Expert Systems With Applications 199 (2022) 116700

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

On validating web information extraction proposals

Patricia Jiménez *, Rafael Corchuelo

Universidad de Sevilla, ETSI Informática, Avda. Reina Mercedes s/n, Sevilla, E-41012, Spain

ARTICLE INFO

Keywords:
Web information extractors
Validation method

ABSTRACT

Many people who have to make informed decisions in today's always-on culture use information extractors to feed their systems with information that comes from human-friendly documents. Unfortunately, many proposals that validate information extractors have deficiencies that make it difficult to perform homogeneous comparisons, confirm or refute performance hypotheses, or draw unbiased conclusions. Consequently, it is very difficult to select the best-performing proposal on a sound basis. The state-of-the-art validation method overcomes many deficiencies in the previous proposals, but still overlooks the following issues: completeness of the validation datasets, that is, whether they provide a complete set of annotations or not; structure of the information, that is, whether they check the structure of the record instances extracted or just the attribute instances; and, finally, how extractions and annotations are matched. The decisions made regarding the previous issues have an impact on the effectiveness results. In this article, we have exhaustively analysed the literature and we have also highlighted the main weaknesses to tackle. We present a guideline and a method to compute the effectiveness, which complements and enhances the state-of-the-art validation method.

1. Introduction

Today's always-on culture is pushing forward a new generation of systems that help people make informed decisions building on the information that they extract from human-friendly documents on the Web. In the literature, there are many proposals to implement information extractors (Baumgartner et al., 2018; Chang et al., 2006; Ferrara et al., 2014; Sleiman & Corchuelo, 2013; Turmo et al., 2006). Many of them require the user to provide a learning dataset with some sample documents from which extraction rules are learnt; depending on whether the learning dataset is required to provide annotations or not, the learning method is said to be supervised or unsupervised, respectively. There are also heuristic-based proposals that use built-in rules that have proven to work well with many different documents. The information extracted by a supervised proposal has user-defined labels with a meaning; contrarily, the information extracted by unsupervised or heuristic-based proposals have computer-generated labels that must be mapped onto user-defined labels later.

Jiménez et al. (2016) found that the methods used to validate information extractors are often poorly documented and have some common deficiencies that may have a significant impact on the experimental results. This is particularly important insofar it hampers confirming or refuting the results, makes the comparisons with other proposals heterogeneous, and may easily bias the conclusions. They devised ARIEX, which is the state-of-the-art method to validate information extraction proposals. Unfortunately, we have found out that there are three important issues that it does not take into account, namely: (a) Completeness of the validation datasets, that is, to what extent the validation dataset has been annotated. They are commonly fully annotated in the context of supervised proposals, but partially annotated in the context of unsupervised and heuristic-based proposals. (b) The structure of the information and how the validation process took it into account. The information is commonly structured as collections of (possibly nested) attributes or record instances. And (c) how extractions and annotations are matched. The matching strategies are commonly classified as exact, contains, or overlapping matchings. The decisions made regarding the previous issues have an impact on the way that confusion matrices are computed, which, in turn, has an impact on the effectiveness measures used to make comparisons and rankings.

In this article, we advocate that researchers who use ARIEX to validate their proposals must also report on the completeness of their validation datasets, on how they take the structure of the information into account, and how they compute matches amongst the annotations and the extractions; we also describe a method to compute confusion matrices that takes the previous decisions into account. This constitutes a novel contribution since it complements the state-of-the-art validation method with additional guidelines regarding issues that were overlooked previously.

The rest of the article is organised as follows: in Section 2, we report on the most closely-related proposals and the extent to which they have

* Corresponding author.
E-mail addresses: patriciajimenez@us.es (P. Jiménez), corchu@us.es (R. Corchuelo).

https://doi.org/10.1016/j.eswa.2022.116700
Received 17 November 2020; Received in revised form 7 February 2022; Accepted 19 February 2022
Available online 19 March 2022

overlooked the three issues that were mentioned previously; in Section 3, we present some preliminary concepts that are used throughout the article; in Section 4, we describe the issues and propose a guideline to extend ARIEX; in Section 5, we present our method to compute the confusion matrix and the effectiveness measures according to the guideline; in Section 6, we report on the results of our experimentation; finally, we present our conclusions in Section 7.

2. Related work

Validating an information extractor amounts to confronting it with a series of documents and checking the extent to which it can extract information from them. The results allow to compare an extractor to its competitors and to rank them according to different effectiveness measures.

The first records of formal methods to validate information extractors date back to the end of the last century (Chinchor et al., 1993; Hirschman, 1998; Lehnert & Sundheim, 1991). They were intended to semi-automatically compare proposals to extract information from free-text documents in the context of the well-known MUC conferences. Later, Lavelli et al. (2004), Ireson et al. (2005), and Lavelli et al. (2008) pointed out a few common mistakes in the previous methods and improved on them. Since then, there has been an increasing interest in extracting data from semi-structured documents (Baumgartner et al., 2018; Chang et al., 2006; Ferrara et al., 2014; Roldán et al., 2020; Sleiman & Corchuelo, 2013; Turmo et al., 2006) in which the information is written in forms, listings, or tables. We have reviewed most of the proposals and our conclusion is that they do not typically unveil some key details regarding the validation process. Most of the articles simply present the results of a new approach and there is rarely a detailed analysis to ensure that the same methodology is used across different experiments. Furthermore, the validation process is typically poorly documented and there exists much heterogeneity in the experimental settings; in a few cases, no experimental results are reported at all (Baumgartner et al., 2007; Raposo et al., 2002; Sahuguet & Azavant, 2001).

Jiménez et al. (2016) devised ARIEX, which is the state-of-the-art method to validate information extractors. It allows to check them on a collection of well-known datasets and allows to compare the effectiveness results as homogeneously as possible and to rank them as automatically as possible. However, our recent experience with devising new information extractors (Jiménez & Corchuelo, 2016a, 2016b; Jiménez et al., 2021, 2020; Roldán et al., 2017, 2020, 2021) reveals that it can be further improved to take some additional issues into account, namely: (a) whether the validation datasets are completely or partially annotated; (b) whether they contain record values or not and how their structure is taken into account to compute the effectiveness measures; and (c) how the matchings amongst the annotations and the extractions are computed.

