A Method to Evaluate and Compare Web Information Extractors

Patricia Jiménez, Rafael Corchuelo, and Hassan A. Sleiman

Abstract—Web mining is gaining importance at an increasing pace. Currently, there are many complementary research topics under this umbrella. Their common theme is that they all focus on applying knowledge discovery techniques to data that is gathered from the Web. Sometimes, these data are relatively easy to gather, chiefly when it comes from server logs. Unfortunately, there are cases in which the data to be mined is the data that is displayed on a web document. In such cases, it is necessary to apply a pre-processing step to first extract the information of interest from the web documents. Such pre-processing steps are performed using so-called information extractors, which are software components that are typically configured by means of rules that are tailored to extracting the information of interest from a web page and structuring it according to a pre-defined schema. Paramount to getting good mining results is that the technique used to extract the source information is exact, which requires to evaluate and compare the different proposals in the literature from an empirical point of view. According to Google Scholar, about 4 200 papers on information extraction have been published during the last decade. Unfortunately, they were not evaluated within a homogeneous framework, which leads to difficulties to compare them empirically. In this paper, we report on an original information extraction evaluation method. Our contribution is three-fold: a) this is the first attempt to provide an evaluation method for proposals that work on semi-structured documents; the little existing work on this topic focuses on proposals that work on free text, which has little to do with extracting information from semi-structured documents. b) It provides a method that relies on statistically-sound tests to support the conclusions drawn; the previous work does not provide clear guidelines or recommend statistically-sound tests, but rather a survey that collects many features to take into account as well as related work; c) We provide a novel method to compute the performance measures regarding unsupervised proposals; otherwise they would require the intervention of a user to compute them by using the annotations on the evaluation sets and the information extracted. Our contributions will definitely help researchers in this area make sure that they have advanced the state of the art not only conceptually, but from an empirical point of view; it will also help practitioners make informed decisions on which proposal is the most adequate for a particular problem. This conference is a good forum to discuss on our ideas so that we can spread them to help improve the evaluation of information extraction proposals and gather valuable feedback from the researchers in this field.

Keywords—Information extraction, web mining, evaluation methods, non-parametric statistical tests.

P. Jiménez and R. Corchuelo are with the ETSI Informática of the University of Sevilla, Avda. Reina Mercedes, s/n, Sevilla E-41012, Spain. e-mail: {patriciajimenez, corchu}@us.es

H. Sleiman with the CEA, LIST Institute, Gif-sur-Yvette 91191 CEDEX, France. e-mail: hassan.sleiman@cea.fr

This 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).

I. Introduction

Nowadays the World Wide Web is the largest resource of information of the Humanity, and it is still growing and evolving. Never before have companies been more interested in the information it provides, since mining it may result in better efficiencies and more business opportunities. Web Mining is the research field that provides the techniques required to analyse the data that originates from the Web and to produce knowledge out of them. Web mining is commonly structured into three sub-fields, namely: web usage mining, which focuses on analysing server logs to discover user interaction patterns, web structure mining, which focuses on analysing the node and connection structure of a web site, and web content mining, which focuses on analysing the data provided by a web site.

In this paper, we focus on web content mining. These techniques are somewhat complicated by the fact that the data to be mined needs first be retrieved from web documents that are typically crawled from a web site and indexed building on their key words [26]; then, the pages need to be analysed so that the data that they provide can be extracted in a structured format that is amenable for computer processing; note that this is not usually a trivial task because typical web documents do not focus on how the data is structured, but on how it must be rendered using HTML tags, CSS classes, and JS scripts. (Typical web documents are commonly referred to as semi-structured documents.) In the literature, there are many techniques that help create information extractors, which are software components that are typically configured by means of rules that are specifically tailored to extracting structured information from a web site [4, 19–21, 23, 28, 37].

Web information extraction has been quite an active research field during the last decade. For instance, as of the time of writing this paper, Google Scholar reports on roughly 4 200 papers on this topic. They can be classified according to the kind of document on which they work, namely: free-text, e.g., newspapers, semi-structured, e.g., HTML-rich documents, and structured, e.g., XML documents. Our focus is on semi-structured documents, which are characterised by a common template that is filled out on the fly with user-requested information retrieved from a database, and displayed in a human-friendly format using HTML. The proposals in this field can be further classified according to the degree of user involvement as supervised, which require the user to provide an annotated training set from which extraction rules are

machine learnt, and unsupervised, which can either learn the extraction rules from an non-annotated training set or extract as much information as possible. The latter require the user to interpret the results.

