence of tokens in the text of a node that fulfils a given property, e.g., there is a bracketed number or a question mark; (f) prefix is denotes a feature that checks if the text in a node matches a given pattern, e.g., whether it is capitalised or a phone number; (g) finally, prefix $last$ denotes features that return a suffix of the text in a node, e.g., the last token or the last bigram.

The catalogue of relational features provides common features to navigate from a node to its neighbours in a DOM tree, namely: ancestor, children, first sibling, last sibling, left sibling, right sibling, and parent.

Recall that Roller is not bound with a particular choice of features. It is open to work with the features that a user thinks are the most appropriate for a given problem. The previous features provide just a catalogue that has proven to work very well in our experiments.

3.3 Our base learner and rule scorer

Regarding the base learner, we have explored Conjunctive Rule, Decision Table, JRip, NNge, PART, and Ridor [22]. They are available in Weka and can deal with multi-class problems and both numeric and categoric attributive features. A problem with them is that they do not work well with training sets that are unbalanced, which is the case in our context. The reason is that input documents are composed of hundreds of nodes, most of which are negative examples, cf. Table 8 in "Appendix 1". Thus the base learner must balance the training sets on which it works. We have explored several alternatives in the literature [4,30], and our conclusion was that the one that best performs consists in computing the number of examples of the majority slot and then replicating as many examples of the other slots as needed to assemble a training set that has approximately the same number of examples for every slot.

Regarding the rule scorer, Information Content is the most common in practice [51]. It has proven to guide the search process very well when dealing with classical inductive logic programming problems. It relies exclusively on the number of true positives and false positives that a rule produces when it is evaluated. We wished to explore some rule scorers that also take into account the number of true negatives and false negatives. We have surveyed the literature and we have found several alternatives [27], namely: Collective Strength, Confidence, Jaccard, Kappa, Laplace, Leverage, Odds Ratio, Phi Coefficient, Satisfaction, Support, and Yule's Q.

The cartesian product of base learners and rule scorers resulted in 72 variations of Roller. Table 5 summarises the results that we obtained when we run each variation on our datasets, including the mean and standard deviations of precision (P), recall (R), the F_1 score (F_1), learning time (LT), and extraction time (ET), as well as our rank (K) and the failure ratio (FR).

In our experimentation, we set the relative weights of the performance measures to $\beta(F_1) = 0.70$, $\beta(LT) = 0.10$, and $\beta(ET) = 0.20$. In other words, we think that a good

Table 5 Experimental results regarding several variants of Roller

	Information content							Collective strength							Confidence						
	P	R	F1	LT	ET	FR	K	P	R	F1	LT	ET	FR	K	P	R	F1	LT	ET	FR	K
Conj. rule																					
Mean	0.12	0.32	0.18	15.79	0.71	0.00	0.29	0.11	0.32	0.17	16.58	0.76	0.02	0.29	0.13	0.33	0.18	16.96	0.76	0.00	0.28
Stdev.	0.13	0.13	0.13	14.66	0.78			0.06	0.09	0.07	16.26	0.90			0.13	0.13	0.14	15.84	0.85		
mdr	0.12	0.80	0.25	17.02	0.64			0.23	1.19	0.38	16.91	0.63			0.13	0.86	0.24	18.16	0.67		
Decision table																					
Mean	0.72	0.72	0.72	410.16	1.15	0.00	0.29	0.84	0.77	0.76	246.00	0.14	0.77	0.40	0.93	0.88	0.88	81.75	0.14	0.00	0.59
Stdev.	0.39	0.33	0.36	1773.50	4.14			0.13	0.14	0.15	518.41	0.18			0.08	0.12	0.13	268.89	0.12		
mdr	1.32	1.59	1.45	94.86	0.32			5.35	4.39	3.92	116.73	0.11			11.26	6.44	6.14	24.85	0.17		
JRip																					
Mean	0.87	0.88	0.88	54.51	2.26	0.00	0.51	0.89	0.86	0.84	55.58	0.07	0.81	0.70	0.96	0.94	0.94	27.86	0.06	0.00	0.92
Stdev.	0.19	0.16	0.17	275.35	13.49			0.07	0.07	0.09	78.42	0.07			0.05	0.07	0.07	46.01	0.05		
mdr	3.93	4.95	4.43	10.79	0.38			11.03	9.79	8.34	39.39	0.08			18.82	13.04	12.31	16.87	0.07		
NNge																					
Mean	0.79	0.63	0.70	13.95	4.36	0.00	0.37	0.00	0.00	0.00	0.00	0.00	1.00	0.30	0.80	0.63	0.61	14.77	4.53	0.00	0.32
Stdev.	0.19	0.18	0.19	15.78	6.23			0.00	0.00	0.00	0.00	0.00			0.17	0.18	0.20	16.66	6.40		
mdr	3.30	2.14	2.62	12.34	3.05			0.00	0.00	0.00	0.00	0.00			3.65	2.20	1.82	13.10	3.20		
PART																					
Mean	0.91	0.90	0.91	39.67	1.25	0.00	0.60	0.94	0.92	0.91	37.07	0.09	0.29	0.78	0.95	0.94	0.93	18.79	0.07	0.00	0.86
Stdev.	0.14	0.13	0.14	158.24	3.68			0.07	0.08	0.09	50.86	0.09			0.05	0.08	0.08	28.62	0.07		
mdr	5.69	6.37	6.03	9.94	0.42			13.55	10.70	9.70	27.02	0.09			17.53	11.63	10.93	12.34	0.08		
Ridor																					
Mean	0.76	0.80	0.78	29.45	0.69	0.00	0.39	0.91	37.07	0.09	0.94	0.92	0.86	0.09	0.97	0.95	0.95	42.32	0.07	0.16	0.97
Stdev.	0.29	0.24	0.26	40.09	1.11			0.09	50.86	0.09	0.07	0.08			0.05	0.07	0.07	63.17	0.09		
mdr	2.00	2.63	2.30	21.63	0.43			9.70	27.02	0.09	13.55	10.70			20.24	13.89	13.50	28.35	0.06		

Table 5 continued

	Jaccard							Kappa							Laplace						
	P	R	F1	LT	ET	FR	K	P	R	F1	LT	ET	FR	K	P	R	F1	LT	ET	FR	K
Conj. rule																					
Mean	0.13	0.33	0.18	18.09	0.79	0.00	0.28	0.13	0.33	0.18	16.41	0.75	0.00	0.28	0.13	0.33	0.18	15.75	0.74	0.00	0.29
Stdev.	0.13	0.13	0.14	18.38	0.87			0.13	0.13	0.14	14.61	0.83			0.13	0.13	0.14	14.11	0.84		
mdr	0.13	0.86	0.24	17.82	0.71			0.13	0.86	0.24	18.44	0.68			0.13	0.86	0.24	17.58	0.64		
Decision table																					
Mean	0.93	0.88	0.88	87.53	0.14	0.00	0.59	0.93	0.88	0.88	90.83	0.14	0.00	0.59	0.93	0.88	0.88	86.43	0.15	0.00	0.59
Stdev.	0.08	0.12	0.13	295.85	0.12			0.08	0.12	0.13	305.40	0.12			0.08	0.12	0.13	292.86	0.13		
mdr	11.26	6.44	6.14	25.90	0.17			11.26	6.44	6.14	27.01	0.16			11.26	6.44	6.14	25.51	0.17		
JRip																					
Mean	0.96	0.94	0.94	30.55	0.06	0.00	0.92	0.96	0.94	0.94	25.88	0.05	0.00	0.92	0.96	0.94	0.94	26.02	0.06	0.00	0.92
Stdev.	0.05	0.07	0.07	51.19	0.04			0.05	0.07	0.07	43.55	0.05			0.05	0.07	0.07	44.15	0.05		
mdr	18.82	13.04	12.31	18.23	0.08			18.82	13.04	12.31	15.38	0.06			18.82	13.04	12.31	15.33	0.07		
NNge																					
Mean	0.80	0.63	0.61	14.97	4.21	0.00	0.33	0.80	0.63	0.61	17.99	4.70	0.00	0.32	0.80	0.63	0.61	15.32	4.64	0.00	0.32
Stdev.	0.17	0.18	0.20	16.49	6.08			0.17	0.18	0.20	21.46	6.70			0.17	0.18	0.20	16.98	6.64		
mdr	3.65	2.20	1.82	13.58	2.92			3.65	2.20	1.82	15.08	3.30			3.65	2.20	1.82	13.83	3.24		
PART																					
Mean	0.95	0.94	0.93	17.93	0.08	0.00	0.86	0.95	0.94	0.93	17.43	0.07	0.00	0.86	0.95	0.94	0.93	17.70	0.07	0.00	0.86
Stdev.	0.05	0.08	0.08	26.56	0.09			0.05	0.08	0.08	24.68	0.07			0.05	0.08	0.08	26.02	0.07		
mdr	17.53	11.63	10.93	12.10	0.07			17.53	11.63	10.93	12.31	0.08			17.53	11.63	10.93	12.05	0.07		
Ridor																					
Mean	0.97	0.95	0.95	43.17	0.08	0.16	0.97	0.97	0.95	0.95	37.82	0.07	0.16	0.98	0.97	0.95	0.95	36.29	0.07	0.16	0.98
Stdev.	0.05	0.07	0.07	64.73	0.09			0.05	0.07	0.07	57.17	0.08			0.05	0.07	0.07	52.13	0.09		
mdr	20.24	13.89	13.50	28.79	0.06			20.24	13.89	13.50	25.02	0.05			20.24	13.89	13.50	25.26	0.05		

Table 5 continued

	Leverage							Odds ratio							Phi coefficient						
	P	R	F1	LT	ET	FR	K	P	R	F1	LT	ET	FR	K	P	R	F1	LT	ET	FR	K
Conj. rule																					
Mean	0.12	0.32	0.17	24.90	0.99	0.00	0.28	0.13	0.33	0.18	14.57	0.70	0.00	0.29	0.12	0.32	0.17	21.59	0.92	0.00	0.29
Stdev.	0.14	0.13	0.14	39.40	1.28			0.13	0.13	0.14	13.71	0.80			0.14	0.13	0.14	34.02	1.18		
mdr	0.11	0.77	0.21	15.74	0.76			0.13	0.86	0.24	15.48	0.61			0.11	0.79	0.21	13.70	0.71		
Decision table																					
Mean	0.87	0.83	0.82	316.82	0.71	0.00	0.39	0.85	0.82	0.81	293.71	0.65	0.00	0.38	0.89	0.85	0.84	287.91	0.57	0.00	0.41
Stdev.	0.25	0.22	0.25	1770.51	3.75			0.28	0.24	0.27	1575.30	3.01			0.23	0.21	0.23	1673.95	2.96		
mdr	2.98	3.12	2.68	56.69	0.14			2.63	2.87	2.42	54.76	0.14			3.48	3.51	3.07	49.52	0.11		
JRip																					
Mean	0.92	0.91	0.90	132.72	1.53	0.00	0.55	0.89	0.89	0.88	113.77	1.59	0.00	0.48	0.93	0.92	0.92	97.69	1.16	0.00	0.64
Stdev.	0.13	0.12	0.14	361.33	4.96			0.18	0.16	0.18	391.26	5.06			0.10	0.10	0.12	321.07	4.30		
mdr	6.42	6.67	5.80	48.75	0.47			4.37	5.04	4.24	33.08	0.50			8.40	8.31	7.25	29.72	0.31		
NNge																					
Mean	0.80	0.63	0.61	14.75	4.49	0.00	0.32	0.80	0.63	0.61	14.32	4.30	0.00	0.33	0.80	0.63	0.61	13.54	4.24	0.00	0.33
Stdev.	0.17	0.18	0.20	16.39	6.40			0.17	0.18	0.20	16.11	6.13			0.17	0.18	0.20	15.09	6.05		
mdr	3.65	2.20	1.82	13.28	3.14			3.65	2.20	1.82	12.73	3.01			3.65	2.20	1.82	12.15	2.98		
PART																					
Mean	0.95	0.94	0.94	35.10	0.32	0.00	0.84	0.93	0.92	0.91	63.10	1.46	0.00	0.61	0.95	0.94	0.94	34.04	0.48	0.00	0.86
Stdev.	0.06	0.08	0.08	90.17	1.15			0.12	0.11	0.13	252.71	5.09			0.06	0.08	0.08	81.51	1.79		
mdr	14.92	11.20	10.72	13.67	0.09			6.96	7.31	6.31	15.76	0.42			14.68	11.64	11.02	14.22	0.13		
Ridor																					
Mean	0.92	0.92	0.91	232.88	1.39	0.16	0.51	0.87	0.88	0.86	139.21	1.21	0.16	0.45	0.94	0.94	0.93	123.45	0.57	0.16	0.58
Stdev.	0.15	0.13	0.16	798.71	4.94			0.20	0.17	0.20	564.92	4.35			0.13	0.12	0.14	475.31	1.86		
mdr	5.56	6.45	5.25	67.90	0.39			3.70	4.62	3.66	34.30	0.33			6.54	7.38	6.05	32.07	0.17		