Regarding the completeness of the validation datasets, we can make a distinction between proposals that seem to have been validated manually (Park & Barbosa, 2007) or automatically (Crescenzi et al., 2001; Freitag, 2000; Hsu & Dung, 1998; Jiménez & Corchuelo, 2016a; Kayed & Chang, 2010; Kushmerick et al., 1997; Shen & Karger, 2007; de Sitter & Daelemans, 2003; Sleiman & Corchuelo, 2014; Zhang et al., 2015). In the first case, a user analyses the extractions made by the technique, decides on whether they are correct or not, and then computes some effectiveness measures; clearly this method is subjective and may introduce biases very easily. In the second case, a validation dataset is provided and the extractions are somehow matched with the annotations automatically so that the effectiveness measures can also be computed automatically; clearly an automatic method is preferable and less biased. Unfortunately, some authors did not provide any clues on how they created their validation datasets (Álvarez et al., 2010; Hogue & Karger, 2005; Irmak & Suel, 2006; Zhai & Liu, 2005). One might assume that they were annotated completely to compute the effectiveness measures, but the authors emphasised that they spent very little time on annotating and supervising their proposals; thus, it is not clear how the annotations were made and whether they were partial or complete. Irmak and Suel (2006) just mentioned that their technique worked well with a single randomly chosen document for training and ten documents for validation. Hogue and Karger (2005) did not provide any effectiveness measures; they only evaluated if their proposal was able to learn a good information extractor for one site. Although the datasets were annotated, how they performed the validation was not documented. It is unclear if the evaluation by Álvarez et al. (2010) was manual or not since their technique is unsupervised and they reported on precision and recall on a set of 200 sites that were not documented; neither was it documented how the effectiveness measures were computed automatically. Zhai and Liu (2005) used 49 sites from which 72 documents were collected. No clue was provided regarding whether they learnt and validated their information extractors with different documents from the same site or how the validation dataset was created. The authors reported on correctness, which was surely computed automatically, but it is not clear if the validation dataset was complete or not since they emphasised that their goal was to reduce the annotation effort. Unfortunately, none of the proposals surveyed documented whether spurious information was extracted or not. That is, it is likely that the extractions did not match some annotations perfectly. In such a case, the information that does not match the annotations may be spurious or count as false positives. If the validation dataset was partially annotated, we cannot make sure if that information is actually a false positive or a missed true positive, so that counting it as spurious information seems to make sense.

Regarding the structure of the information, none of the proposals provide any details regarding how they validated the structure of the records. Some proposals are supposed to learn a template for the documents in a site (Crescenzi et al., 2001; Kayed & Chang, 2010), others learn a template for records and attributes (Sleiman & Corchuelo, 2014), other can identify data regions only (Sleiman & Corchuelo, 2013), others can extract records only (Park & Barbosa, 2007; Shen & Karger, 2007), others extract relations between attributes only (Zhang et al., 2015), others can extract only attributes (Freitag, 2000; de Sitter & Daelemans, 2003), and many of them can extract records and attributes without an explicit schema (Álvarez et al., 2010; Hogue & Karger, 2005; Hsu & Dung, 1998; Irmak & Suel, 2006; Jiménez & Corchuelo, 2016a; Kushmerick et al., 1997; Zhai & Liu, 2005). Unfortunately, it is difficult to guess how the effectiveness measures were computed according to the type of information extracted, which is especially tricky when a proposal first extracts records and then extracts the attributes within them, but also when the proposal is able to deal with nested records and attributes. For instance, Trinity (Sleiman & Corchuelo, 2014) is supposed to extract records and attributes, both within other records or in isolation. However, the authors did not mention how they computed the effectiveness measures regarding the records. They just explained how to compute them at the attribute level. Summing up, declaring how the effectiveness measures are computed regarding the structure of the validation datasets should be a must.

Regarding the matching amongst annotations and extractions, Freitag (2000) stated that it is commonly assumed that the matchings must be exact. However, Lavelli et al. (2008) recommended that this should be made explicit because it is not always crystal clear. We agree with them since making assumptions might lead to biased and misleading conclusions. Only two of the proposals surveyed (Freitag, 2000; de Sitter & Daelemans, 2003) made it explicit the kind of matching performed; they both identified the problem and proved that the way the matching is interpreted has a significant impact on precision and recall. Intuitively, correct matchings should be exact, but this interpretation might be very stringent, so the criterion to be used should be established according to the goal of the system. If precision is very important, then exact matching should be used; if having high recall is more important, chiefly if some post-processing can be applied

to the extracted information, then contains matching should be used. Unfortunately, it is not clear to us when overlapping matching may be a sensible choice.

Summing up, the validations performed in the literature are very diverse and many details have commonly not been unveiled, which makes it difficult to determine which proposal actually performs better than the others. ARIEX was developed on the hope to help researchers validate their proposals (Jiménez et al., 2016), but an in-depth analysis of the literature has revealed three deficiencies that motivated us to work on this article, namely: how complete the annotations in the validation datasets are, how the structure of the extracted information is taken into account, and how the matchings amongst the annotations and the extractions are computed.

3. Preliminaries

In this section, we present some preliminaries that basically introduce the vocabulary used throughout the article.

Definition 1 (*Notation*)**.** A mapping from set X onto set Y is a function that establishes correspondences between the elements of both sets. X is referred to as the domain of the mapping and Y as its range. We denote the set of mappings from X onto Y as $X \mapsto Y$; given a mapping $m \in X \mapsto Y$ we denote its domain as $\operatorname{dom} m$ and its range as $\operatorname{ran} m$; we denote the extension of mapping m as $\{x_1 \mapsto y_1, x_2 \mapsto y_2, \dots, x_n \mapsto y_n\}$; the components of a mapping are referred to as correspondences. Without any loss of generality, we assume that mappings are implicitly sorted according to an arbitrary total pre-order on their domains, e.g., the standard lexicographic total pre-order.

Definition 2 (*Values*)**.** A value is a string of tokens. We denote the set of all values as V; given a value $v \in V$, we denote its length in tokens as $|v|$. In our proposal, we use the following kinds of tokens: words (sequences of letters, digits, and dashes), blanks (sequences of spaces, tabulators, line feeds, carriage returns, and form feeds), and other symbols (punctuation symbols, currency symbols, math symbols, and the like). Given a mapping $m \in X \mapsto Y$, we denote its value as $\widetilde{m}$ and define it as its textual serialisation; given a correspondence c in mapping m, we also denote its value as $\widetilde{c}$.