The authors of new proposals that work on semi-structured documents are usually interested in evaluating them and wish to compare their performance to other proposals, where performance refers to both effectiveness and efficiency. Practitioners of web content mining need make informed decisions regarding which the most adequate information extraction proposal is regarding a particular problem. Unfortunately, performing such evaluations within a homogeneous framework is not easy at all, because of the many aspects that must be made explicit so that the comparisons are fair and statistically sound.

Unfortunately, the little existing work in this field focuses on proposals that work on free text, which has little to do with extracting information from semi-structured documents. Lavelli et al. [22] presented the most recent method, but their focus was on issues that are not applicable to our context, namely: a) defining the extraction task precisely (entity recognition, relation recognition, or event recognition, to mention a few); in our context, there is only one extraction task that consists in learning to extract the pieces of text in which a user is interested. b) How to collect effectiveness measures, which is not trivial because the information extracted can be classified as correct, partially correct, incorrect, missing, or spurious; furthermore, multiple instances can be extracted for the same attribute, which complicates everything. In our context, an information extractor can be seen as a binary classifier that either puts a piece of text or a DOM node into a user-defined class (e.g., Book or title) or in a null class; that is, the effectiveness measures can be easily defined in terms of the components of a typical confusion matrix. However, a few guidelines can be borrowed from this evaluation method, namely, collecting datasets, splitting them, and reporting on the results providing enough statistical evidence to support the conclusions drawn. Regarding the first two, they support the idea that the proposals must be compared on the same datasets and splits, which is also applicable to our context. Regarding the last one, they do not propose a method to analyse or present the results, but rather suggest that it is necessary and they provide some references to classical work that used statistical inference. They also provide much background on the evaluations performed at the MUC conferences; Hirschman [14] provided complementary information regarding such evaluations. In the literature, there are some tools that were developed to evaluate free text information extraction proposals [8, 9, 12, 27], but their focus is on computing the matchings and the performance measures; none of them provides a clear guideline or a statistically-sound method to carry out a fair and reliable comparison. An additional problem with the previous proposals is that they focus exclusively on supervised proposals, which leaves the many unsupervised existing proposals out.

This paper aims to provide an information extraction evaluation method that has solid foundations to carry out fair side-by-side comparisons in the context of web information extraction from semi-structured documents. It is novel in that it is the first attempt to provide such a method, it is rigorous and statistically sound, and we take into account both supervised and unsupervised proposals. We expect that our method can help practitioners in the field of web content mining make informed decisions regarding which the most adequate proposal to proposal to extract information from a web site is.

The rest of this paper is organised as follows: Section II describes some repositories and datasets used on information extraction proposals during the last decade; Section III presents the common approaches to split the datasets; Section IV describes the different measures that can be collected to compare the performance of several proposals; Section V reports on how to present the results, draw the conclusions, and support them by statistically sound methods; Section VI presents our method to compare information extraction proposals and Section VII illustrates it; Section VIII concludes our work.

II. Repositories

Many authors have evaluated their information extraction proposals on private repositories or they have used different repositories, which makes it difficult to compare them fairly. Thus, it is important that the authors of new proposals provide a full description of the repositories used, which includes where they were taken from, the datasets involved (including the version number if applicable), and a description of the attributes to extract.

Below, we briefly describe some of the repositories available:

RISE: This is the Repository of on-line Information Sources used in information Extraction tasks [25]. It provides datasets used to evaluate information extraction proposals for both semi-structured and free-text documents. The datasets were collected from 10 different web sites, and each of them provides from 9 to 255 documents.

TBDW: This is the Test Bed for information extraction from Deep Web [38]. It provides a collection of datasets from 51 web sites and each dataset provides five semi-structured documents. It only includes the annotations of the first few data records in each document. The annotations in TBDW are included in a separate document, but they do not have an explicit model.

TIPSTER: The TIPSTER project [36] includes a repository of documents [13] that range from documents from the Wall Street Journal to the USA Federal Register. The annotations in TIPSTER are formatted using SGML-like tags in separate documents.