Table 5 continued

	Satisfaction							Support							Yule's Q						
	P	R	F1	LT	ET	FR	K	P	R	F1	LT	ET	FR	K	P	R	F1	LT	ET	FR	K
Conj. rule																					
Mean	0.13	0.33	0.18	16.46	0.78	0.00	0.28	0.13	0.33	0.18	16.68	0.75	0.00	0.28	0.13	0.33	0.18	16.57	0.76	0.00	0.28
Stdev.	0.13	0.13	0.14	15.26	0.92			0.13	0.13	0.14	15.48	0.85			0.13	0.13	0.14	16.02	0.85		
mdr	0.13	0.86	0.24	17.75	0.66			0.13	0.86	0.24	17.98	0.67			0.13	0.86	0.24	17.14	0.69		
Decision table																					
Mean	0.93	0.88	0.88	88.62	0.15	0.00	0.59	0.93	0.88	0.88	85.53	0.15	0.00	0.59	0.86	0.83	0.81	128.18	0.27	0.00	0.40
Stdev.	0.08	0.12	0.13	304.88	0.14			0.08	0.12	0.13	287.70	0.12			0.25	0.22	0.25	390.45	0.60		
mdr	11.26	6.44	6.14	25.76	0.16			11.26	6.44	6.14	25.42	0.18			2.95	3.15	2.66	42.08	0.12		
JRip																					
Mean	0.96	0.94	0.94	25.19	0.06	0.00	0.92	0.96	0.94	0.94	25.74	0.06	0.00	0.92	0.92	0.91	0.90	122.52	1.20	0.00	0.54
Stdev.	0.05	0.07	0.07	39.78	0.05			0.05	0.07	0.07	41.09	0.05			0.15	0.13	0.15	422.22	4.52		
mdr	18.82	13.04	12.31	15.95	0.07			18.82	13.04	12.31	16.13	0.07			5.77	6.35	5.42	35.55	0.32		
NNge																					
Mean	0.80	0.63	0.61	14.70	4.56	0.00	0.32	0.80	0.63	0.61	15.10	4.63	0.00	0.32	0.80	0.63	0.61	14.90	4.61	0.00	0.32
Stdev.	0.17	0.18	0.20	16.32	6.53			0.17	0.18	0.20	16.61	6.59			0.17	0.18	0.20	16.29	6.56		
mdr	3.65	2.20	1.82	13.24	3.19			3.65	2.20	1.82	13.72	3.26			3.65	2.20	1.82	13.63	3.24		
PART																					
Mean	0.95	0.94	0.93	18.42	0.08	0.00	0.85	0.95	0.94	0.93	19.88	0.08	0.00	0.85	0.93	0.92	0.91	65.00	1.40	0.00	0.61
Stdev.	0.05	0.08	0.08	24.85	0.07			0.05	0.08	0.08	31.05	0.08			0.12	0.11	0.13	245.23	4.59		
mdr	17.53	11.63	10.93	13.65	0.08			17.53	11.63	10.93	12.73	0.09			7.03	7.38	6.38	17.23	0.43		
Ridor																					
Mean	0.97	0.95	0.95	35.30	0.06	0.16	0.98	0.97	0.95	0.95	36.33	0.07	0.16	0.98	0.88	0.90	0.88	169.33	1.21	0.16	0.47
Stdev.	0.05	0.07	0.07	53.38	0.09			0.05	0.07	0.07	54.39	0.09			0.19	0.16	0.19	703.64	4.53		
mdr	20.24	13.89	13.50	23.34	0.05			20.24	13.89	13.50	24.26	0.05			4.04	5.01	4.00	40.75	0.32		

proposal must be able to learn extraction rules that are very effective, that is, that achieve a high precision and recall, and, consequently, a high F_1 score. Note that learning a rule is a process that is executed every now and then, when a new site needs to be analysed or when a rule breaks because the corresponding site has undergone a change to its layout; since our experimental analysis confirmed that Roller is quite effective and can learn in a matter of seconds, we did not think that the learning time could make a big difference between two alternatives. Contrarily, once a rule is learnt, it must be executed as quickly as possible in a production environment, so the extraction time is much more important than the learning time.

Note that the best variations achieve $K = 0.98$ and $K = 0.97$; they all rely on Ridor as the base learner and Jaccard, Laplace, Satisfaction, or Support as the rule scorers; unfortunately, all of them have a failure ratio of 0.16, which means that they cannot deal with some datasets. The problem is that Ridor is a learner that uses a technique called Reduced Error Pruning to prune the resulting rules; unfortunately, there are a number of datasets that do not provide enough data for this technique to work, which means that it cannot be applied to relatively small documents. As a conclusion, we have to resign to use Ridor, even though it works well with sufficiently large documents.

Thus, the best variations seem to be those that achieve $K = 0.92$ with a 0.00 failure ratio. They all correspond to using JRip as the base learner and Confidence, Jaccard, Kappa, Laplace, Satisfaction, and Support as rule scorers. Since there are multiples ties, we decided to select JRip and Kappa because this is the variation that achieves the minimum extraction time.

4 Experimental analysis

In this section, we first report on the results of our experimental analysis regarding effectiveness and then regarding efficiency. The details regarding our experimentation environment, including the proposals with which we have compared ours, and the performance measures that we have used are described in "Appendices 1 and 2", respectively. In every case, we have conducted a statistical analysis to make sure that the differences in rank that our experiments have found are statistically significant at the standard significance level $\alpha = 0.05$.

Following the results by Demšar [16] and García and Herrera [25], we have used Iman–Davenport's test to find out whether there are statistically significant differences in the empirical ranks and then Hommel's test to compare the best-ranked technique to the others. Note that we have to resort to nonparametric tests because the distribution of the performance measures is not normal and they are not homoscedastic [54]. Regarding the normality of Roller's precision, for instance, Kolmogorov–Smirnov's test returns $D = 0.78$ with a p value less than 2.20×10^{-16}, Shapiro–Wilk's test returns $W = 0.79$ with p value 2.39×10^{-07}, and Arlinton–Darling's test returns $AD = 49.85$ with p value 1.11×10^{-05}; regarding homoscedasticity, the comparison between Roller's and SoftMealy's precision, for instance, returns $F = 49.64$ (with one degree of freedom) and p value 1.94×10^{-10} using Levene's test, $K = 69.66$ (with one degree of freedom) and p value less than 2.20×10^{-16} using Bartlett's test, and $F = 0.08$ with p value less than 2.20×10^{-16} using the F test. Note that the previous results provide a strong indication that the data does not behave normally and is not homoscedastic, which precludes using parametric tests like the well-known t test or ANOVA tests.

4.1 Effectiveness analysis

Table 6 reports on the raw effectiveness data that we got from our experimentation. For each proposal, we report on its precision (P), recall (R), and F_1 score (F_1) regarding our

Table 6 Effectiveness results regarding precision (P), recall (R), and the F_1 score

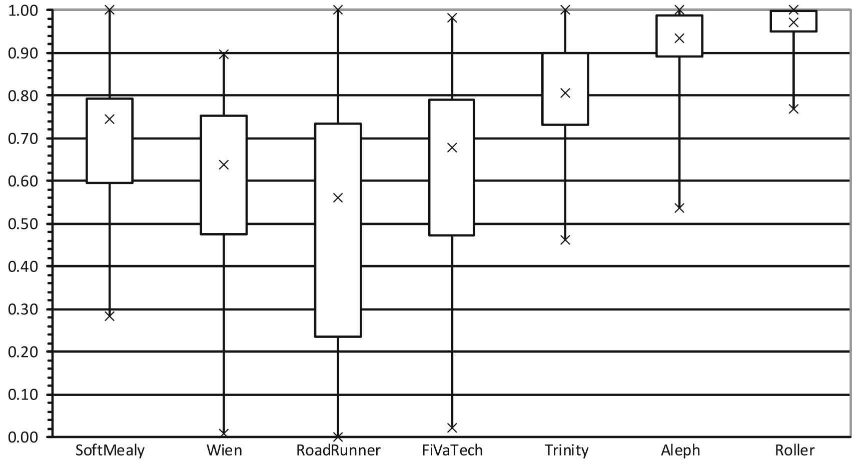
Dataset	SoftMealy			Wien			RoadRunner			FivaTech			Trinity			Aleph			Roller		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Mean	0.71	0.61	0.63	0.59	0.68	0.58	0.51	0.68	0.52	0.62	0.81	0.67	0.80	0.90	0.84	0.91	0.90	0.90	0.96	0.95	0.94
Std. deviation	0.17	0.33	0.31	0.22	0.33	0.27	0.29	0.38	0.32	0.25	0.20	0.24	0.12	0.10	0.09	0.10	0.11	0.10	0.05	0.07	0.07
Abe books	0.63	1.00	0.77	0.50	0.09	0.15	0.60	0.53	0.56	0.75	1.00	0.86	0.90	0.96	0.93	0.94	0.92	0.93	1.00	1.00	1.00
Awesome books	0.85	0.37	0.52	0.77	0.20	0.31	0.70	0.43	0.54	0.80	0.96	0.87	0.91	0.86	0.88	0.99	0.91	0.95	1.00	1.00	1.00
Better world books	0.78	0.96	0.86	0.37	0.34	0.36	–	–	–	0.91	0.93	0.92	0.71	0.70	0.70	0.95	0.95	0.95	1.00	1.00	1.00
Many books	0.70	1.00	0.83	0.01	0.22	0.02	0.88	1.00	0.94	0.54	1.00	0.70	0.77	0.90	0.83	0.99	0.99	0.99	0.98	0.97	0.98
Waterstones	1.00	0.92	0.96	0.68	0.67	0.68	0.71	0.77	0.74	0.73	0.86	0.79	0.87	0.89	0.88	0.96	1.00	0.98	1.00	1.00	1.00
IMDB	0.69	0.78	0.74	0.19	0.30	0.23	0.20	1.00	0.33	0.68	0.73	0.70	0.80	0.81	0.80	0.93	0.93	0.93	0.95	0.94	0.94
Disney movies	0.93	0.92	0.92	0.60	1.00	0.75	0.41	1.00	0.58	0.68	0.58	0.62	0.78	0.76	0.77	0.97	0.96	0.96	0.99	0.99	0.99
Albania movies	1.00	0.37	0.54	0.72	1.00	0.83	0.48	0.77	0.59	0.75	0.73	0.74	0.81	0.76	0.78	0.92	0.94	0.93	1.00	1.00	1.00
All movies	0.78	0.20	0.32	0.02	0.04	0.03	0.23	1.00	0.38	0.62	0.66	0.64	0.92	0.81	0.86	0.91	0.80	0.85	0.92	0.89	0.89
Soul films	0.90	1.00	0.95	0.72	1.00	0.83	0.49	0.45	0.47	0.41	0.96	0.58	0.86	0.91	0.88	0.95	0.92	0.94	0.97	0.96	0.96
Auto trader	0.75	1.00	0.86	0.64	0.00	0.00	–	–	–	–	–	–	0.81	0.81	0.81	0.90	0.91	0.91	0.96	0.94	0.95
Car max	0.78	0.80	0.79	0.76	0.78	0.77	0.76	0.95	0.84	0.32	0.82	0.46	0.83	0.80	0.81	0.80	0.81	0.81	0.93	0.92	0.92
Car zone	0.67	0.02	0.04	0.73	0.77	0.75	0.56	1.00	0.72	0.83	0.99	0.90	0.91	0.91	0.91	0.89	0.83	0.86	0.96	0.94	0.94
Classic cars for sale	0.86	0.89	0.88	0.10	1.00	0.19	0.36	0.46	0.40	–	–	–	0.61	0.84	0.71	0.92	0.83	0.87	0.96	0.95	0.95
Internet autoguide	0.46	0.43	0.44	0.17	0.02	0.04	0.90	0.99	0.94	0.85	0.93	0.89	0.76	0.99	0.86	0.83	0.88	0.85	0.99	0.99	0.99
Linked in	0.78	0.53	0.63	0.56	0.20	0.29	0.38	0.49	0.43	0.78	0.87	0.83	0.89	0.86	0.87	0.92	1.00	0.96	0.94	0.90	0.91
All conferences	0.96	0.17	0.28	0.78	0.35	0.48	0.61	1.00	0.76	0.71	0.80	0.75	0.97	0.96	0.96	0.99	0.95	0.97	0.77	0.77	0.74