Definition 3 (*Datasets*)**.** An attribute is a label that we use to group values that have the same semantics. We denote the set of all attributes as A. An instance of a record is a mapping from $A \mapsto V$, where each correspondence is an instance of an attribute. A dataset is a collection of record instances that must be extracted from a collection of documents (positive dataset) or must not be extracted from them (negative dataset). A validation dataset is a tuple of the form (P, N), where P denotes a positive dataset and N a negative dataset. A dataset that provides the record instances that have been actually extracted from a document using an information extractor is referred to as an extracted dataset. The record instances in a validation dataset are referred to as annotations; the record instances in an extracted dataset are referred to as extractions. For the sake of readability, we assume that A can be decomposed into subsets A_V, which provides the attributes used in validation datasets, and A_E, which provides the attributes used in extracted datasets. Given an attribute $a \in A$ and a dataset D, we denote the set of instances of a in D as $\operatorname{instances}(a, D)$.

Note 1. Our definition requires datasets to be composed of flat record instances, which does not imply any loss of generality. Given an arbitrary real-world dataset with attribute instances and possibly nested record instances, one can transform it into our model as follows: every record-based dataset can be represented as a tree in which the nodes are the attribute or the record instances and the edges represent the containment relationships amongst them; to transform it into our model, it suffices to represent that tree as a flat collection of paths from the top level records to their attributes; top-level attributes can be transformed into records by grouping them into a fictitious global record.

Definition 4 (*Similarity*)**.** We have devised the following similarity measure: $\operatorname{sim}(v, w) = 1 - (i + d)/(i + d + p)$, where v and w denote two values, $(i, d, p) = \operatorname{diff}(v, w)$, and diff denotes the classical diff algorithm (Hunt & McIlroy, 1976). When the diff algorithm is applied to values v and w, it finds the tokens that must be, respectively, inserted, deleted, or preserved in value v to transform it into value w; we do not actually require to know the exact tokens to be inserted, deleted, or preserved, but the number of such tokens, which we denote as (i, d, p).

Note 2. The literature provides a variety of similarity measures (Yu et al., 2016), each of which was devised to address particular problems. We resorted to the previous definition of similarity because it interprets the concept as the percentage of changes that must be applied to a value in order to transform it into another value. This is a problem-agnostic formulation that proved to work very well to determine the mappings amongst the annotations and the extractions (which are generally similar, but not identical). Exploring other similarity measures and determining which one maximises the difference between the effectiveness measures attained with previous validations and our proposal would have biased it.

4. Guideline

In this section, we present a guideline that complements the ARIEX method (Jiménez et al., 2016). It describes our three new recommendations, which are aligned with the three issues that we have identified regarding validating information extractors.

4.1. Completeness of validation datasets

We recommend that researchers must report on the degree of completeness of their validation datasets.

A validation dataset is complete if it provides an annotation for every piece of information to be extracted. In the context of web-scale, unsupervised, or heuristic-based information extractors, producing complete validation datasets is a very difficult and error-prone task, mainly due to the human effort required to manually annotate web documents and to polish the annotations. As a conclusion, we think that typical web-scale validation datasets are partially annotated, which implies that confusion matrices can only be computed partially; this, in turn, has an impact on the resulting effectiveness measures. For instance, a piece of information that is extracted but does not correspond to any annotations in the validation dataset cannot be computed as a false positive because it is not possible to discern if it must not have been extracted or if it was simply not annotated.

4.2. Structure of information

We recommend that researchers must make it explicit how they validated the structure of the information extracted.

We have found that researchers focus on computing effectiveness measures on a per-attribute basis and then average the results to compute per-record measures, if any. Simply put, complex record structures are neglected since the information is basically dealt with as if it was organised into tuples with simple attributes. As a conclusion, the ability of an information extractor to extract information that is properly structured is not taken into account when computing the effectiveness measures.

4.3. Matching extractions and annotations

We recommend that researchers must make it explicit how they computed the matches amongst annotations and extractions.

The standard in the literature is that matches are computed as true positives, unmatched extractions are computed as false positives, unmatched annotations are computed as false negatives, and true negatives are computed from the annotations that explicitly refer to pieces of information not to be extracted. The problem is regarding the definition of matching (Lavelli et al., 2008), namely: (a) exact matching, which requires the annotated and the extracted values to be exactly the same to be considered a matching; (b) contains matching, which requires the annotated value to contain the extracted value; and (c) overlapping, which requires the annotated and the extracted values to have some tokens in common. We recommend that partial matches must contribute to the counter of true positives (the part of an extraction that coincides with its corresponding annotation), the counter of false positives (the part of the extraction that is not in the annotation), and the counter of false negatives (the part of the annotation that is not in the extraction).

5. Computing effectiveness

In this section, we present our method to compute a confusion matrix, which is the basis to compute a variety of standard effectiveness measures (Ferri et al., 2009; Sokolova & Lapalme, 2009), and the degree of spuriousness, which helps understand them better.

It works on a tuple of the form (P, N, E, p), where (P, N) is a validation dataset, E is an extracted dataset, and p indicates whether the validation dataset is complete or not. It performs the following steps: it first finds a mapping from the attributes in the validation dataset onto the attributes in the extracted dataset; it then re-normalises the record instances in the extracted dataset according to the previous mapping; next, it finds a mapping amongst the record instances in the validation dataset and the extracted dataset; finally, it computes a confusion matrix and a spuriousness degree.

The following subsections describe the details behind each step. Two of them rely on a general algorithm to compute mappings from a similarity matrix, which is presented at the end of the section.

5.1. Find an attribute mapping

The goal is to find a mapping that makes it explicit the correspondences between the attributes used in the validation dataset and the attributes used in an extracted dataset. This is trivial in cases in which the information extractor was learnt so that it labels the values that it extracts with the labels in the validation dataset, which is typically the case of information extractors that are learnt supervisedly. It is more involved in cases in which the information extractor uses computer-generated labels that have nothing to do with the labels used in the validation dataset, which is typically the case of information extractors that are learnt unsupervisedly or are based on heuristics.

Basically, we need to compute a mapping that assigns a similarity score in interval $[0.00, 1.00]$ to every pair of attributes (a, e), where a is an attribute in the validation dataset (P, N), i.e., $a \in A_V$, and e is an attribute in the extracted dataset, i.e., $e \in A_E$; generally speaking, the higher the score, the more similar two attributes are and vice versa. To compute this attribute mapping, we first need to find the maximum similarity between a value i of an attribute a from the validation dataset and any of the values of attribute e from the extraction dataset as follows:

$$d(i, e) = \max_{j \in \text{instances}(e,E)} \text{sim}(\tilde{\text{i}}, \tilde{\text{j}}).$$

We then compute the similarity between attributes a and e as follows:

$$m(a, e) = \operatorname{avg}_{i \in \text{instances}(a, P \cup N)} d(i, e).$$

Note that mapping m can be interpreted as a similarity matrix because it provides a similarity score for every pair of attributes in the Cartesian product of A_V and A_E. The cells of this matrix represent the average maximum similarity between the values of a and the values of e. Computing an attribute mapping from m is relatively straightforward, since we only need to select the pairs in $A_V \times A_E$ that maximise m. To prevent producing a mapping for each attribute in cases in which the maximum similarity is very small, we introduce a user-defined threshold θ below which no attribute mappings are accepted. The details of the ancillary procedure to compute the mapping are provided at the end of the section.