Other repositories: Some authors have produced repositories of their own [1, 2, 7, 18, 31, 35]. For instance, Sleiman and Corchuelo [31] published the most up-to-date repository. It provides 24 real-word datasets on books, cars, conferences, doctors, jobs, movies, real estates, and sports. These categories were randomly sampled from

The Open Directory sub-categories, and the web sites inside each category were randomly selected from the 100 best ranked web sites between December 2010 and March 2011 according to Google's search engine. Each dataset provides a total of 30 semi-structured documents and includes a complete set of annotations.

III. Partitioning datasets

To evaluate a proposal, the selected repositories must be partitioned into two parts each, namely: a training set, that must be used to learn the extraction rules, and an evaluation partition, that must be used to compute the performance of the rules learnt. Heuristic-based proposals do not learn extraction rules; thus, they do not need the datasets to be partitioned.

There exist the following procedures to split the datasets:

***N*-repeated random partitions:** It partitions the datasets randomly into training and evaluation partitions. The procedure is repeated *N* times using different randomly selected train/evaluation partitions.

***k*-fold cross validation:** It partitions the datasets into *k* subsets of documents and then iterates over them. At each iteration, it considers one of these partitions for evaluation, whereas the remaining ones are considered as a unique training set. The rules learnt at each iteration are evaluated on the selected evaluation partition.

The decision on what procedure to use is deferred to the experimenters. However, to perform a reliable comparison, every proposal to be compared must be evaluated on the same training/evaluation splits. This is the reason why the repositories used to compare the different proposals must be published as well as the partitions, so that other researchers can use them to compare their proposals.

IV. Performance measures

Below, we report on the two kinds of performance measures available, namely: effectiveness and efficiency measures.

A. Effectiveness measures

These measures aim to characterise how well a proposal works in terms of its ability to extract relevant information and discard spurious information.

In information extraction, each type of information (user-defined class) to be extracted from a document amounts to a single binary classification task. In other words, we may see an information extractor as a text classifier that puts every text fragment or DOM node in the input documents in a user-defined class; the information that is not extracted is usually classified in a pre-defined class to which we refer to as null. Thus, effectiveness measures can be computed on the basis of the components of a typical confusion matrix, namely: true positives (*tp*), false positives (*fp*), true negatives (*tn*), and false negatives (*fn*).

The most common effectiveness measures regarding a class are the following:

Precision: It refers to the ratio of true positives of a class to the total amount of information returned by an information extractor as belonging to that class. Intuitively, the higher the precision, the less incorrect information is extracted as belonging to a given class.
This measure is formally defined as follows:

$$P = \frac{tp}{tp + fp}.$$

Recall: It refers to the ratio of true positives of a class to the total amount of information that actually belongs to that class. Intuitively, the higher the recall, the more correct information is extracted as belonging to a class.
This measure is formally defined as follows:

$$R = \frac{tp}{tp + fn}.$$

F_β measure: Note that neither a high precision nor a high recall indicates that an information extractor is good. For instance, an information extractor that achieves perfect precision might be the worst information extractor in the world, e.g., an information extractor that does not extract any information at all has perfect precision since it does not make any mistakes; similarly, an information extractor that achieves perfect recall might not be useful at all, e.g., an information extractor that returns every piece of information as belonging to a given class has perfect recall with respect to that class. The F_β measure is a weighted combination of both precision and recall in a β-harmonic mean that is close to 1.00 when both precision and recall are high, and close to 0.00 when any of them is not good enough. Usually, β is set to 1, which results in the standard harmonic mean of precision and recall.
This measure is formally defined as follows:

$$F_\beta = (1 + \beta^2)\frac{P\,R}{(\beta^2\,P) + R}.$$

Note that previous measures are defined at the class level. To compute these measures at the information extractor level we must compute the so-called weighted average for each measure, i.e., the sum of the product of each measure times its number of occurrences divided by the total number of occurrences.