Table 6 continued

Dataset	SoftMealy			Wien			RoadRunner			FivaTech			Trinity			Aleph			Roller		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Mbendi	1.00	0.55	0.71	0.66	0.40	0.50	0.62	0.82	0.71	0.61	0.99	0.76	0.81	0.97	0.88	0.96	0.96	0.96	0.98	0.98	0.98
Net lib	0.87	0.44	0.59	0.40	0.40	0.40	0.00	0.00	0.00	0.39	0.50	0.44	0.96	0.98	0.97	0.92	0.93	0.92	0.94	0.98	0.96
RD learning	0.34	0.33	0.33	0.35	1.00	0.52	0.73	1.00	0.85	0.86	0.74	0.80	0.75	0.94	0.83	0.98	1.00	0.99	1.00	1.00	1.00
Web MD	0.76	0.40	0.52	0.59	0.57	0.58	0.22	0.04	0.07	0.54	0.95	0.69	0.96	0.94	0.95	0.93	0.94	0.93	1.00	1.00	1.00
Ame. Medical Assoc.	0.53	0.31	0.39	0.55	0.56	0.55	–	–	–	0.11	0.19	0.14	0.73	0.93	0.82	0.80	0.79	0.80	1.00	1.00	1.00
Dentists	0.56	0.60	0.58	0.88	0.99	0.93	0.84	1.00	0.92	0.40	0.95	0.56	0.86	0.99	0.92	1.00	0.89	0.94	0.95	0.93	0.93
Dr. score	0.73	0.80	0.77	0.71	0.78	0.74	0.65	0.98	0.78	0.67	0.95	0.79	0.72	0.95	0.82	0.61	0.64	0.62	0.98	0.98	0.98
Steady health	0.56	0.19	0.28	0.62	0.66	0.64	0.81	0.99	0.89	0.75	1.00	0.86	0.79	0.94	0.86	1.00	0.89	0.94	0.90	0.79	0.77
Insight into diversity	0.45	0.45	0.45	0.76	0.97	0.85	0.42	0.48	0.45	0.98	0.67	0.80	0.63	1.00	0.77	0.98	0.92	0.95	0.96	0.95	0.95
4 jobs	0.42	0.15	0.22	0.86	0.85	0.85	0.10	1.00	0.18	0.87	0.60	0.71	0.82	0.90	0.86	0.82	0.77	0.80	0.95	0.94	0.94
6 figure jobs	0.53	1.00	0.69	0.21	1.00	0.35	0.23	1.00	0.38	0.93	0.92	0.93	0.70	0.95	0.81	0.86	0.88	0.87	1.00	1.00	1.00
Career builder	0.70	0.09	0.16	0.48	1.00	0.65	0.02	0.07	0.03	0.59	1.00	0.74	0.85	0.92	0.88	1.00	0.92	0.96	0.86	0.77	0.77
Job of mine	0.46	0.05	0.10	0.34	0.42	0.38	0.61	0.66	0.63	0.53	0.52	0.53	0.67	0.99	0.80	0.89	0.99	0.94	0.89	0.77	0.76
Yahoo!	1.00	1.00	1.00	0.83	0.83	0.83	–	–	–	0.77	0.97	0.86	1.00	1.00	1.00	–	–	–	0.95	0.93	0.94
Haart	0.79	0.90	0.84	0.67	0.68	0.68	0.56	0.78	0.65	0.67	0.95	0.78	0.88	0.95	0.91	0.90	1.00	0.95	0.95	0.93	0.94
Homes	0.77	0.72	0.74	0.82	0.88	0.85	0.99	0.95	0.97	–	–	–	0.96	0.98	0.97	0.83	0.81	0.82	0.84	0.81	0.80
Remax	0.59	0.79	0.68	0.80	1.00	0.89	0.26	0.05	0.08	0.70	0.71	0.71	0.47	0.95	0.63	0.99	0.98	0.98	0.99	0.99	0.99
Trulia	0.78	0.85	0.81	0.76	0.87	0.81	0.21	0.04	0.06	–	–	–	0.46	0.99	0.63	0.93	0.93	0.93	0.96	0.95	0.95
Player profiles	0.66	0.06	0.11	0.90	0.99	0.94	–	–	–	0.08	0.93	0.14	0.81	1.00	0.89	0.97	0.95	0.96	0.93	0.88	0.88
UEFA	0.75	1.00	0.86	0.83	0.94	0.88	0.86	0.91	0.88	–	–	–	0.97	0.92	0.94	1.00	1.00	1.00	0.91	0.86	0.84
ATP world tour	0.78	1.00	0.88	0.48	0.67	0.56	0.72	0.89	0.80	0.91	1.00	0.95	0.79	0.90	0.84	0.93	0.97	0.95	0.99	0.99	0.99

Table 6 continued

Dataset	SoftMealy			Wien			RoadRunner			FivaTech			Trinity			Aleph			Roller		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
NFL	0.56	1.00	0.71	0.62	1.00	0.77	0.83	0.92	0.87	0.40	0.78	0.53	0.72	1.00	0.84	0.76	0.65	0.70	0.98	0.98	0.98
Soccer base	0.74	0.87	0.80	0.64	1.00	0.78	0.66	0.95	0.77	–	–	–	0.96	0.97	0.97	0.89	0.90	0.89	0.90	0.90	0.90
Amazon cars	0.76	0.91	0.83	0.72	0.95	0.82	1.00	0.10	0.18	0.37	0.63	0.47	0.63	0.65	0.64	0.97	1.00	0.99	1.00	1.00	1.00
UEFA players	0.80	0.87	0.83	0.73	0.48	0.58	0.81	0.96	0.88	0.65	1.00	0.79	0.74	0.83	0.78	1.00	0.99	0.99	1.00	1.00	1.00
Amazon pop artists	0.88	0.68	0.77	0.74	1.00	0.85	0.23	0.07	0.10	0.94	1.00	0.97	0.86	1.00	0.92	–	–	–	0.98	0.98	0.98
UEFA teams	0.80	0.84	0.82	0.38	0.75	0.51	0.86	1.00	0.93	0.87	0.91	0.89	0.91	0.92	0.91	0.54	0.50	0.52	0.96	0.95	0.95
Aus open players	0.40	0.22	0.29	0.47	0.25	0.33	0.61	1.00	0.76	0.04	0.77	0.08	0.70	0.94	0.81	1.00	0.93	0.96	1.00	1.00	1.00
eBay bids	0.63	0.07	0.13	0.65	0.09	0.15	0.70	0.79	0.74	0.68	0.99	0.81	0.70	0.96	0.81	0.64	0.66	0.65	0.84	0.80	0.78
Major league baseball	0.87	0.37	0.52	0.47	0.28	0.35	0.19	1.00	0.32	0.73	1.00	0.84	0.75	0.48	0.58	1.00	0.98	0.99	1.00	1.00	1.00
Netflix films	0.65	0.78	0.71	0.79	0.97	0.87	0.54	0.74	0.63	0.77	0.73	0.74	0.79	1.00	0.89	0.99	0.93	0.96	0.98	0.98	0.98
RPM find packages	0.71	0.04	0.08	0.84	0.94	0.88	0.01	0.07	0.02	0.02	0.63	0.04	0.75	1.00	0.86	1.00	0.93	0.96	1.00	1.00	1.00
Bigbook	0.79	0.75	0.77	0.58	0.91	0.70	0.29	0.03	0.05	–	–	–	0.87	0.92	0.89	0.80	0.80	0.80	0.96	0.95	0.95
IAF	0.28	0.41	0.34	0.74	1.00	0.85	0.87	0.08	0.15	0.25	0.67	0.37	0.60	1.00	0.75	0.92	0.87	0.90	1.00	1.00	1.00
Okra	0.66	1.00	0.80	0.36	0.63	0.46	0.01	0.03	0.02	0.31	0.33	0.32	0.98	0.78	0.87	0.96	0.95	0.95	0.98	0.98	0.98
LA weekly	0.44	0.56	1.30	0.63	0.79	0.70	0.06	1.00	0.11	0.62	0.47	0.54	0.77	0.88	0.82	0.99	0.83	0.91	0.97	0.96	0.96
Zagat	0.60	0.62	1.50	0.53	1.00	0.70	0.24	1.00	0.39	0.87	0.94	0.90	0.95	0.85	0.90	1.00	0.97	0.98	0.97	0.97	0.97



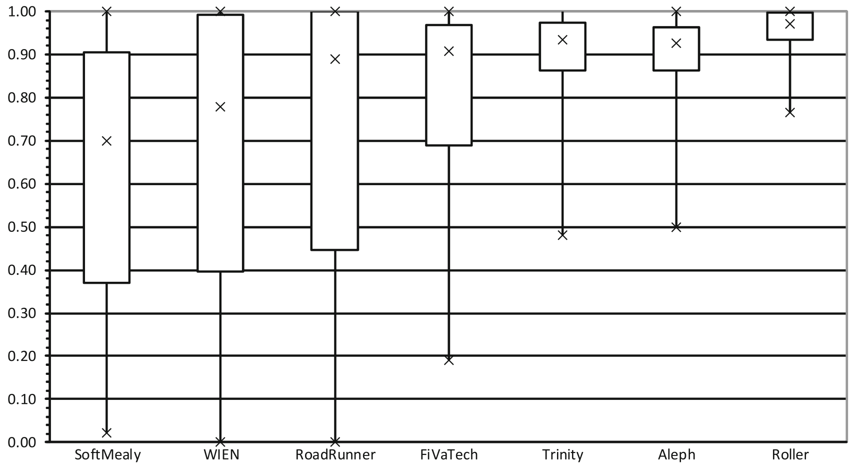
	SoftMealy	Wien	RoadRunner	FiVaTech	Trinity	Aleph	Roller
Quartile 1	0.59	0.48	0.23	0.47	0.73	0.89	0.95
Minimum	0.28	0.01	0.00	0.02	0.46	0.54	0.77
Median	0.75	0.64	0.56	0.68	0.81	0.94	0.97
Maximum	1.00	0.90	1.00	0.98	1.00	1.00	1.00
Quartile 3	0.79	0.75	0.73	0.79	0.90	0.99	1.00

Sample ranking		Iman-Davenport's	Hommel's							Statistical ranking	
Technique	Rank	p-value	adjusted p-values							Technique	Rank
Roller	1.50	8.88E-16		Aleph	Trinity	SoftMealy	FiVaTech	Wien	RoadRunner	Roller	1
Aleph	2.57		Roller	1.54E-03	1.17E-09	4.37E-09	1.71E-13	1.55E-15	9.18E-09	Aleph	2
Trinity	3.36									Trinity	2
SoftMealy	4.50									SoftMealy	2
FiVaTech	5.22									FiVaTech	2
Wien	5.26									Wien	2
RoadRunner	5.58									RoadRunner	2

Fig. 6 Summary of results regarding precision

datasets. The first two lines also provide a summary of the results in terms of mean value and the standard deviation of each measure. Since it is difficult to spot a trend in this table, we decided to summarise the data using boxplots.