5.2. Re-normalise the extracted records

This step consists in changing the names of the attributes in E according to the mapping that we have computed in the previous step. This also requires to re-sort the attribute instances in the records according to the total pre-order used (by default, the lexicographic one). This helps align the annotated records and the extracted records, which facilitates mapping them in the next step. Furthermore, if the validation dataset is not complete, then we must remove every attribute in E for which a mapping has not been found in the previous step. The values of the unmapped attributes from E contribute to increasing the spuriousness of the results if the validation dataset is partially annotated; otherwise, they are counted as false positives. A similar argument follows for the unmapped attributes in $P \cup N$: the unmapped attributes from P contribute to the count of false negatives and the unmapped attributes from N contribute to the count of true negatives.

5.3. Find a record mapping

We rely on the same generic mapping algorithm as before to compute the correspondences between the records in a validation dataset and an extracted dataset. In this case, we compute a similarity matrix m as follows:

$$m(s, t) = \text{sim}(\tilde{s}, \tilde{t}),$$

for any $\tilde{s} \in (P \cup N)$ and $\tilde{t} \in E$. For this definition to work well, it is necessary that the record instances in $P \cup N$ and E be processed by means of the previous steps so that the records are well aligned before their similarity is computed.

Once the similarity matrix m is computed, we can use the same generic method as before to compute the record mapping. The details are provided at the end of the section.

5.4. Compute the confusion matrix

Given a validation dataset (P, N), an extracted dataset E, and a mapping r amongst their records, we first set every component of the confusion matrix to zero and then iterate as follows: (a) for every record value s in the positive dataset that has been mapped onto a record in the extracted dataset, we compute $t = r(s)$, $k = i + d + p$, and $(i, d, p) = \text{diff}(\tilde{s}, \tilde{t})$; we then increase the count of true positives by p/k, i.e., the percentage of tokens that have been correctly extracted, the count of false negatives by d/k, i.e., the percentage of tokens that are in the annotation but have not been extracted, and the count of false positives by i/k, i.e., the percentage of tokens that have been extracted but do not correspond to any tokens in the annotation. (b) For every record value s in the positive dataset that has not been mapped onto a record in the extracted dataset, we increase the count of false negatives by one. (c) For every record value in the negative dataset that has been mapped onto a record value in the extracted dataset, we increase the count of false positives by one. (d) For every record value in the negative dataset that has not been mapped onto any record values in

Table 1
Description of the datasets.

Category	Dataset	Schema	Docs	Size (KiB)	Positives
Jobs	Insight into Diversity	Job{company, location, category}	30	80	30.00
	4 Jobs	Job{company, location, category}	30	80	30.00
	6 Figure Jobs	Job{company, location, category}	30	73	30.00
	Career Builder	Job{company, location, category}	30	54	30.00
	Job of Mine	Job{company, location, category}	30	24	30.00
Cars	Auto Trader	Car{color, doors, engine, mileage, model, price, transmission, type}	30	184	30.00
	Car Max	Car{color, mileage, model, price, transmission, year, type}	30	67	30.00
	Car Zone	Car{color, doors, engine, location, make, model, price, transmission, year, type}	30	71	30.00
	Classic Cars for Sale	Car{color, location, make, model, price, transmission, year, type}	30	76	28.90
	Internet Autoguide	Car{color, doors, engine, location, mileage, price, transmission, type}	30	154	30.00
Books	Abe Books	Book{title, author, price, isbn}	30	38	35.60
	Awesome Books	Book{title, author, price, isbn, year}	30	20	37.17
	Better World Books	Book{title, author, price}	30	125	34.50
	Many Books	Book{title, author, year}	30	27	30.50
	Waterstones	Book{title, author, price}	30	80	31.50

the extracted dataset, we increase the count of true negatives by one. (e) If the validation dataset is complete, we also increase the number of false positives by the count of record values in the extracted dataset that do not correspond to any record values in the positive dataset. From this confusion matrix, one can compute a variety of effectiveness measures (Yu et al., 2016).

5.5. Compute the spuriousness degree

If the annotation of the validation dataset is not complete, we then compute the spuriousness degree as the percentage of record values in the extracted dataset for which there is not a correspondence in the positive dataset.

5.6. Generic method to compute mappings

Given a matrix m that provides the similarity between any two objects in the Cartesian product of two arbitrary sets, the method works as follows: it first finds the pair of objects (p, q) whose similarity is maximum; then, as long as the matrix is not empty and the maximum similarity is not smaller than a user-defined threshold θ, the algorithm maps p onto q, removes that pair from matrix m, and continues iterating. Threshold θ must be set by the user prior to executing our method. It allows to fine-tune how demanding the mapping is: the greater this threshold, the less mappings are computed and vice versa.

Note that our similarity measure returns values in range $[0.00, 1.00]$, which facilitates interpreting θ. Simply put, θ is the one-complement of the percentage of changes that must be carried out in a value to transform it into another value. For instance, setting $\theta = 0.80$ means that the maximum allowable percentage of change to assume that an object can be mapped onto another object is 20%.

6. Experimentation

In this section, we report on the results of our experimental analysis. First, we present the details of our experimental setting and then present and analyse two experiments that help us prove that our guideline to complement ARIEX may lead to results that are significantly different, which proves that the three issues that we have identified are really important.

6.1. Experimental setting

We used a collection of 15 datasets on jobs, cars, and books that were randomly selected from the ARIEX repository (Jiménez et al., 2016). Table 1 shows a description, namely: the columns represent the domains, the sites, the schema of the records and attributes that were annotated, the number of documents collected, their size in KiB, and the average number of positive values annotated. Note that each document provides a form with information about one item, that is, one record must be extracted from each of them. When the average number of positive examples is greater than the number of documents, it means that there are several values for a given attribute. Contrarily, if this number is smaller, it means that there are some missing values for some of the attributes. These datasets were enough to prove that the decisions made regarding the three issues that we have identified have a significant impact on the effectiveness results.