Assume that we are working with n classes, that P_i and R_i ($1 \leq i \leq n$) denote, respectively, the precision and recall of an information extractor on those classes, and that it returns m_i ($1 \leq i \leq n$) pieces of information in each class; we then define the information-extractor-level precision and recall as follows:

$$P = \frac{\sum_{i=1}^{n} m_i P_i}{\sum_{i=1}^{n} m_i},$$

$$R = \frac{\sum_{i=1}^{n} m_i R_i}{\sum_{i=1}^{n} m_i}.$$

The information-extractor-level F_β can be computed as usual, using the β-harmonic mean of the information-extractor-level precision and recall.

In the previous paragraphs, we have assumed that the classes that an information extractor returns are meaningful. This is true in the case of supervised proposals since they are trained to extract information of a number of user-defined classes. In the case of unsupervised proposals, the classes are not meaningful since they are computer-generated; it is the responsibility of the user to interpret them and assign a meaning to them; in the case of heuristic-based information extractors, there are not any explicit computer-generated classes, but groups of information that are extracted together. This makes computing the effectiveness measures of an unsupervised proposal a little more difficult since prior to computing them, we need map each computer-generated class onto a user-defined class in the evaluation set. An effective solution to this problem is to compute the measures on every possible mapping and select the ones with the highest F_1 measure. Our experience proves that this is quite an effective approach that saves much user effort to evaluate both unsupervised rule-based proposals and heuristic-based proposals [32, 33].

B. Efficiency measures

These measures are related to the amount of resources that a proposal consumes. In theory, the best possible evaluation consists in analysing the theoretical complexity to learn extraction rules and to apply them, or to extract information directly if a proposal is based on heuristics. The analysis should be performed both regarding time complexity and space complexity, which refer to the theoretical minimum upper limit to the number of elementary operations or memory cells that an algorithm requires in terms of the size of the input. Unfortunately, information extraction proposals are far too difficult to analyse since their complexities depend on many variables that are very difficult to characterise. In such cases, it generally suffices to compute an upper bound that proves that the algorithm is computationaly tractable, i.e., is not exponential or worse in the size of the input; unfortunately, these upper bounds are not appropriate to compare them side-by-side.

Thus, our conclusion is that in spite of the fact that timings are related to a particular implementation run on a particular computer and sensitive to environmental conditions, they are a must to evaluate a proposal and compare it to others.

Prior to evaluating an information extractor in terms of efficiency, it is important to describe the experimentation environment: processor features (model, number of cores, and clock speed), RAM capacity, operating system, tools used, and their configuration.

Typical efficiency measures include the CPU time, which is the actual time the CPU is allocated, the IO time, which is the time the IO devices are allocated to reading or writing data, and total time, or simply time, which is the total amount of time that elapses since an algorithm starts running until it finishes (this includes the CPU time, the IO time, and the time the algorithm waits for the CPU or the IO devices to be allocated). CPU and IO times are quite stable, i.e., when an algorithm is repeatedly executed on the same input they do not vary largely; contrarily, total times are not so stable because they depend on many other processes that can run concurrently on the same machine. As a conclusion, to measure accurate total times it is a good idea to repeat the experiments a sufficiently large number of times, typically 25 times, and to average the results after analysing outliers using, for instance, the well-known Cantelli inequality or other more sophisticated methods [15], and discarding those that are due to environmental causes.

V. Reporting on the results

Once the experiments are run and the performance measures are gathered, it is time to provide a discussion regarding the conclusions that can be drawn from the results. Ideally, a new proposal should outperform others in the literature regarding at least effectiveness or efficiency. Then, the conclusions drawn must be supported by statistically-sound methods.

For instance, assume that the CPU learning times of proposal A are 4.50, 4.60, 4.80, 4.30, 4.10, and 2.70 seconds, i.e., the mean is 4.17 seconds and the standard deviation is 0.69 seconds; assume now that the CPU learning times of proposal B are 4.30, 4.20, 4.40, 4.50, 4.30, and 4.10 seconds, i.e., the mean is 4.31 seconds and the standard deviation is 0.12 seconds. That is, proposal A seems to perform better than proposal B, but note that is was actually the last experiment that lowered its mean time. In other words, we need discern if the difference in performance is actually an intrinsic difference between these proposals or if they are just a consequence of the random factors that underlie the evaluation: the selection of the datasets, how they were partitioned, the documents that were selected, and so on.