Figure 6 summarises the results regarding precision. By precision, we mean the ability of a proposal to learn a rule that makes as few classification mistakes as possible; simply put, it is the ratio of true positives to the total number of true positives and false positives. Empirically, Roller seems to be the proposal that can achieve a better precision, and it is, indeed, the one that is more stable regarding this effectiveness measure; the other proposals can achieve precisions that are as high as Roller's, but their deviation with respect to the mean is larger. Iman–Davenport's test returns a p value that is nearly zero, which is a strong indication that there are differences in rank amongst the proposals that we have compared. We then have to compare Roller, which ranks the first regarding precision, and the other techniques. Hommel's test confirms that the differences in rank amongst Roller and the other techniques are statistically significant because it returns adjusted p values that are very small with regard to the significance level. In other words, our experimental data provides enough evidence to reject the hypothesis that Roller behaves similarly to the other proposals regarding precision, that is, it supports the idea that Roller can learn rules that are more precise than the rules learnt by the other proposals.

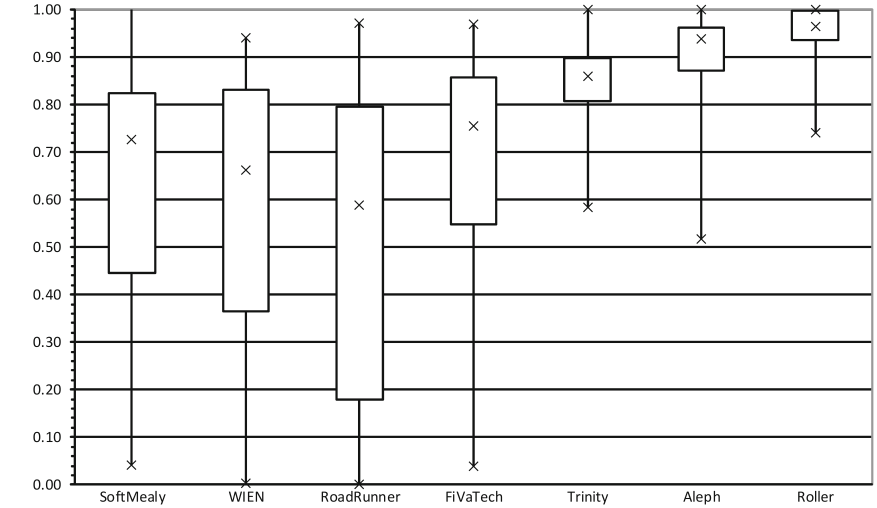


	SoftMealy	WIEN	RoadRunner	FiVaTech	Trinity	Aleph	Roller
Quartile 1	0.37	0.40	0.45	0.69	0.86	0.86	0.93
Minimum	0.02	0.00	0.00	0.19	0.48	0.50	0.77
Median	0.70	0.78	0.89	0.91	0.93	0.93	0.97
Maximum	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Quartile 3	0.90	0.99	1.00	0.97	0.97	0.96	1.00

Sample ranking		Iman-Davenport's	Hommel's							Statistical ranking	
Technique	Rank	p-value	adjusted p-values							Technique	Rank
Roller	2.69	1.30E-06		Trinity	Aleph	RoadRunner	FiVaTech	Wien	SoftMealy	Roller	1
Trinity	3.25		Roller	5.08E-02	3.74E-03	2.74E-03	5.60E-04	2.44E-05	6.25E-08	Trinity	1
Aleph	3.94									Aleph	2
RoadRunner	4.24									RoadRunner	2
FiVaTech	4.44									FiVaTech	2
Wien	4.48									Wien	2
SoftMealy	4.96									SoftMealy	2

Fig. 7 Summary of results regarding recall

Figure 7 summarises the results regarding recall. By recall, we mean the ability of a proposal to learn a rule that assigns as many nodes as possible to their correct slots; simply put, it is the ratio of true positives to the total number of true positives and false negatives. Empirically, Roller seems to be the proposal that can achieve a higher recall and it is the one that seems more stable regarding this measure because its deviation is the smallest and its interquartile range is also the smallest. Note, however, that the other techniques can achieve results that are very good, too, chiefly Trinity and Aleph. Iman–Davenport's test returns a p value that is very close to zero, which is a clear indication that there are differences in rank amongst the proposals that we have compared. Hommel's test confirms that the differences in rank between Roller, which ranks the first from an empirical point of view, Aleph, RoadRunner, FiVaTech, Wien, and SoftMealy are statistically significant at the standard significance level; note, however, that the adjusted p value that corresponds to the comparison between Roller and Trinity is not greater than the standard significance level, which means that the difference in empirical rank between these two proposals is not statistically significant. As a conclusion, the experimental data does not provide enough evidence to reject the hypothesis that Roller and Trinity behave similarly regarding recall, that is, they both rank statistically at the first position; however, it provides enough evidence to reject the hypothesis that Roller behaves similarly to Aleph, RoadRunner, FiVaTech, Wien, and SoftMealy, that is, they rank worse than Roller and Trinity.



	SoftMealy	WIEN	RoadRunner	FiVaTech	Trinity	Aleph	Roller
Quartile 1	0.44	0.36	0.18	0.55	0.81	0.87	0.94
Minimum	0.04	0.00	0.00	0.04	0.58	0.52	0.74
Median	0.73	0.66	0.59	0.75	0.86	0.94	0.96
Maximum	1.50	0.94	0.97	0.97	1.00	1.00	1.00
Quartile 3	0.83	0.83	0.80	0.86	0.90	0.96	1.00

Sample ranking		Iman-Davenport's	Hommel's							Statistical ranking	
Technique	Rank	p-value	adjusted p-values							Technique	Rank
Roller	1.70	4.44E-16		Aleph	Trinity	SoftMealy	FiVaTech	Wien	RoadRunner	Roller	1
Aleph	2.72		Roller	3.08E-03	2.57E-06	3.00E-10	2.62E-11	3.34E-14	4.91E-11	Aleph	2
Trinity	3.23									Trinity	2
SoftMealy	4.82									SoftMealy	2
FiVaTech	4.89									FiVaTech	2
Wien	5.22									Wien	2
RoadRunner	5.41									RoadRunner	2

Fig. 8 Summary of results regarding the F_1 score

Figure 8 summarises the results regarding the F_1 score. This score is the harmonic mean of precision and recall; thus, it rewards proposals that achieve both a high degree of precision and recall and penalises those that do not. Empirically, Roller seems to be the proposal that can achieve the best F_1 score, and it is, again, the most stable. Iman–Davenport's test returns a p value that is nearly zero, which strongly supports the hypothesis that there are statistically significant differences in rank. Hommel's test returns adjusted p values that are clearly smaller than the significance level in every case, which supports the hypothesis that the differences in rank between Roller and every other proposal are statistically significant, too.

Since Roller works on the tree representation of the input documents, we need to parse them and correct the errors in their HTML code. Such errors are very common, cf. Table 8 in Appendix 1. As a conclusion, it was also necessary to carry out a statistical analysis to find out if our experiments provide enough evidence to conclude that the presence of errors in the input documents has an impact on the effectiveness of our proposal. We have used Kendall's Tau test, which returned $\tau = -0.09$ with p value 0.37. Note that τ is very close to zero and that the p value is clearly greater than the standard significance level, which means that the experimental data does not provide enough evidence to reject the hypothesis that the correlation is zero. In other words, our experiments do not provide any evidence that the effectiveness of our proposal may be biased by the presence of errors in the HTML code of the input documents.

Our conclusions are that Roller outperforms the other proposals regarding effectiveness and that it is the proposal whose results are more stable. The statistical tests that we have performed have found enough evidence in our experimental data to support the hypothesis that the differences in the empirical rank amongst Roller and the other proposals are significant at the standard significance level, except for the case of recall, in which case the experimental data does not provide enough evidence to conclude that Roller and Trinity perform differently. Note, too, that proposals like RoadRunner and FiVaTech cannot deal with all of our datasets; in Table 6 such situations are indicated with a dash. The reason is that they took more than 15 CPU minutes to learn a rule or that they raised an exception; in both cases, we could not compute effectiveness measures for the corresponding datasets.

4.2 Efficiency analysis

Table 7 reports on the raw efficiency data that we got from our experimentation. For each proposal, we report on its mean learning time (LT) and its mean extraction time (ET) regarding our datasets. The first two lines also provide a summary of the results in terms of mean value and standard deviation of each measure. Since it is difficult to spot a trend in such a table, we decided to summarise the data using boxplots.

Figure 9 summarises the results regarding learning times, that is, the mean CPU time that each proposal took to learn a rule. Experimentally, it seems that Trinity is the proposal that takes less time to learn a rule; in most cases, it does not take more than a tenth of a second. It is followed by RoadRunner, SoftMealy, and Wien, whose learning times are very similar; Roller seems to rank at the fifth position, before FiVaTech and Aleph, which are the most inefficient. Iman–Davenport's test returns a p value that is very close to zero, which clearly supports the hypothesis that there are differences in rank amongst these proposals. Hommel's test also returns adjusted p values that are very small with respect to the significance level, which also reveals that the experimental data provides enough evidence to support the hypothesis that Trinity is the proposal that performs the best and that the others rank below it. Note that we do not think that this is a serious shortcoming since our learning times still lie within the range of a few seconds in most cases and we assume that learning rules is not a task that must be executed continuously in a production scenario.

Figure 10 summarises the results regarding extraction times, that is, the mean CPU time that it took to apply a rule to a dataset. Aleph, SoftMealy, and Wien seem to be the proposals that have the worst performance; RoadRunner and Trinity seem to be very similar in both mean extraction time and deviation since their inter-quartile ranges are identical. FiVaTech seems to be a little worse than RoadRunner and Trinity since its mean extraction time is larger, but note that it is a little more stable since the inter-quartile range is smaller. The timings regarding Roller are good since most rules do not take more than a tenth of a second to extract information, and its inter-quartile range is also very small with regard to the best-ranked proposals, but its mean time is slightly larger than in the case of RoadRunner and Trinity. Iman–Davenport's test returns a p value that is nearly zero, which clearly indicates that there are statistically significant differences in the empirical rank. Hommel's test returns adjusted p values that are not smaller than the standard significance level regarding the comparisons of Trinity, which is the best-ranked proposal according to the empirical ranking, Roller, RoadRunner, and FiVaTech. This means that the experimental data does not provide

Table 7 Efficiency results regarding learning time (*LT*) and extraction time (*ET*)