We experimented with four web information extractors, namely: (a) Wien (Kushmerick et al., 1997), which is a classical supervised proposal that learns the delimiters around the information to be extracted; (b) Tango (Jiménez & Corchuelo, 2016a), which is a recent supervised proposal that learns first-order rules whose predicates are based on visual, structural, user-defined, and content-based features; (c) RoadRunner (Crescenzi et al., 2001), which is a classical unsupervised proposal that attempts to infer the template of several documents by comparing their shared and non-shared tokens; and (d) HotWeb (Roldán et al., 2017),[1] which is a heuristic-based proposal that attempts to identify common visual patterns to present information.

We carried out two experiments. The first one confronted Wien and Tango; the goal was to confirm that the way the matchings are computed may have a significant impact on the results. The second one confronted RoadRunner and HotWeb; in this case, the emphasis was on confirming that the degree of spuriousness matters significantly when comparing unsupervised or heuristic-based information extractors. In both cases, we computed the standard effectiveness measures, namely: precision (ratio of true positives to true positives plus false positives), recall (ratio of true positives to true positives plus false negatives), and the F_1 score (the two-harmonic average of precision and recall).

We performed the statistical analyses using the Wilcoxon signed-rank test, which is a non-parametric test to compare two populations. In our case the populations correspond to the results when applying the original validation procedure and the results attained when applying the new validation procedure, regarding each of the effectiveness measures. If the resulting p-value is smaller than the standard significance level ($\alpha = 0.05$), the differences are significant, which demonstrates that the impact of the three issues under study may actually bias the conclusions.

6.2. Experiment #1

First, we experimented with Wien and Tango, which are supervised. The results of this experiment are shown in Table 2. The first two

[1] Roldán et al. (2017) did not use a specific name to refer to their proposal; we have dubbed it HotWeb after the name of the conference where it was presented to facilitate referencing it.

Table 2
Results of Experiment #1.

	Category/Dataset	Wien Original validation			Wien New validation			Tango Original validation			Tango New validation		
		P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Jobs	Insight into Diversity	1.00	0.33	0.50	1.00	0.39	0.56	0.97	0.92	0.94	0.77	0.92	0.84
	4 Jobs	1.00	1.00	1.00	0.95	0.99	0.97	0.97	0.78	0.87	0.93	0.78	0.85
	6 Figure Jobs	1.00	1.00	1.00	0.97	0.99	0.98	0.82	0.93	0.87	0.82	0.93	0.87
	Career Builder	1.00	0.33	0.50	0.02	0.05	0.08	0.93	0.89	0.91	0.79	0.84	0.81
	Job of Mine	1.00	0.67	0.80	1.00	0.63	0.77	0.99	0.96	0.97	0.96	1.00	0.98
Cars	Auto Trader	1.00	1.00	1.00	1.00	1.00	1.00	0.96	0.96	0.96	0.92	0.96	0.94
	Car Max	1.00	0.86	0.92	0.94	0.94	0.94	1.00	1.00	1.00	0.99	0.99	0.99
	Car Zone	1.00	0.87	0.93	0.82	0.69	0.75	0.95	0.97	0.96	0.94	0.97	0.95
	Classic Cars for Sale	-	-	-	-	-	-	0.91	0.97	0.94	0.91	0.97	0.94
	Internet Autoguide	-	-	-	-	-	-	0.92	0.95	0.94	0.89	0.95	0.92
Books	Abe Books	1.00	1.00	1.00	0.99	0.92	0.95	1.00	0.94	0.97	0.78	0.94	0.85
	Awesome Books	1.00	0.67	0.80	1.00	0.67	0.80	0.99	1.00	0.99	0.83	0.99	0.90
	Better World Books	1.00	0.01	0.17	0.01	0.00	0.00	1.00	1.00	1.00	0.66	0.95	0.78
	Many Books	1.00	0.96	0.98	0.99	0.96	0.98	0.97	0.98	0.97	1.00	0.99	0.99
	Waterstones	1.00	0.92	0.96	1.00	0.91	0.96	1.00	1.00	1.00	0.95	1.00	0.97
	Average	1.00	0.74	0.81	0.82	0.70	0.75	0.96	0.95	0.95	0.88	0.95	0.91
	Standard deviation	0.00	0.32	0.26	0.36	0.35	0.34	0.05	0.06	0.04	0.10	0.06	0.07

columns report on the category and the datasets, and the remaining ones report on the effectiveness measures computed for each proposal on those datasets in the context of the original validation and the new validation performed in this experiment. The cells with a dash indicate that the proposal in the corresponding column was unable to extract information from the dataset in the corresponding row. The last two rows report on the average and the standard deviation of the measures.

Regarding the completeness of the validation datasets, it was 100% in this case because the datasets were completely annotated with the exact information to be extracted. That means that every piece of information extracted that was not mapped was counted as a false positive in the confusion matrix. Regarding how the information is structured, the annotated datasets have one level of nesting: first the record values were annotated and then the attribute values were annotated within them. Thus, we computed the results as an average of their attribute and records values. If a record was not extracted or incorrectly extracted, their corresponding attribute values were assumed not to be extracted and their annotations were therefore counted as false negatives. Regarding how matches were computed, the original validations used contains matchings; the new validation used exact matchings, which are more stringent and appropriate for supervised proposals since they are intended to learn the exact pieces of information to extract from the annotations. The user-defined threshold θ to find the attribute or record mappings was set to 80% in both cases.

The results in Table 2 make it clear that there are differences in the effectiveness between the original and the new evaluation. The results are worse in the new validation, which makes it clear that using exact matchings is more stringent. It has a clear negative impact on the precision and the F_1 score, even though the recall is similar. In both proposals, there are three frequent situations, namely: the precision gets worse, which occurs when there are more extracted tokens than annotated ones because they count as false positives; the recall gets better, which happens when the number of tokens that have not been extracted is a small fraction of the total number of tokens in the record so that they count as fractional numbers instead of whole numbers; the recall gets worse, which is the opposite case.

Now, we need to prove that the differences are actually significant at the standard significance level. The results of the statistical analysis are shown in Table 3. The first two columns refer to the effectiveness measures and whether they correspond to the original or the new validation; the following columns report on their empirical ranks, minimum and maximum values, average value and standard deviation, and the p-value computed by the Wilcoxon signed-rank test. Note that all of the p-values are clearly smaller than the standard significance level, which is a strong indication that the differences between the results in the original validation and the new validation are significant.