The previous question can be addressed using statistical tests, which basically provide a procedure to analyse the following hypothesis:

H_0 **(Null hypothesis):** There are no statistically significant differences in the behaviour of a number of proposals regarding a given performance measure.

H_1 **(Alternate hypothesis):** There are statistically significant differences in the behaviour of a number of proposals regarding a given performance measure.

In order to prove that a proposal outperforms the state of the art, the authors must gather enough evidence to reject the null hypothesis at a given confidence level, which is denoted as α and typically set to 0.05, i.e., 95% confidence.

Statistical tests build on computing a statistic from a sample, i.e., from the results of a number of experiments. That statistic is distributed according to a well-known theoretical distribution, so it is easy to compute its probability, which is referred to as the p-value. Then, the p-value is compared to the confidence level: if it is smaller, then the conclusion is that there is enough evidence in the data to reject the null hypothesis; otherwise, the data does not provide enough evidence and the alternate hypothesis must be accepted.

In the literature, there are a variety of tests available [30], but not every test is applicable in our context. The existing tests can be roughly classified into parametric and non-parametric [24]. The former are generally applicable to data that is normally distributed and homoscedastic, i.e., have homogeneous variances, whereas the latter are applicable to any data. Normal and homoscedastic data are very common when analysing natural phenomena, but they are not so common in other contexts. Our experience proves that it is very unlikely that the values gathered for a performance measure are distributed normally and that it is even more unlikely that the values gathered for different proposals are homoscedastic [32, 33]. In other words, non-parametric tests are the choice in our context. Furthermore, there are non-parametric tests that work on only two samples and others that can work on multiple samples. In our context it is very unlikely that comparing two proposals only is enough to prove that one of them is promising enough; it is generally a good idea to compare new proposals to as many existing proposals as possible, which makes non-parametric tests for multiple samples the choice in our context. (Note that one might think that performing a test on multiple samples can be easily implemented using a two-sample test on every pair of samples, but this intuitive idea does not work because it does not keep the so-called accumulative family-wise error under control, i.e., the more pairs to be compared, the more likely that the results of a test are wrong.) Such tests can be further classified into bulk tests, which analyse if a number of proposals can be considered similar or different regarding a performance measure, and post-hoc tests, which compare a proposal to the rest ($1 \times n$ tests) or every possible pair ($n \times n$ tests). Obviously, it only makes sense to use a post-hoc test if a previous bulk test reveals that the proposals that are being compared do not behave similarly. Regarding the samples, there are paired and non-paired tests. The former work on samples that were gathered from exactly the same datasets, whereas the latter work on samples that were gathered from different datasets. Our proposal is to produce comparisons that are as homogeneous as possible, so it makes sense to perform the experiments on exactly the same datasets, which clearly justifies using paired tests.

Our conclusion is then that we have to use non-parametric, multiple-sample, paired bulk or post-hoc tests. In the literature, there are a variety of such tests available [10, 30]. Regarding the bulk tests, our recommendation is to use Iman-Davenport's test [17], because it is an extension of the classical Friedman's test that overcomes all of its limitations. There are other such tests in the literature. For instance, Chinchor et al. [5] recommended using the Approximate Randomization test and the Bootstrap test but Sprent [34] and Conover [6] pointed out several limitations regarding the former, including that it is unreliable in the presence of outliers; consequently, we cannot recommend it since it is not unlikely that a sample includes outliers that are intrinsic to the proposal being analysed (recall that only outliers that are due to environmental conditions can be removed from a sample); similarly, Efron and Tibshirani [11] found out that the Bootstrap method is not very accurate in general, which does not make it the appropriate choice in our context. Regarding $1 \times n$ and $n \times n$ post-hoc tests, our recommendation is to use Hommel's [16] and Bergmann-Hommel's [3] tests, respectively. The reason is that Derrac et al. [10] carried out an exhaustive experimental evaluation according to which these tests proved to provide the best balance between efficiency and robustness. Note, however, that Bergmann-Hommel's test is computationaly intractable when comparing more than 9 or 10 proposals. In such cases, the best choice is to use Shaffer's test [29]; unfortunately, this test also becomes intractable when comparing too many proposals; in such cases, the only applicable test is Hommel's test.