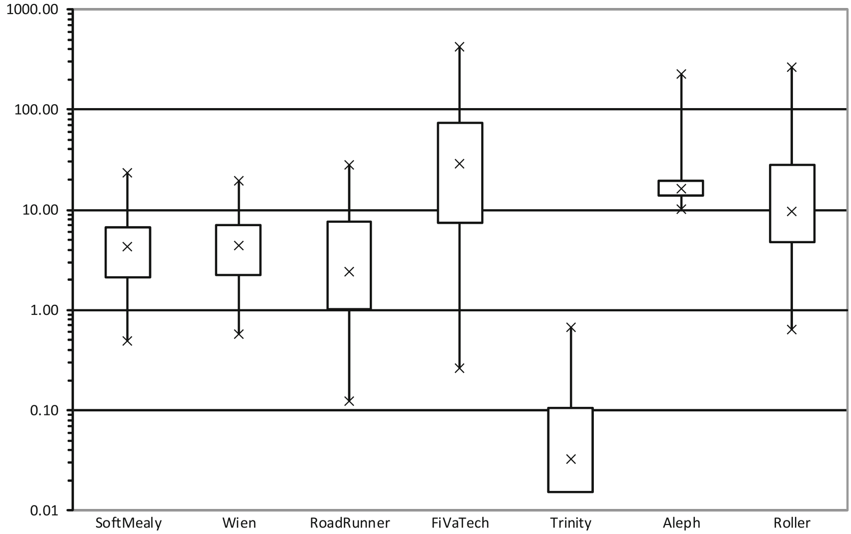
Dataset	SoftMealy		Wien		RoadRunner		FivaTech		Trinity		Aleph		Roller	
	LT	ET	LT	ET	LT	ET	LT	ET	LT	ET	LT	ET	LT	ET
Mean	5.24	35.95	5.45	8.62	5.14	0.35	66.78	0.44	0.10	0.34	27.34	52.78	24.44	0.05
Std. deviation	4.52	38.46	4.06	9.70	6.18	0.48	105.77	0.57	0.13	0.47	36.42	46.92	42.84	0.05
Abe books	9.95	18.71	10.74	10.27	3.29	0.01	9.78	0.09	0.02	0.01	15.06	32.12	8.39	0.08
Awesome books	2.53	11.49	3.25	6.28	1.55	0.01	3.67	0.10	0.02	0.01	14.92	25.81	5.18	0.03
Better world books	19.55	28.83	7.93	11.89	–	–	47.54	0.27	0.10	0.01	12.21	73.74	16.66	0.09
Many books	7.39	21.03	2.82	5.62	1.31	0.01	82.71	0.09	0.13	0.01	10.08	49.78	4.67	0.03
Waterstones	5.00	53.46	5.58	6.03	3.47	1.00	29.37	1.38	0.04	0.01	11.45	49.49	9.41	0.06
IMDB	19.89	63.25	19.47	11.47	9.92	0.01	32.13	2.32	0.24	1.00	111.09	68.47	143.09	0.13
Disney movies	4.35	31.14	2.00	2.67	1.99	0.01	121.28	0.05	0.67	0.01	74.71	25.91	41.05	0.05
Albania movies	2.08	3.45	1.36	1.16	0.91	0.01	1.88	0.01	0.01	1.00	15.19	67.41	7.08	0.05
All movies	11.54	38.87	3.23	4.11	1.90	0.01	6.23	1.00	0.27	0.01	125.19	37.21	108.73	0.09
Soul films	6.68	26.09	4.03	9.37	1.91	0.01	10.64	0.03	0.02	0.01	27.51	122.65	2.14	0.02
Auto trader	6.78	72.39	6.07	9.73	–	–	–	–	0.12	0.02	18.40	41.81	12.53	0.03
Car max	9.72	22.91	3.50	5.34	13.27	0.01	19.24	0.14	0.14	0.01	15.54	32.91	1.33	0.02
Car zone	3.43	28.88	5.76	3.32	2.72	1.00	198.79	0.41	0.02	1.00	15.23	30.35	2.59	0.00
Classic cars for sale	13.74	78.55	11.89	7.19	28.40	0.01	–	–	0.13	0.01	22.01	129.59	36.99	0.08
Internet autoguide	4.33	68.24	4.01	6.26	4.42	1.00	71.50	1.00	0.05	1.00	15.73	31.38	5.15	0.03
Linked in	6.06	29.18	1.87	3.34	1.66	0.01	34.56	2.35	0.02	0.01	16.15	26.32	2.11	0.02
All conferences	6.34	43.01	3.68	3.32	1.88	0.01	18.54	1.00	0.05	0.01	18.49	36.57	7.96	0.03
Mbendi	1.64	3.98	2.08	1.28	0.75	1.00	0.97	0.02	0.00	0.01	13.82	19.06	31.93	0.06
Net lib	6.65	7.86	4.43	2.31	0.12	0.00	0.27	0.03	0.00	0.00	13.82	38.57	6.34	0.05
RD learning	2.24	4.96	1.44	1.36	0.33	0.01	3.51	0.01	0.01	0.01	12.43	11.31	1.94	0.00

Table 7 continued

Dataset	SoftMealy		Wien		RoadRunner		FivaTech		Trinity		Aleph		Roller	
	LT	ET	LT	ET	LT	ET	LT	ET	LT	ET	LT	ET	LT	ET
Web MD	4.54	20.85	10.47	10.17	14.49	0.01	7.64	1.00	0.01	0.01	18.24	46.42	1.05	0.00
Ame. Medical Assoc.	4.16	7.30	3.68	7.39	–	–	1.53	0.20	0.03	1.00	16.05	33.56	3.70	0.02
Dentists	1.56	5.67	1.36	1.28	0.39	0.01	4.86	0.05	0.01	1.00	12.78	9.40	4.06	0.00
Dr. score	2.81	17.51	2.07	2.51	1.01	1.00	32.29	1.00	0.01	0.01	13.89	17.72	11.17	0.08
Steady health	4.85	40.00	5.95	6.78	7.61	0.02	5.71	0.08	0.30	0.01	34.05	97.53	21.67	0.05
Insight into diversity	4.90	11.67	3.20	4.99	2.41	1.00	9.35	0.08	0.05	1.00	13.40	50.57	39.38	0.02
4 jobs	4.07	36.60	6.09	6.57	1.40	1.00	8.32	0.07	0.04	0.01	17.08	37.44	9.25	0.08
6 figure jobs	7.65	18.51	7.44	8.32	13.83	1.00	53.21	0.24	0.02	0.01	12.20	71.17	4.71	0.03
Career builder	5.20	35.44	5.62	5.39	5.75	0.01	136.03	0.22	0.03	1.00	16.24	34.12	19.92	0.05
Job of mine	3.63	15.61	2.86	3.32	1.49	0.01	30.33	0.14	0.03	0.01	15.42	30.23	33.56	0.05
Yahoo!	16.32	86.55	15.62	16.75	–	–	246.95	0.89	0.02	1.00	–	–	10.47	0.06
Haart	4.03	69.80	3.27	4.82	3.12	0.01	9.40	0.06	0.02	1.00	17.36	41.82	10.47	0.06
Homes	4.36	45.57	4.80	8.18	2.43	0.01	–	–	0.11	0.01	13.79	23.64	29.33	0.05
Remax	3.09	17.80	8.30	5.17	7.85	0.01	47.73	0.07	0.22	0.01	16.40	36.77	4.71	0.03
Trulia	11.87	196.63	12.94	25.37	19.97	1.00	–	–	0.48	1.00	18.95	119.56	7.00	0.02
Player profiles	2.92	17.43	5.83	3.43	–	–	8.06	1.00	0.06	0.16	19.90	103.81	53.18	0.05
UEFA	4.00	23.08	2.69	3.97	6.89	0.02	–	–	0.03	0.01	18.20	31.93	9.81	0.02
ATP world tour	9.31	125.86	11.81	12.76	8.87	0.03	50.01	0.58	0.38	0.02	19.60	56.60	5.19	0.03
NFL	6.34	27.66	9.69	6.64	19.88	0.02	72.49	1.00	0.08	0.01	16.55	43.50	25.16	0.11
Soccer base	8.07	33.30	12.95	7.56	10.06	1.00	–	–	0.34	1.00	51.36	277.67	28.46	0.06
Amazon cars	0.68	7.02	8.41	5.66	0.80	1.00	2.55	1.00	0.01	1.00	13.39	11.00	9.70	0.08
UEFA players	1.43	11.64	3.79	2.58	0.41	0.01	12.58	0.03	0.01	0.01	16.93	18.68	8.64	0.06

Table 7 continued

Dataset	SoftMealy		Wien		RoadRunner		FivaTech		Trinity		Aleph		Roller	
	LT	ET	LT	ET	LT	ET	LT	ET	LT	ET	LT	ET	LT	ET
Amazon pop artists	3.61	16.20	7.96	10.22	1.03	1.00	107.04	0.08	0.01	0.01	–	–	16.93	0.09
UEFA teams	0.49	3.33	0.80	2.28	0.47	0.01	0.54	0.01	0.02	1.00	11.74	15.23	262.21	0.28
Aus open players	0.81	15.27	5.26	16.96	3.12	1.00	80.70	0.18	0.14	1.00	21.67	147.52	5.43	0.05
eBay bids	1.14	11.11	5.59	13.65	2.11	0.01	397.98	0.34	0.25	0.06	22.60	119.96	31.70	0.02
Major league baseball	1.33	13.67	4.39	4.28	1.75	0.01	184.47	1.00	0.02	0.01	21.28	13.79	3.43	0.05
Netflix films	2.26	23.88	4.85	31.55	4.73	0.01	399.01	0.53	0.08	0.01	17.60	105.26	0.64	0.00
RPM find packages	0.83	18.23	1.39	10.42	0.75	1.00	28.59	0.06	0.02	1.00	14.93	68.16	3.67	0.05
Bigbook	1.61	114.85	1.78	62.29	14.27	1.00	–	–	0.06	1.00	17.81	48.88	26.82	0.05
IAF	1.77	9.23	0.90	1.67	0.44	0.01	7.00	0.04	0.07	0.01	68.21	17.15	11.34	0.06
Okra	0.78	150.01	1.13	24.48	10.39	1.00	425.36	0.26	0.06	0.01	228.41	28.76	103.19	0.11
LA weekly	1.30	23.45	0.57	1.89	0.48	0.01	2.69	0.03	0.01	0.01	13.27	16.60	20.34	0.09
Zagat	1.50	14.11	5.83	13.75	3.85	1.00	73.51	0.04	0.07	1.00	13.54	19.61	28.19	0.16



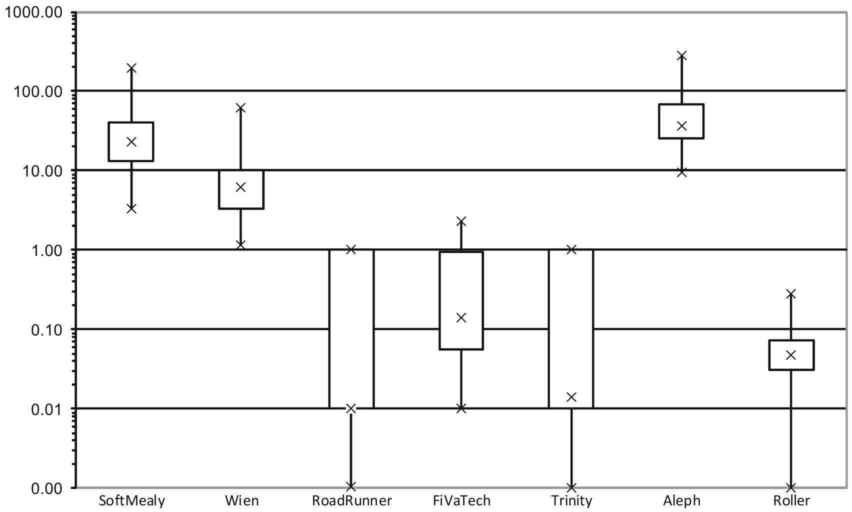
	SoftMealy	Wien	RoadRunner	FiVaTech	Trinity	Aleph	Roller
Quartile 1	2.12	2.23	1.01	7.32	0.02	13.82	4.82
Minimum	0.49	0.57	0.12	0.27	0.00	10.08	0.64
Median	4.25	4.41	2.41	28.59	0.03	16.32	9.76
Maximum	23.45	19.47	28.40	425.36	0.67	228.41	262.21
Quartile 3	6.76	7.10	7.61	73.00	0.11	19.67	27.85

Sample ranking		Iman-Davenport's	Hommel's							Statistical ranking	
Technique	Rank	p-value	adjusted p-values							Technique	Rank
Trinity	1.00	6.66E-16		RoadRunner	SoftMealy	Wien	Roller	FiVaTech	Aleph	Trinity	1
RoadRunner	3.28		Trinity	1.42E-14	6.66E-16	6.66E-16	3.25E-10	4.26E-14	2.22E-15	RoadRunner	2
SoftMealy	3.44									SoftMealy	2
Wien	3.57									Wien	2
Roller	4.83									Roller	2
FivaTech	5.87									FivaTech	2
Aleph	6.00									Aleph	2

Fig. 9 Summary of results regarding learning times

enough evidence to conclude that there is a statistically significant difference between Trinity, Roller, RoadRunner, and FiVaTech regarding extraction times, that is, they all rank at the first position. The test, however, finds enough evidence to reject the hypothesis that the previous proposals and the others behave similarly regarding extraction time. These results are very important, because they confirm that the rules that Roller learns are very competitive regarding efficiency.