6.3. Experiment #2

Second, we experimented with RoadRunner, which is unsupervised, and HotWeb, which is heuristic-based. Regarding RoadRunner, we learnt a template from two random documents and then applied it to the remaining ones. We repeated the process with random pairs of learning documents and we selected the best performing one. We did not use more learning documents because their variability makes it very difficult to find a common template as the number of learning documents increases and the proposal either found meaningless templates or did not stop in a sensible time. Regarding HotWeb, it does not learn any rules, but has built-in heuristics that are directly applied to the input documents. Thus, all of the documents in the datasets were used for validation. Although the datasets are the same as in the previous experiment, they were partially annotated in this context because the proposals do not learn to extract the annotations, but extract as much information as possible building on the variability that they discover in the input documents. Consequently, they typically extract much more information than expected. The user-defined threshold θ to find the attribute mappings was set to 80% in the case of the HotWeb proposal and 20% in the case of RoadRunner. The reason is that RoadRunner extracted values that were typically much larger than the annotated values; thus, their similarity dropped significantly.

The results of this experiment are shown in Table 4, which has an additional column called SD that reports on the degree of spuriousness. (The figures were rounded up to two decimals, which means that the cells with a 1.00 actually represent a figure in between 0.995 and 1.000.) The conclusion is that either the datasets are not sufficiently annotated, which is something common in the context of unsupervised or heuristic-based proposals, or that the proposals extract much irrelevant information in which we are not interested. Ours is the former case: only a few attributes per document were annotated, but all of their values were annotated. Note that the interpretation of the results might be misleading when the datasets are partially annotated since a proposal that reaches a precision and a recall close to 1.00 with a spuriousness degree close to 1.00 might be worse than another proposal that reaches a precision and a recall between 0.80 and 0.90 with a spuriousness degree close to 0.30. The spuriousness degree is computed as the average of spurious information in every record within the validation datasets.

According to the results, RoadRunner performs very poorly in our validation datasets because it frequently extracted very long excerpts that contain the attribute values and much irrelevant information; in the new validation, this contributed to worsening the effectiveness measures, which makes the problem more evident. It also tends to

Table 3
Statistical analysis of Experiment #1.

Measure	Validation	Rank	Min	Max	Mean	Stdev	P-value
P	Original	1.19	1.00	1.00	1.00	0.00	3.96E-03
	New	1.81	0.01	1.00	0.82	0.36	
R	Original	1.27	0.01	1.00	0.74	0.32	5.40E-02
	New	1.73	0.00	1.00	0.70	0.35	
F1	Original	1.31	0.17	1.00	0.81	0.26	3.74E-02
	New	1.69	0.00	1.00	0.75	0.34	

(a) Results regarding Wien.

Measure	Validation	Rank	Min	Max	Mean	Stdev	P-value
P	Original	1.07	0.82	1.00	0.96	0.05	9.83E-04
	New	1.93	0.66	1.00	0.88	0.10	
R	Original	1.57	0.78	1.00	0.95	0.06	4.44E-01
	New	1.43	0.78	1.00	0.95	0.06	
P	Original	1.20	0.87	1.00	0.95	0.04	5.30E-03
	New	1.80	0.78	0.99	0.91	0.07	

(b) Results regarding Tango.

Table 4
Results of Experiment #2.

	Dataset	RoadRunner Original validation			RoadRunner New validation					HotWeb Original validation			HotWeb New validation				
		P	*R*	F_1	*P*	*R*	F_1	*S*	*SD*	*P*	*R*	F_1	*P*	*R*	F_1	*S*	*SD*
Jobs	Insight into Diversity	1.00	0.08	0.15	0.19	0.02	0.04	0.14	0.38	0.93	0.50	0.65	0.93	0.50	0.65	0.97	0.03
	4 Jobs	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.98	0.82	0.90	0.98	0.82	0.90	1.00	0.00
	6 Figure Jobs	1.00	0.29	0.44	0.43	0.17	0.24	0.46	0.50	0.82	0.96	0.88	0.82	0.96	0.88	0.99	0.00
	Career Builder	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.99	0.16	0.27	0.99	0.16	0.27	1.00	0.00
	Job of Mine	1.00	0.46	0.63	0.88	0.47	0.61	0.70	0.46	1.00	1.00	1.00	1.00	1.00	1.00	0.01	0.00
Cars	Auto Trader	-	-	-	-	-	-	-	-	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.00
	Car Max	0.82	0.47	0.60	0.82	0.40	0.54	0.77	0.42	0.96	0.63	0.76	0.96	0.63	0.76	0.99	0.00
	Car Zone	1.00	0.01	0.02	0.00	0.00	0.00	0.00	0.00	0.88	0.92	0.90	0.88	0.92	0.90	0.99	0.00
	Classic Cars for Sale	0.84	0.11	0.20	0.87	0.14	0.25	0.82	0.39	0.90	0.86	0.88	0.90	0.86	0.88	0.99	0.00
	Internet Autoguide	1.00	0.43	0.60	0.10	0.02	0.03	0.38	0.49	0.99	0.99	0.99	0.99	0.99	0.99	1.00	0.00
Books	Abe Books	1.00	0.41	0.58	0.94	0.47	0.63	0.81	0.39	0.98	0.92	0.95	0.98	0.92	0.95	0.97	0.01
	Awesome Books	1.00	0.80	0.89	0.97	0.70	0.81	0.68	0.47	1.00	0.65	0.79	1.00	0.65	0.79	0.95	0.03
	Better World Books	-	-	-	-	-	-	-	-	1.00	0.91	0.95	1.00	0.91	0.95	0.99	0.00
	Many Books	0.58	0.37	0.45	0.73	0.22	0.34	0.86	0.35	1.00	0.38	0.56	1.00	0.38	0.56	1.00	0.00
	Waterstones	1.00	0.33	0.50	0.30	0.13	0.18	0.83	0.37	1.00	0.89	0.94	1.00	0.89	0.94	0.99	0.00
	Average	0.79	0.29	0.39	0.48	0.21	0.28	0.50	0.32	0.96	0.77	0.83	0.96	0.77	0.83	0.92	0.00
	Standard deviation	0.37	0.24	0.29	0.40	0.23	0.28	0.35	0.19	0.06	0.26	0.20	0.06	0.26	0.20	0.25	0.01

Table 5
Statistical analysis of Experiment #2.