Before concluding, we would like to highlight that none of the previous tests work on the original samples, but on transformed rank samples. In other words, instead of working on the values of a performance measure, they work on the equivalent ranks. Note that this is not a shortcoming, but a feature that is intrinsic to non-parametric tests.

VI. Our method

Below, we report on a series of steps that we recommend should be followed when evaluating and comparing a new information extraction proposal, namely:

Step 1 Select some proposals with which the comparison will be performed. They should be the most recent and closely related, but it is strongly recommended that some state-of-the-art proposals are included even if they are not so closely related. This will help prove that the new proposal actually advances on the state of the art. From a statistical point of view, at least five proposals must be compared so that the results are statistically sound [10].

Step 2 Select the appropriate datasets from public repositories and document the attributes that you are going to extract from each one. They should be the same that were used when the other proposals where evaluated. More than that, the same splits must be used for training and evaluation purposes or, otherwise, the results will not be homogeneous. Derrac et al. [10] explained that there is not a consensus in the literature regarding how many experiments must be performed so that there are enough results to draw statistically-sound conclusions. Their experience proves that it is strongly advisable that the number of samples is between $2\,k$ and $8\,k$, where k denotes the number of proposals to compare.

Step 3 Describe the experimentation environment, including information regarding the hardware and the software used to run the experiments.

Step 4 Run the selected proposals on the selected datasets and collect performance measures. If the measures may be influenced by environmental conditions, then the experiments must be repeated several times and the environmental outliers must be removed.

Step 5 Provide tables with the results and analyse them intuitively using the average values of the performance measures and their standard deviations or variances. Pay special attention to the intrinsic outliers, and explain why they occur.

Step 6 Support your conclusions using the statistical tests that we recommended in the previous section. First, you must use Iman-Davenport's test to find out if the differences are statistically significant or not; if they are, then you can use Hommel's test to compare your proposal to the others (if you are just interested in proving that yours is better than the others); you can also use Bergman-Hommel's or Shaffer's tests to compare every proposal to every other, depending on the number of proposals to be compared.

VII. Illustrating our method

To illustrate our method, we next report on a comparison amongst five proposals. They were selected from the literature, but we keep them anonymous since our goal is not to prove that one of them is better than the others, but just to illustrate our method. We assume that the first proposal is a new proposal and that the authors wish to compare it to prove that it outperforms the others in terms of effectiveness or efficiency. We, however, will not focus exclusively on the first proposal; we will also analyse the others and will draw some conclusions that are statistically sound, but difficult to realise from the empirical data.

We have selected a total of 38 datasets from the RISE repository [25], Crescenzi and Mecca's repository [7], and Sleiman and Corchuelo's repository [33]. The first four columns of Table I show a summary of these datasets, which includes their names, the attributes to extract, and the number of documents in each dataset. Note that we are comparing five proposal using 38 datasets, which meets Derrac et al.'s recommendation.

The experimentation environment consisted of a computer that was equipped with a four-threaded Intel Core i7 processor than ran at 2.93 GHz, had 4 GiB of RAM, Windows 7 Pro 64-bit, and Oracle's Java Development Kit 1.7.0_02. The configuration parameters of the Java Virtual Machine were set to their default values.

The effectiveness measures collected were precision (*P*), recall (*R*), and the F_1 measure. The efficiency measures collected were the learning time (*LT*), i.e., the time to learn an extraction rule from a training dataset, and the extraction time (*ET*), i.e., the time to execute the extraction rule on a evaluation dataset; they both were measured in CPU seconds. Their means and deviations were computed and shown in the first and the second row of Table I. A dash in a cell means that the corresponding proposal was not able to learn an extraction rule in 15 CPU minutes, which we considered was quite a large timeout. Some of the proposals were unsupervised, and we used the method that we described in Section IV to deal with them automatically.