As a conclusion, our experiments support the idea that Roller is very efficient. It is not the best performing regarding learning times, but it still lies within the range of seconds, which we do not think is a serious shortcoming from a practical point of view. However, the rules that it learns are as efficient as the rules that other state-of-the-art proposals can learn, which makes them competitive from a practical point of view. The reason why Roller takes a little more time to learn a rule than other proposals is that it has to create several training sets and then apply the base learner several times; its efficiency clearly depends on how effective the base learner is. Anyway, we think that the efficiency results are quite reasonable and that its superiority regarding effectiveness clearly compensates for its slightly worse performance.



	SoftMealy	Wien	RoadRunner	FiVaTech	Trinity	Aleph	Roller
Quartile 1	13.17	3.33	0.01	0.06	0.01	25.89	0.03
Minimum	3.33	1.16	0.00	0.01	0.00	9.40	0.00
Median	23.00	6.15	0.01	0.14	0.01	36.99	0.05
Maximum	196.63	62.29	1.00	2.35	1.00	277.67	0.28
Quartile 3	40.75	10.21	1.00	0.94	1.00	67.60	0.07

Sample ranking		Iman-Davenport's	Hommel's							Statistical ranking	
Technique	Rank	p-value	adjusted p-values							Technique	Rank
Trinity	2.13	5.55E-16		Roller	RoadRunner	FiVaTech	Wien	SoftMealy	Aleph	Trinity	1
Roller	2.26		Trinity	5.67E-01	1.00E+00	2.73E-01	6.66E-16	6.66E-16	2.22E-15	Roller	1
RoadRunner	2.59									RoadRunner	1
FivaTech	3.67									FivaTech	1
Wien	4.87									Wien	2
SoftMealy	5.98									SoftMealy	2
Aleph	6.50									Aleph	2

Fig. 10 Summary of results regarding extraction times

5 Related work

In this section, we first provide an overview of the proposals in the literature that were specifically tailored to learning Web information extraction rules; we then delve into propositio-relational machine-learning proposals, which are closely related to ours, but have not been explored so far in our context; finally, we compare them with ours from a conceptual point of view.

5.1 Specific-purpose proposals

There are literally hundreds of proposals that were specifically tailored to learning Web information extraction rules in the context of semi-structured Web documents [11,56]. Thus, we restrict our attention to those that pioneered a new research path.

Many authors have devised techniques that work on the text of the input documents, namely: Kushmerick et al. [44] presented a proposal that learns two patterns of tokens that characterise the left and the right context of the information to extract; Hsu and Dung [34] presented a proposal that relies on using automata to model the structure of the information and regular patterns to control the transitions amongst states; Chidlovskii [12] and Muslea et al. [49] also explored the idea of learning automata and patterns; Crescenzi and Mecca [13] and Crescenzi and Merialdo [14] explored learning regular expressions to extract information; Chang and Kuo [10] explored a multiple string alignment technique; Arasu and Garcia-Molina [2] presented other proposals to learn regular expressions; and sleiman and Corchuelo [55,58] presented two proposals that are based on multi-string alignment techniques.

There are also many authors who have devised techniques that work on the DOM tree representation of the input documents, namely: Hogue and Karger [32] presented a proposal that is based on tree similarity; Park and Barbosa [50] devised a technique that combines tree matching and clustering; Shen and Karger [53] devised a heuristic-based proposal; Álvarez et al. [1] devised a proposal that relies on clustering, tree matching, string matching, and string alignment; Su et al. [61] presented a proposal that is based on aligning DOM trees using a maximum entropy model; and Kayed and Chang [38] introduced a technique that first learns an information schema and then a context-free grammar using a tree similarity and a tree alignment technique.

The previous techniques work on the documents themselves, that is, on their tokens or their nodes. A few authors have explored transforming the tokens or the nodes into vectors of attributive features that are related to others by means of relational features. (See Reference [19] to find details on an alternate first-order representation of documents, not necessarily semi-structured Web documents.) Such a representation allows to use techniques that got inspiration from inductive logic programming. Soderland [59] and Califf and Mooney [9] pioneered this research path with two proposals that learn ground first-order rules that work on the textual representation of the input documents; Bădică et al. [8] presented a technique that learns first-order rules with variables by applying the FOIL system to a first-order tree-based representation of the input documents. The previous techniques rely on quite a limited catalogue of built-in features; Freitag [24], Irmak and Suel [35], and Fernández-Villamor et al. [20] worked on proposals that learn first-order rules using open catalogues of features.

5.2 Exploring propositio-relational learning

Inductive logic programming is a natural approach to deal with relational data. Unfortunately, it is inefficient when the datasets scale in the number of data or features because the search space is typically huge [7,8,23,24,48]. This has motivated some authors to work on adapting efficient propositional techniques so that they can work on relational data. The proposals in the literature can be broadly classified as follows [29,40]: upgrading, flattening (aka. proposionalisation), and multiple view.

Upgrading proposals rely on a conventional propositional learner that is upgraded to deal with relational features. Some proposals upgrade a propositional learner with the ability to learn first-order rules, namely: TILDE [6] upgrades C4.5, SCART [41] upgrades CART, RIBL [18] and RIBL2 [33] upgrade k-NN, Cumby and Roth [15] and Gärtner et al. [26] upgraded some kernel methods, PRM [28,36] upgrades Bayesian networks, SLP [47] upgrades stochastic grammars, and 1BC and 1BC2 [21] upgrade Bayesian classifiers. Unfortunately, these proposals did not prove to be efficient enough [29], which motivated other authors to work on so-called relational database proposals that transform the original problems into SQL representations that can be handled more efficiently with commodity database management systems. There are two approaches in the literature: selection graph model, which includes MRDTL [3], which builds on TILDE but represents the data in SQL, and MRDTL2 [3], which is an optimised version of MRDTL that can also handle missing attributes using a proposal based on Naive Bayes classifiers; other proposals are based on a technique called tuple ID propagation, which basically attempts to join related vectors virtually; for instance, CrossMine [64] and GraphNB [65] follow this approach by extending FOIL [51] and a Bayesian classification algorithm, respectively.

Flattening proposals convert relational data into table-based representations to which standard propositional techniques can be applied. There are two approaches in the literature: creating universal vectors that join all of the data in the training sets, which was pioneered by LINUS [17], DINUS [45], and SINUS [43], or creating vectors that summarise and/or aggregate the data in the neighbourhood of every vector, e.g., RollUp [39] and RELAGGS [42].

Guo and Viktor [29] devised the only multiple view proposal of which we are aware. It relies on a meta-learning approach that can learn from multiple views of the data, that is, multiple subsets of data that result from projecting them using different feature subsets, and then integrates the results using a novel technique that does not require the complex preprocessing required by flattening proposals.

Although propositio-relational proposals seem very adequate to deal with the problem of learning information extraction rules, it remains an almost unexplored research path. The only exception is the work by Sleiman and Corchuelo [57], who devised a proposal that hybridises finite automata and neural networks; the states of the automata represent the information to be extracted and the transitions the next-token relational feature; the transitions are controlled by means of neural networks that recognise token patterns building on simple features like they their HTML tags or their lexical classes.

5.3 Discussion

Typically, researchers who are interested in Web information extraction have designed ad hoc proposals that are specifically tailored to this problem, which has led to a variety of alternatives. Although many of them were proven to be very effective and efficient, the problem is that they cannot leverage the many advances in the field of machine learning; neither can the general machine-learning field easily benefit from them. Furthermore, many of them have faded away quickly as their inherent assumptions about the structure of documents have become obsolete as the Web has evolved. Unfortunately, they could not be easily adapted to deal with such evolution because this would have required to rework them, that is, to have devised completely new proposals. Some of the ad hoc proposals that we have surveyed work on the textual representation of the input documents and their goal is to characterise the left and the right context of the information to extract; others work on their DOM tree representations and their goal is to characterise the path from the root node to the nodes that provide the information to be extracted.

Contrarily to the previous proposals, Roller can leverage many machine-learning techniques in the literature and benefit from the advances in this field. Furthermore, it is based on an open catalogue of features that can be easily extended and adapted as the Web evolves, without changing the proposal itself. Neither does Roller attempt to characterise the left or the right context of the information to extract or the path from the root to the nodes that provide the information to extract; but it tries to characterise a context in the DOM tree. Note that this may involve tokens in disparate positions, not necessarily on the left or the right, as well as tokens that are not on the same path to the root node, e.g., siblings or children of siblings.

A few authors have explored using techniques that got inspiration from inductive logic programming since the tokens or the DOM nodes of semi-structured documents can be naturally represented as relational data. Their proposals are expected to be easier to adapt as the Web evolves since they need not be adapted, but their catalogues of features. In general, they can achieve high effectiveness at the cost of efficiency. They explore an unbounded context, which does not restrict them to the left or the right context or nodes within a given path, as was the case for the ad hoc proposals. Unfortunately, they use the same heuristic to guide the search through both attributive and relational features; furthermore, although they all are analysed together in every step, only one of them is selected to grow the rule being learnt, which typically results in a problem called myopia. The reason is that when a relational feature is selected, the attributive features of the target node are not taken into account; in other words, there are cases in which a decision to explore a neighbouring node may lead to a local minimum. Except for L-Wrappers, none of the proposals that we have surveyed can backtrack to explore other choices. Note, too, that L-wrappers is the only proposal that advocates transforming the problem of Web information extraction into a first-order knowledge base and then learning extraction rules using an inductive logic system like FOIL. This proved not to be efficient enough, even with relatively simple documents. This problem was first pointed out by Freitag, who suggested that learning from a first-order representation would simply be too inefficient. Our experimental results using Aleph prove that the problem can be tackled using a pure inductive logic programming technique, but the rules learnt are not the most effective and they are inefficient.

Roller also works on a relational representation of the input documents that builds on an open catalogue of features that can easily evolve as the Web evolves, without making a change to the proposal itself. Furthermore, it relies on a propositional base learner that can be integrated in our proposal without a change; that is, it can benefit from the advances in the general field of machine learning. Our experiments prove that Roller is very effective and efficient. This is because it relies on a propositional learner to analyse the attributive features of the nodes to extract and then explores their context using relational features in an attempt to find neighbouring nodes whose attributive features can contribute to learning a better rule. Furthermore, two different search heuristics are involved: one that is provided by the base learner, which is ad hoc and was designed to guide the search through attributive features as effectively and efficiently as possible, and another one that was designed to guide the search through the relational features and helps explore the context as effectively and efficiently as possible. Roller also reduces myopia because it deals with all of the attributive features at the same time, not one after the other as was the case for the existing proposals; furthermore, the decision on which relational feature has to be explored next does not depend only on that feature itself, but on the attributive features of the target nodes. Obviously, this is not a solution to myopia, but our experiments prove that it reduces the odds of making wrong decisions; we explored using backtracking, but our experiments proved that the mechanism was not actually necessary, so we decided not to include it in the final version of Roller.