Measure	Validation	Rank	Min	Max	Mean	Stdev	P-value
P	Original	1.27	0.00	1.00	0.79	0.37	1.80E-02
	New	1.73	0.00	0.97	0.48	0.40	
R	Original	1.31	0.00	0.80	0.29	0.24	1.96E-02
	New	1.69	0.00	0.70	0.21	0.23	
F1	Original	1.23	0.00	0.89	0.39	0.29	1.05E-02
	New	1.77	0.00	0.81	0.28	0.28	

(a) Results regarding RoadRunner.

Measure	Validation	Rank	Min	Max	Mean	Stdev	P-value
P	Original	1.50	0.82	1.00	0.96	0.06	5.800E-01
	New	1.50	0.82	1.00	0.96	0.06	
R	Original	1.50	0.16	1.00	0.77	0.26	5.00E-01
	New	1.50	0.16	1.00	0.77	0.26	
*F*1	Original	1.50	0.27	1.00	0.83	0.20	5.00E-01
	New	1.50	0.27	1.00	0.83	0.20	

(b) Results regarding HotWeb.

extract several attributes as one single attribute that is not easy to split using post-processing. Very often, too, this proposal extracts all of the attributes in a few documents, but fails with the others because it cannot infer a good common template, which is the reason why its recall seldom exceeds 0.40–0.50. It seems that HotWeb provides very reliable results because its effectiveness measures keep almost the same in the original validation and the new validation. The authors originally computed the effectiveness measures at the attribute level using exact matchings, but they did not analyse the amount of extractions that cannot be mapped onto annotations.

We have also conducted a statistical analysis. The results are shown in Table 5. Note that the *p*-value returned by the Wilcoxon signed-rank test is below the standard significance level in the case of RoadRunner, which clearly supports the idea that the differences in rank between the original and the new validation are statistically significant. Note, too, that the p-values coincide with the standard significance level in

the case of HotWeb, which means that the experiment does not support the hypothesis that the differences in rank are significant regarding this proposal. The reason is, basically, that the original validation was as stringent as the new one.

7. Conclusions

In this article, we have identified three key issues regarding the validation of information extractors, namely: (a) completeness of the validation datasets, which is of uttermost importance to put the effectiveness measures in a proper context regarding the degree of spuriousness; (b) the structure of the information to be validated, so that one can know if the proposal is able to extract flat structures or nested structures, and how good it is in the latter case; (c) the kind of matching selected, which makes the conclusions more or less stringent.

The previous issues have a significant impact on the effectiveness measures, which means that the comparisons might be heterogeneous and unfair if we do not report on the previous issues. We have performed two experiments regarding the previous ideas. In the first experiment, we worked with two supervised proposals; the results proved that precision decreased when our proposal was used, which had a negative impact on the F_1 score. In the second experiment, we worked with an unsupervised and a heuristic-based proposal; our empirical results proved that the degree of spuriousness can be significant and must be reported. Our statistical analyses confirmed our ideas.

Summing up, we strongly recommend that researchers should conduct the evaluation of their proposal following the guideline provided so that the results they publish are easier to compare fairly. They also have to carefully compare their proposals to their competitors, as long as the competitors have also provided the results following the same guideline.

CRediT authorship contribution statement

Patricia Jiménez: Conceptualisation, Methodology, Software, Validation, Resources, Data curation, Writing – original draft, Writing – review & editing. **Rafael Corchuelo:** Conceptualisation, Methodology, Writing – original draft, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors were partially supported by the Spanish R&D programme through grants TIN2016-75394-R and PID2020-112540RB-C44 (MCIN/AEI/ 10.13039/501100011033), as well as the Andalusian R&D programme through grants P18-RT-1060 and US-1381375.

References

Álvarez, M., Pan, A., Raposo, J., Bellas, F., & Cacheda, F. (2010). Finding and extracting data records from web pages. *Signal Processing Systems*, *59*(1), 123–137. http://dx.doi.org/10.1007/s11265-008-0270-y.

Baumgartner, R., Frölich, O., & Gottlob, G. (2007). The Lixto systems applications in business intelligence and the Semantic Web. In *ESWC* (pp. 16–26). http://dx.doi.org/10.1007/978-3-540-72667-8_3.

Baumgartner, R., Gatterbauer, W., & Gottlob, G. (2018). Web data extraction system. In *Encyclopedia of database systems* (2nd ed.). Springer, http://dx.doi.org/10.1007/978-1-4614-8265-9_1154.

Chang, C.-H., Kayed, M., Girgis, M. R., & Shaalan, K. F. (2006). A survey of web information extraction systems. *IEEE Transactions on Knowledge and Data Engineering*, *18*(10), 1411–1428. http://dx.doi.org/10.1109/TKDE.2006.152.

Chinchor, N., Hirschman, L., & Lewis, D. D. (1993). Evaluating message understanding systems: an analysis of the third Message Understanding Conference (MUC-3). *Computational Linguistics*, *19*(3), 409–449.

Crescenzi, V., Mecca, G., & Merialdo, P. (2001). RoadRunner: towards automatic data extraction from large web sites. In *VLDB* (pp. 109–118). URL http://www.vldb.org/conf/2001/P109.pdf.

Ferrara, E., Meo, P. D., Fiumara, G., & Baumgartner, R. (2014). Web data extraction, applications and techniques: a survey. *Knowledge Based System*, *70*, 301–323. http://dx.doi.org/10.1016/j.knosys.2014.07.007.

Ferri, C., Hernández-Orallo, J., & Modroiu, R. (2009). An experimental comparison of performance measures for classification. *Pattern Recognition Letters*, *30*(1), 27–38. http://dx.doi.org/10.1016/j.patrec.2008.08.010.

Freitag, D. (2000). Machine learning for information extraction in informal domains. *Machine Learning*, *39*(2/3), 169–202. http://dx.doi.org/10.1023/A:1007601113994.

Hirschman, L. (1998). The evolution of evaluation: lessons from the Message Understanding Conferences. *Computer Speech and Language*, *12*(4), 281–305. http://dx.doi.org/10.1006/csla.1998.0102.

Hogue, A. W., & Karger, D. R. (2005). Thresher: automating the unwrapping of semantic content from the World Wide Web. In *WWW* (pp. 86–95). http://dx.doi.org/10.1145/1060745.1060762.

Hsu, C.-N., & Dung, M.-T. (1998). Generating finite-state transducers for semi-structured data extraction from the Web. *Information Systems*, *23*(8), 521–538. http://dx.doi.org/10.1016/S0306-4379(98)00027-1.