The results regarding the performance measures collected are presented in Table I. Proposal 1 seems to outperform the others regarding effectiveness since it achieves the highest precision, recall, and F_1 mean values; note, too, that the standard deviation is very small, which means that the proposal is quite stable, that is, that its results do not deviate largely from dataset to dataset. It also seems to outperform the others regarding learning time because this measure has the smallest value amongst the proposals that we have compared. Note, however, that Proposal 2 achieves an extraction time that is slightly smaller and Proposal 3 achieves an extraction time that is similar. Summing up, Proposal 1 seems to clearly outperform the others, except regarding the extraction time, which seems very similar to Proposal 2 and Proposal 3. However, Proposal 2 is not comparable to Proposal 1 in terms of effectiveness measures since the results regarding precision, recall, and F_1 are very poor. Actually, it seems to be the less effective one. Only Proposal 3 is comparable to Proposal 1 in terms of effectiveness since it achieved competitive results regarding precision, recall, and F_1, which were close to 0.8. Unfortunately, Proposal 3 achieved poor results regarding learning time which means that its learning process is very slow. Note that Proposal 3 has 214.70 deviation, which means that some learning times differ very much from one dataset to another. Proposal 4 and Proposal 5 would be in the middle of the ranking regarding effectiveness measures, in this order, and Proposal 4 is the worst one regarding extraction time. Intuitively, if we gave a little more importance to effectiveness measures instead of efficiency, the ranking of the proposals would be in this order: Proposal 1, Proposal 3, Proposal 4 and Proposal 5 (there is a tie between them), and, finally, Proposal 2.

To confirm that the previous intuitive conclusions are sound from a statistical point of view, we first need to run Iman-Davenport's test regarding every performance measure. To run this test, we first need transform the data in Table I into the corresponding ranks. Since this is quite a trivial process, we don't report on these data; instead, we just report on the average ranks of each proposal in Table II. Note that the p-value that this test outputs is nearly zero in every case, which is a strong indication that there are statistically significant differences between the proposals that we have analysed. It then makes sense to run Bergmann-Hommel's test to compare every proposal to the others, so that we can find a complete rank order amongst them. Table II also reports on the p-values that this test outputs for each comparison; for the sake of readability, the last column also shows the interpretation of these p-values since it provides an explicit statistical rank for each proposal. Note that these results confirm our intuitive interpretation of the experimental results: Proposal 1 ranks the first regarding every effectiveness and efficiency measure; the only tie is regarding extraction time, which does not seem to be significantly different from the extraction time of Proposal 2.

Surprisingly, we cannot uphold that Proposal 3 is the second one in the rank of effectiveness measures since no statistical