Since information extraction problems can be naturally represented using relational data, one might think that it would be easy to leverage a proposal from the field of propositio-relational learning. Unfortunately, few such proposals exist in the literature since there are a number of intrinsic problems: according to Guo and Viktor [29], upgrading proposals are not generally scalable-enough, chiefly those that rely on inductive logic programming approaches, and cannot generally achieve high effectiveness when they deal with numeric data, which is very common in our context, e.g., depth of a node, number of children, font size, ratio of letters or figures, text length, coordinates, and the like. Relational database proposals are more efficient because they rely on a database management system, but they do not seem easy to adapt to the problem of information extraction because they rely on a full-fledged relational schema, that is, they were designed to deal with actual relational databases that build on a rich data schema that includes information about every attribute, primary keys, foreign keys, and so on; in other words, they are schema-driven proposals. In our context, there is not such a schema, which requires the proposals to be instance-driven, that is: they must explore the context of every instance individually, without an explicit schema. The existing multi-view proposal in the literature improves on efficiency, but it does not seem appropriate in our context because it is based on aggregating neighbouring vectors. Numeric features are aggregated using the standard SQL aggregation functions (sum, average, minimum, maximum, standard deviation, and count), but categoric features are aggregated using counts only. That means that the classification power that those features can provide is lost, but they are very common in our context, e.g., font family, colour, horizontal alignment, floating specification, and the like; furthermore, it does not take into account that attributes in disparate nodes can contribute to obtaining a good rule. Flattening proposals require much computation to flatten the datasets to be analysed and the resulting vectors may have an arbitrarily large number of components, which hinders the applicability of many learners in practice; some of the proposals require data to be duplicated, which increases statistical skewness, whereas others require data to be aggregated, which implies that data distributions are neglected; furthermore, they need to put a limit to the amount of context that can be explored because the context of the data is not explored on-demand, but pre-computed.

Roller naturally fits within the category of flattening proposals, but it differs significantly from the existing ones: instead of pre-processing the vectors in an attempt to make the context of every node explicit, it first tries to learn a perfect rule building solely on the attributive features of the nodes to be extracted; if no such a rule can be learnt, then it explores the context by means of the available relational features, which involves flattening the vectors that correspond to the nodes being analysed and the vectors that correspond to their neighbours. This results in a dynamic flattening proposal that has proven to work very well in practice according to our experiments. Note that, contrarily to other existing proposals, no aggregation of data is required; it works on the attributive features themselves, which implies that no classification power is lost in the flattening process.

6 Conclusions

In this article, we have introduced Roller, which is a new proposal to learn Web information extraction rules in the context of semi-structured Web documents.

It is a highly configurable proposal: it relies on an open catalogue of both attributive and relational features, which helps adapt it as the Web evolves; furthermore, it does not commit to a specific base learner or rule scorer, but can leverage many proposals in the literature and thus benefit from the continuous advances in the general field of machine learning. This clearly deviates from the many existing ad hoc proposals in the literature and from the few existing proposals that are based on inductive logic programming techniques. Technically, the learner that underlies our proposal relies on a search procedure that uses a new dynamic flattening technique to explore the context of the nodes that provide the information to be extracted; our survey of the literature proves that is a novel approach to the problem.

We have conducted a series of experiments on a collection of 54 real-world datasets. The experiments confirm that our proposal is very effective and efficient in practice. It can outperform state-of-the-art proposals in terms of effectiveness and it is very competitive in terms of efficiency; although it is a little more inefficient than other proposals regarding learning times, it can still learn a rule in a matter of seconds, which we do not think is a serious shortcoming; the rules that it learns can, however, be executed as effectively as the rules learnt by other state-of-the-art proposals.

Our results clearly support our idea that using standard machine-learning techniques to learn Web information extraction rules is a promising approach. Note that this clearly deviates from the existing proposals in the literature, which build on ad hoc machine-learning techniques that were specifically tailored to the problem of learning Web information extraction rules. They have proved to be very effective, but the problem is that they tend to fade away because their learning components are not clearly differentiated, which makes it difficult to evolve them as the Web evolves and precludes reusing the many advances that are published in the general field of machine learning. Contrarily, Roller relies on a standard machine-learning technique and a standard rule scorer, which are reused as is, and an open catalogue of features, which can be easily extended. This proves that it makes sense to keep working on trying to use general-purpose machine-learning techniques instead of working on new ad hoc techniques.

Acknowledgments We are grateful to Dr. Hassan A. Sleiman, from the Commissariat à l'Énergie Atomique et aux Énergies Alternatives, LIST Institute, France, and Dr. Toñi Reina, from the University of Sevilla, for their help and support with previous proposals that have paved the way for Roller. We would also like to thank Dr. Francisco Herrera, from the University of Granada, Spain, for sharing his statistical analysis software with us, and also to Dr. José C. Riquelme, from the University of Sevilla, Spain, for the many fruitful discussions regarding evaluating our proposal and comparing our experimental results. Last, but not least, we would like to thank our reviewers and our editor for their insightful comments on the earlier versions of this article, which helped us to improve it very much. Our work was supported by the European Commission (FEDER), the Spanish and the Andalusian R&D&I programmes (Grants TIN2007-64119, P07-TIC-2602, P08-TIC-4100, TIN2008-04718-E, TIN2010-21744, TIN2010-09809-E, TIN2010-10811-E, TIN2010-09988-E, TIN2011-15497-E, and TIN2013-40848-R). The work by Patricia Jiménez was also partially supported by the University of Southern California during a research visit that she paid to the Information Sciences Institute.

Appendix 1: The experimentation environment

We performed our experiments on a four-threaded Intel Core i7 computer that ran at 2.93 GHz, had 4 GB of RAM, Windows 7 Pro 64-bit, Oracle's Java Development Kit 1.7.9_02, JTidy 9.38, and Weka 3.6.8. No changes were performed to the default configurations of the hardware or the software.

We used a collection 40 datasets on books, films, cars, events, doctors, jobs, realty, and players, plus the nine datasets from the ExAlg repository and the five datasets from the RISE repository that provide semi-structured documents. The categories regarding the first group of datasets were randomly sampled from The Open Directory sub-categories, and the websites inside each category were randomly selected from the 100 best-ranked websites between December 2010 and March 2011 according to Google's search engine; we downloaded 30 documents from each website and handcrafted a set of annotations with the slots that we wished to extract from each document. Table 8 describes our datasets; for each category, we report on the sites from which they were downloaded, the slots that model the information that they provide, the number of documents that they have, their average size in KiB, the average number of HTML errors that they have (as reported by JTidy), the average number of positive examples, and the average number of negative examples. The datasets were split ten times; in each split, we randomly selected six documents for training purposes and the remaining ones for testing purposes. The results on which we report in this article were obviously computed on the testing sets.

Table 8 Description of our datasets

Category	Site	Slots	Docs	Size (KiB)	Errors	Positives	Negatives
Jobs	Insight into diversity	Job{company, location, category}	30	30	67.07	30.00	12, 905.50
	4 jobs	Job{company, location, category}	30	80	110.47	30.00	9657.75
	6 figure jobs	Job{company, location, category}	30	73	169.37	30.00	18, 687.25
	Career builder	Job{company, location, category}	30	54	93.97	30.00	10, 055.00
	Job of mine	Job{company, location, category}	30	24	41.03	30.00	7210.75
Cars	Auto trader	Car{colour, doors, engine, mileage, model, price, transmission, type}	30	184	273.23	30.00	12, 462.67
	Car max	Car{colour, mileage, model, price, transmission, year, type}	30	67	191.47	30.00	9222.63
	Car zone	Car{colour, doors, engine, location, make, mileage, model, price, transmission, year, type}	30	71	118.80	30.00	7211.25
	Classic cars for sale	Car{colour, location, make, model, price, transmission, year, type}	30	76	25.03	28.90	10, 678.50
	Internet autoguide	Car{colour, doors, engine, location, mileage, price, transmission, type}	30	154	163.90	30.00	9656.78
ExAlg	Amazon cars	Car{make, model, price}	21	25	20.00	54.00	3179.25
	UEFA players	Player{name, country}	20	12	10.40	180.33	7010.67
	Amazon pop artists	Artist{name}	19	34	35.00	2805.00	18, 322.00
	UEFA teams	Team{association, country, FIFA affiliation, founded, general secretary, president, Press officer, team, UEFA affiliation}	20	7	32.80	19.90	5412.80
	Aus open players	Player{name, birth date, birth place, country, height, money, weight}	29	41	66.73	29.00	44, 437.13
	eBay bids	Bid{price, bids, location}	50	26	18.12	30.00	33, 393.00

Table 8 continued

Category	Site	Slots	Docs	Size (KiB)	Errors	Positives	Negatives
	Major league baseball	Player{name, position, team}	9	40	26.00	550.40	9544.80
	Netflix films	Film{title, director, length, year}	50	44	125.86	30.00	30,733.00
	RPM find packages	Package{name, description, operating system}	20	35	9.90	2562.00	22,359.00
Real estate	Haart	Property{address, bedrooms, price}	30	90	40.00	9.00	9662.00
	Homes	Property{address, bedrooms, bathrooms, size, price}	30	59	99.93	30.00	5974.33
	Remax	Property{address, bedrooms, bathrooms, size, price}	30	70	75.70	30.00	12,097.50
	Trulia	Property{address, bedrooms, bathrooms, size, price}	30	175	312.73	30.00	27,320.00
Doctors	Web MD	Doctor{name, address, phone, specialty}	30	59	24.10	30.00	12,872.00
	Ame. Medical Assoc.	Doctor{name, address, phone, specialty}	30	25	36.00	30.00	8682.00
	Dentists	Doctor{name, address, phone, fax, specialty}	30	12	103.27	28.00	2752.50
	Dr. score	Doctor{name, address, phone, specialty}	30	24	33.07	27.14	5679.00
	Steady health	Doctor{name, address, specialty}	30	81	24.00	30.00	24,811.00
Events	Linked In	Event{date, place, title, url}	30	10	23.67	29.00	9077.80
	All conferences	Event{date, place, title, url}	30	18	30.47	30.00	12,739.40
	Mbendi	Event{date, place, title, url}	30	7	27.00	30.00	6120.00
	RD learning	Event{date, place, title, url}	30	4	14.00	30.00	3388.80
Rise	Bigbook	Business{name, city, phone, street}	235	25	20.61	566.00	16,825.20
	IAF	Finder{name, email, organisation, service provider}	252	14	13.20	64.33	8791.17
	Okra	Citizen{name, email}	10	8	15.42	380.00	14,086.00
	LA weekly	Restaurant{ name, address, phone}	28	5	4.93	126.25	3508.75
	Zagat	Restaurant{name, address, type}	91	18	31.92	32.75	4894.25

Table 8 continued

Category	Site	Slots	Docs	Size (KiB)	Errors	Positives	Negatives
Films	Albania movies	Film{title, director, actor, year, runtime}	30	6	20.90	23.50	6698.33
	All movies	Film{title, director, actor, year, runtime}	30	34	32.33	78.33	36,428.33
	Disney movies	Film{title, actor, year, runtime}	30	47	59.40	30.00	6627.00
	IMDB	film{title, director, actor, year, runtime}	30	97	12.00	40.33	18,023.67
	Soul films	Film{title, director, actor, year}	30	28	66.13	65.40	17,985.00
Books	Abe books	Book{title, author, price, isbn}	30	38	58.73	35.60	8370.40
	Awesome books	Book{title, author, price, isbn, year}	30	20	43.27	37.17	6612.83
	Better world books	Book{title, author, price}	30	125	46.00	34.50	19, 716.50
	Many books	Book{title, author, year}	30	27	130.00	30.50	13, 515.25
	Waterstones	Book{title, author, price}	30	80	129.10	31.50	13,856.00
Players	Player profiles	Player{name, birth, hight, weight, club}	30	21	35.07	30.00	30,840.00
	UEFA	Player{name, birth, country, position}	30	63	31.80	30.00	8956.60
	ATP world tour	Player{name, birth, age, hight, weigth, country}	30	136	92.03	30.00	15,284.57
	NFL	Player{name, birth, hight, weigth, age, college}	30	95	84.16	30.00	12,654.71
	Soccer base	Player{name, birth, age, hight, weigth, country, position, club}	30	85	156.37	30.00	73,712.78

We searched the Web and contacted many authors in order to have access to the implementation of as many proposals as possible. We managed to find an implementation for SoftMealy [34] and Wien [44], which are classical proposals, and RoadRunner [14], FiVaTech [38], and Trinity [58], which are recent proposals. We also experimented with a straightforward approach in which we translated our datasets into first-order knowledge bases and then used Aleph [60] to induce rules.