Hunt, J. W., & McIlroy, M. D. (1976). *Computing Science Technical Report, An algorithm for differential file comparison: Technical report*, Bell Laboratories, URL http://www.cs.dartmouth.edu/~doug/diff.pdf.

Ireson, N., Ciravegna, F., Califf, M. E., Freitag, D., Kushmerick, N., & Lavelli, A. (2005). Evaluating machine learning for information extraction. In *ICML, vol. 119* (pp. 345–352). http://dx.doi.org/10.1145/1102351.1102395.

Irmak, U., & Suel, T. (2006). Interactive wrapper generation with minimal user effort. In *WWW* (pp. 553–563). http://dx.doi.org/10.1145/1135777.1135859.

Jiménez, P., & Corchuelo, R. (2016a). On learning web information extraction rules with Tango. *Information Systems*, *62*, 74–103. http://dx.doi.org/10.1016/j.is.2016.05.003.

Jiménez, P., & Corchuelo, R. (2016b). Roller: a novel approach to web information extraction. *Knowledge and Information Systems*, *49*(1), 197–241. http://dx.doi.org/10.1007/s10115-016-0921-4.

Jiménez, P., Corchuelo, R., & Sleiman, H. A. (2016). ARIEX: automated ranking of information extractors. *Knowledge Based System*, *93*, 84–108. http://dx.doi.org/10.1016/j.knosys.2015.11.004.

Jiménez, P., Roldán, J. C., & Corchuelo, R. (2021). A clustering approach to extract data from HTML tables. *Information Processing & Management*, *58*(6), Article 102683. http://dx.doi.org/10.1016/j.ipm.2021.102683.

Jiménez, P., Roldán, J. C., Gallego, F. O., & Corchuelo, R. (2020). On the synthesis of metadata tags for HTML files. *Software - Practice and Experience*, *50*(12), 2169–2192. http://dx.doi.org/10.1002/spe.2886.

Kayed, M., & Chang, C.-H. (2010). FiVaTech: page-level web data extraction from template pages. *IEEE Transactions on Knowledge and Data Engineering*, *22*(2), 249–263. http://dx.doi.org/10.1109/TKDE.2009.82.

Kushmerick, N., Weld, D. S., & Doorenbos, R. B. (1997). Wrapper induction for information extraction. In *IJCAI (1)* (pp. 729–737).

Lavelli, A., Califf, M. E., Ciravegna, F., Freitag, D., Giuliano, C., Kushmerick, N., & Romano, L. (2004). A critical survey of the methodology for IE evaluation. In *LREC*. URL http://www.lrec-conf.org/proceedings/lrec2004/summaries/416.htm.

Lavelli, A., Califf, M. E., Ciravegna, F., Freitag, D., Giuliano, C., Kushmerick, N., Romano, L., & Ireson, N. (2008). Evaluation of machine learning-based information extraction algorithms: criticisms and recommendations. *Langage Resource and Evaluation*, *42*(4), 361–393. http://dx.doi.org/10.1007/s10579-008-9079-3.

Lehnert, W. G., & Sundheim, B. (1991). A performance evaluation of text-analysis technologies. *AI Magazine*, *12*(3), 81–94. http://dx.doi.org/10.1609/aimag.v12i3.905.

Park, J., & Barbosa, D. (2007). Adaptive record extraction from web pages. In *WWW* (pp. 1335–1336). http://dx.doi.org/10.1145/1242572.1242838.

Raposo, J., Pan, A., Álvarez, M., Hidalgo, J., & na, A. V. (2002). The Wargo system: semi-automatic wrapper generation in presence of complex data access modes. In *DEXA Workshops* (pp. 313–320). http://dx.doi.org/10.1109/DEXA.2002.1045916.

Roldán, J. C., Jiménez, P., & Corchuelo, R. (2017). Extracting web information using representation patterns. In *HotWeb* (pp. 4:1–4:5). http://dx.doi.org/10.1145/3132465.3133840.

Roldán, J. C., Jiménez, P., & Corchuelo, R. (2020). On extracting data from tables that are encoded using HTML. *Knowledge Based System*, *190*, Article 105157. http://dx.doi.org/10.1016/j.knosys.2019.105157.

Roldán, J. C., Jiménez, P., Szekely, P., & Corchuelo, R. (2021). TOMATE: a heuristic-based approach to extract data from HTML tables. *Information Sciences*, *577*, 49–68. http://dx.doi.org/10.1016/j.ins.2021.04.087.

Sahuguet, A., & Azavant, F. (2001). Building intelligent web applications using lightweight wrappers. *Data & Knowledge Engineering*, *36*(3), 283–316. http://dx.doi.org/10.1016/S0169-023X(00)00051-3.

Shen, Y. K., & Karger, D. R. (2007). U-REST: an unsupervised record extraction system. In *WWW* (pp. 1347–1348). http://dx.doi.org/10.1145/1242572.1242844.

de Sitter, A., & Daelemans, W. (2003). Information extraction via double classification. In *ATEM Workshop (ECML/PKDD)* (pp. 1–8).

Sleiman, H. A., & Corchuelo, R. (2013). A survey on region extractors from web documents. *IEEE Transactions on Knowledge and Data Engineering*, *25*(9), 1960–1981. http://dx.doi.org/10.1109/TKDE.2012.135.

Sleiman, H. A., & Corchuelo, R. (2014). Trinity: on using trinary trees for unsupervised web data extraction. *IEEE Transactions on Knowledge and Data Engineering*, *26*(6), 1544–1556. http://dx.doi.org/10.1109/TKDE.2013.161.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427–437. http://dx.doi.org/10.1016/j.ipm.2009.03.002.

Turmo, J., Ageno, A., & Català, N. (2006). Adaptive information extraction. *ACM Computing Surveys*, *38*(2), http://dx.doi.org/10.1145/1132956.1132957.

Yu, M., Li, G., Deng, D., & Feng, J. (2016). String similarity search and join: a survey. *Frontiers of Computer Science*, *10*(3), 399–417. http://dx.doi.org/10.1007/s11704-015-5900-5.

Zhai, Y., & Liu, B. (2005). Web data extraction based on partial tree alignment. In *WWW* (pp. 76–85). http://dx.doi.org/10.1145/1060745.1060761.

Zhang, C., Xu, W., Ma, Z., Gao, S., Li, Q., & Guo, J. (2015). Construction of semantic bootstrapping models for relation extraction. *Knowledge Based System*, *83*, 128–137. http://dx.doi.org/10.1016/j.knosys.2015.03.017.