Appendix 2: Performance measures

We collected the usual effectiveness measures, namely: precision (P), recall (R), and the F_1 score to combine them both (F_1). We also collected some efficiency measures, namely: learning time (LT) and extraction time (ET), both measured in CPU seconds.

Effectiveness measures are stable because the proposals that we have compared are deterministic, that is, they do not change when a proposal is run multiple times on the same datasets. Efficiency measures, on the contrary, are subject to external experimental conditions and may vary from execution to execution. We decided to measure CPU times because they are far more stable than user times; given that the proposals that we have compared are deterministic, that means that they follow the same execution paths every time that they are executed on the same dataset, which implies that they execute exactly the same machine-level instructions. IO activities were not a problem in our experimental study; the reason is that the proposals that we compared are CPU bound, not IO bound. In other words, they read the input documents, which typically takes less than a hundredth of a second, then run their algorithms in memory, and finally output the results to a file, which does not usually take more than a hundredth of a second.

To confirm the idea that CPU times are stable enough, we repeated each experiment 25 times and we averaged the timings after discarding a few outliers using the well-known Cantelli's inequality.[1] We studied these outliers to make sure that they were actual abnormal values. They were due to the fact that our University's cloud infrastructure was reset a couple of times while the experiments were running; that resulted in a few timings that were abnormally large due to the interferences caused by the temporary interruption of the computing service. The rest of the timings were quite stable; the small differences from run to run were mainly due to the performance of the memory cache system, which, in turn, depended on the other processes that were running on our University's cloud infrastructure.

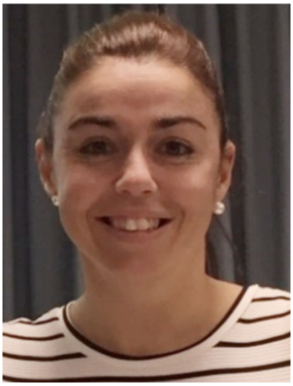
References

1. Álvarez M, Pan A, Raposo J, Bellas F, Cacheda F (2008) Extracting lists of data records from semi-structured web pages. Data Knowl Eng 64(2):491–509
2. Arasu A, Garcia-Molina H (2003) Extracting structured data from web pages. In: SIGMOD conference, pp 337–348
3. Atramentov A, Leiva H, Honavar V (2003) A multi-relational decision tree learning algorithm. In: ILP, pp 38–56
4. Batista GEAPA, Prati RC, Monard MC (2004) A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explor 6(1):20–29

[1] Cantelli's inequality states that given a random variable X, if μ denotes its mean and σ denotes is standard deviation, then the probability that $X \geq \mu + k\,\sigma$ or that $X \leq \mu - k\,\sigma$ is not greater than $1/(1+k^2)$. If we wish to discard, say, 5 % of the data as lower or upper outliers, we then have to set $1/(1+k^2) = 0.05$, that is $k = 4.36$.

5. Bernstein PA, Haas LM (2008) Information integration in the enterprise. Commun ACM 51(9):72–79
6. Blockeel H, Raedt LD (1998) Top-down induction of first-order logical decision trees. Artif Intell 101(1–2):285–297
7. Blockeel H, Raedt LD, Jacobs N, Demoen B (1999) Scaling up inductive logic programming by learning from interpretations. Data Min Knowl Discov 3(1):59–93
8. Bădică C, Bădică A, Popescu E, Abraham A (2007) L-wrappers: concepts, properties and construction. Soft Comput 11(8):753–772
9. Califf ME, Mooney RJ (2003) Bottom-up relational learning of pattern matching rules for information extraction. J Mach Learn Res 4:177–210
10. Chang C-H, Kuo S-C (2004) OLERA: Semisupervised web-data extraction with visual support. IEEE Intell Syst 19(6):56–64
11. Chang C-H, Kayed M, Girgis MR, Shaalan KF (2006) A survey of web information extraction systems. IEEE Trans Knowl Data Eng 18(10):1411–1428
12. Chidlovskii B (2001) Wrapping web information providers by transducer induction. In: ECML, pp 61–72
13. Crescenzi V, Mecca G (2004) Automatic information extraction from large websites. J ACM 51(5):731–779
14. Crescenzi V, Merialdo P (2008) Wrapper inference for ambiguous web pages. Appl Artif Intell 22(1&2):21–52
15. Cumby CM, Roth D (2003) On kernel methods for relational learning. In: ICML, pp 107–114
16. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
17. Džeroski S, Lavrač N (1993) Inductive learning in deductive databases. IEEE Trans Knowl Data Eng 5(6):939–949
18. Emde W and Wettschereck D (1996) Relational instance-based learning. In ICML, pp 122–130
19. Esposito F, Ferilli S, Fanizzi N, Basile TMA, Mauro ND (2003) Incremental multistrategy learning for document processing. Appl Artif Intell 17(8–9):859–883
20. Fernández-Villamor JI, Iglesias CÁ, Garijo M (2012) First-order logic rule induction for information extraction in web resources. Int J Artif Intell Tools 21(6):1–20
21. Flach PA, Lachiche N (2004) Naive bayesian classification of structured data. Mach Learn 57(3):233–269
22. Frank E, Hall MA, Holmes G, Kirkby R, Pfahringer B, Witten IH, Trigg L (2010) Weka-a machine learning workbench for data mining. In: Maimon O, Rokach L (eds) Data mining and knowledge discovery handbook. Springer, Berlin, pp 1269–1277
23. Freitag D (1998) Information extraction from HTML: application of a general machine learning approach. In: AAAI/IAAI, pp 517–523
24. Freitag D (2000) Machine learning for information extraction in informal domains. Mach Learn 39(2/3):169–202
25. García S, Herrera F (2008) An extension on 'statistical comparisons of classifiers over multiple data sets' for all pair-wise comparisons. J Mach Learn Res 9:2677–2694
26. Gärtner T, Lloyd JW, Flach PA (2004) Kernels and distances for structured data. Mach Learn 57(3):205–232
27. Geng L, Hamilton HJ (2006) Interestingness measures for data mining: a survey. ACM Comput Surv 38(3):1–32
28. Getoor L, Friedman N, Koller D, Taskar B (2001) Learning probabilistic models of relational structure. In: ICML, pp 170–177
29. Guo H, Viktor HL (2008) Multirelational classification: a multiple view approach. Knowl Inf Syst 17(3):287–312
30. He H, Garcia EA (2009) Learning from imbalanced data. IEEE Trans Knowl Data Eng 21(9):1263–1284
31. Hickson I, Berjon R, Faulkner S, Leithead T, Navara ED, O'Connor E, Pfeiffer S (2014) HTML 5: a vocabulary and associated APIs for HTML and XHTML. Technical report W3C
32. Hogue AW, Karger DR (2005) Thresher: automating the unwrapping of semantic content from the world wide web. In: WWW, pp 86–95
33. Horváth T, Wrobel S, Bohnebeck U (2001) Relational instance-based learning with lists and terms. Mach Learn 43(1/2):53–80
34. Hsu C-N, Dung M-T (1998) Generating finite-state transducers for semi-structured data extraction from the web. Inf Syst 23(8):521–538
35. Irmak U, Suel T (2006) Interactive wrapper generation with minimal user effort. In: WWW, pp 553–563
36. Jaeger M (2008) Probabilistic-logic models: reasoning and learning with relational structures. In: SCAI, pp 197–200
37. Kavurucu Y, Senkul P, Toroslu IH (2011) A comparative study on ILP-based concept discovery systems. Expert Syst Appl 38(9):11598–11607

38. Kayed M, Chang C-H (2010) FiVaTech: page-level web data extraction from template pages. IEEE Trans Knowl Data Eng 22(2):249–263
39. Knobbe AJ, de Haas M, Siebes A (2001) Propositionalisation and aggregates. In: PKDD, pp 277–288
40. Kramer S, Lavrač N, Flach P (2001a) Propositionalization approaches to relational data mining. In: Džeroski S, Lavrač N (eds) Relational data mining. Springer, Berlin, pp 262–291
41. Kramer S, Widmer G, Pfahringer B, de Groeve M (2001b) Prediction of ordinal classes using regression trees. Fundam Inform 47(1–2):1–13
42. Krogel MA (2005) On propositionalization for knowledge discovery in relational databases. PhD thesis, Otto von Guericke Universität Magdeburg
43. Krogel M-A, Rawles S, Zelezný F, Flach PA, Lavrač N, Wrobel S (2003) Comparative evaluation of approaches to propositionalization. In: ILP, pp 197–214
44. Kushmerick N, Weld DS, Doorenbos RB (1997) Wrapper induction for information extraction. IJCAI 1:729–737
45. Lavrač N, Džeroski S (1994) Inductive logic programming: techniques and applications. Ellis Horwood, Chichester
46. Montoto P, Pan A, Raposo J, Losada J, Bellas F, Carneiro V (2008) A workflow language for web automation. J UCS 14(11):1838–1856
47. Muggleton S (2000) Learning stochastic logic programs. Electron Trans Artif Intell 4(B):141–153
48. Muggleton S, Raedt LD, Poole D, Bratko I, Flach PA, Inoue K, Srinivasan A (2012) ILP turns 20: biography and future challenges. Mach Learn 86(1):3–23
49. Muslea I, Minton S, Knoblock CA (2001) Hierarchical wrapper induction for semistructured information sources. Auton Agents Multi-Agent Syst 4(1/2):93–114
50. Park J, Barbosa D (2007) Adaptive record extraction from web pages. In: WWW, pp 1335–1336
51. Quinlan JR, Cameron-Jones RM (1995) Induction of logic programs: FOIL and related systems. New Gener Comput 13(3&4):287–312
52. Sarawagi S (2008) Information extraction. Found Trends Databases 1(3):261–377
53. Shen YK, Karger DR (2007) U-REST: an unsupervised record extraction system. In: WWW, pp 1347–1348
54. Sheskin DJ (2012) Handbook of parametric and nonparametric statistical procedures, 5th edn. Chapman and Hall/CRC, Boca Raton/London
55. Sleiman HA, Corchuelo R (2013a) TEX: an efficient and effective unsupervised web information extractor. Knowl Based Syst 39:109–123
56. Sleiman HA, Corchuelo R (2013b) A survey on region extractors from web documents. IEEE Trans Knowl Data Eng 25(9):1960–1981
57. Sleiman HA, Corchuelo R (2014a) A class of neural-network-based transducers for web information extraction. Neurocomputing 135:61–68
58. Sleiman HA, Corchuelo R (2014b) Trinity: on using trinary trees for unsupervised web data extraction. IEEE Trans Knowl Data Eng 26(6):1544–1556
59. Soderland S (1999) Learning information extraction rules for semi-structured and free text. Mach Learn 34(1–3):233–272
60. Srinivasan A (2004) The Aleph manual. Technical report, University of Oxford
61. Su W, Wang J, Lochovsky FH (2009) ODE: ontology-assisted data extraction. ACM Trans. Database Syst. 34(2):12.1–12.35
62. Turmo J, Ageno A, Català N (2006) Adaptive information extraction. ACM Comput Surv 38(2):1–47
63. van Kesteren A, Gregor A, Russell A, Berjon R (2014) Document object model 4. Technical report W3C
64. Yin X, Han J, Yang J, Yu PS (2006) Efficient classification across multiple database relations: a crossmine approach. IEEE Trans Knowl Data Eng 18(6):770–783
65. Zhang H, Su J (2004) Conditional independence trees. In: ECML, pp 513–524



Patricia Jiménez is working as a lecturer for the University of Seville. She earned her PhD degree with a thesis in which she presented a number of techniques to extract information from semi-structured web documents and a method to rank web information extractors. Her current research interests focus on open information extraction in the context of semi-structured web sites.

Rafael Corchuelo works for the University of Sevilla as a Reader. His research focus is on Enterprise Application and Information Integration, with an emphasis on technologies to extract data from both semi-structured web sites and social media in as an unsupervised manner as possible. He often works with start-ups so as to transfer his results to the industry.