	Summary	N	P	R	F1	LT	ET	P	R	F1	LT	ET	P	R	F1	LT	ET	P	R	F1	LT	ET	P	R	F1	LT	ET
	Mean	41.42	0.97	0.94	0.95	0.13	0.02	0.52	0.49	0.49	9.03	0.01	0.79	0.88	0.81	141.89	0.19	0.82	0.60	0.65	6.56	46.96	0.74	0.61	0.64	7.96	12.66
	Standard deviation	49.92	0.07	0.13	0.10	0.18	0.04	0.40	0.42	0.41	10.66	0.01	0.22	0.16	0.18	214.70	0.22	0.15	0.32	0.29	5.41	59.50	0.21	0.31	0.28	5.53	14.94
Dataset	**Attributes**	**N**	**P**	**R**	**F1**	**LT**	**ET**	**P**	**R**	**F1**	**LT**	**ET**	**P**	**R**	**F1**	**LT**	**ET**	**P**	**R**	**F1**	**LT**	**ET**	**P**	**R**	**F1**	**LT**	**ET**
Abe Books	title, author, price, isbn	30	1.00	1.00	1.00	0.03	0.00	0.65	0.58	0.61	7.19	0.01	0.92	0.99	0.95	15.46	0.12	0.87	0.58	0.70	9.09	26.96	0.52	0.16	0.24	9.66	10.26
Awesome Books	title, author, price, isbn, year	30	1.00	0.87	0.93	0.03	0.00	0.78	0.53	0.63	3.46	0.01	0.85	1.00	0.92	8.14	0.14	1.00	0.39	0.56	5.68	16.52	0.77	0.26	0.39	5.01	6.27
Better World Books	title, author, price	30	0.99	1.00	0.99	0.17	0.00	-	-	-	12.59	-	0.99	0.96	0.97	85.32	0.39	0.98	0.99	0.98	16.15	41.39	0.43	0.35	0.39	15.57	17.04
Disney Movies	title, actor, year, runtime	30	1.00	1.00	1.00	0.07	0.00	0.67	0.67	0.67	3.03	0.01	0.71	0.67	0.69	259.23	0.08	0.97	0.97	0.97	4.34	44.69	0.72	0.72	0.72	3.95	3.82
OTWF	title, director, actor, year, runtime	30	1.00	1.00	1.00	0.02	0.00	0.94	0.94	0.94	1.40	0.00	1.00	1.00	1.00	29.70	0.05	0.92	0.72	0.81	3.70	10.39	0.39	0.30	0.34	2.79	3.79
Soul Films	title, director, actor, year	30	0.99	0.92	0.95	0.03	0.00	0.50	0.50	0.50	4.54	0.01	0.59	1.00	0.74	17.24	0.05	0.99	0.95	0.97	10.64	37.58	0.91	0.81	0.86	9.73	9.36
Auto Trader	color, doors, engine, mileage, model, price, transmission, type	30	0.99	1.00	1.00	0.28	0.03	-	-	-	-	-	-	-	-	-	-	0.89	0.87	0.88	14.87	103.82	0.89	0.00	0.00	14.18	14.02
Car Max	color, mileage, model, price, transmission, year, type	30	1.00	1.00	1.00	0.21	0.02	0.98	0.98	0.98	12.07	0.01	0.45	0.89	0.60	34.21	0.20	0.89	0.89	0.89	7.66	22.90	0.88	0.88	0.88	7.43	7.63
Internet Autoguide	color, doors, engine, location, mileage, price, transmission, type	30	1.00	1.00	1.00	0.12	0.05	0.92	1.00	0.95	4.02	0.01	0.97	0.94	0.96	117.16	0.19	-	-	-	8.74	-	0.11	0.11	0.11	7.78	9.02
Linked In	date, place, title, url	30	0.96	0.96	0.96	0.04	0.02	0.53	0.53	0.53	1.37	0.00	-	-	-	-	-	0.87	0.55	0.68	5.04	42.04	0.57	0.20	0.30	4.15	3.34
Moendi	date, place, title, url	30	1.00	1.00	1.00	0.01	0.00	0.87	0.87	0.87	0.72	0.00	0.90	1.00	0.95	1.56	0.03	0.60	0.60	0.60	2.61	5.74	0.80	0.40	0.53	1.86	1.28
RD Learning	date, place, title, url	30	0.99	0.99	0.99	0.02	0.00	0.77	0.77	0.77	0.51	0.00	0.99	0.79	0.88	6.21	0.02	0.52	0.39	0.45	4.62	7.10	0.62	0.23	0.34	3.24	1.95
Web MD	name, address, phone, specialty	30	1.00	1.00	1.00	0.02	0.00	0.06	0.06	0.06	25.26	0.01	0.77	1.00	0.87	11.81	0.03	0.86	0.45	0.59	8.28	29.94	0.60	0.60	0.60	9.45	14.54
Ame. Medical Assoc.	name, address, phone, specialty	30	0.98	1.00	0.99	0.02	0.02	-	-	-	5.57	-	-	-	-	-	-	0.79	0.39	0.53	6.07	10.53	0.60	0.60	0.60	6.99	7.38
Steady Health	name, address, specialty	30	1.00	1.00	1.00	0.67	0.02	1.00	1.00	1.00	11.33	0.03	0.83	0.83	0.83	9.59	0.11	0.75	0.25	0.38	9.70	40.00	0.75	0.75	0.75	9.56	9.77
4 Jobs	company, location, category	30	0.92	0.98	0.95	0.09	0.02	0.00	0.00	0.00	2.98	0.01	-	-	-	-	-	0.45	0.25	0.32	9.09	36.60	0.94	0.94	0.94	9.02	9.50
6 Figure Jobs	company, location, category	30	1.00	1.00	1.00	0.04	0.02	0.52	0.52	0.52	21.26	0.02	1.00	0.98	0.99	90.78	0.34	0.75	0.00	0.00	11.76	26.79	0.25	0.25	0.25	11.56	11.93
Career Builder	company, location, category	30	1.00	1.00	1.00	0.05	0.00	0.00	0.00	0.00	5.34	0.01	0.80	0.83	0.82	266.00	0.31	0.75	0.00	0.00	8.13	50.72	0.75	0.75	0.75	8.11	7.72
Yahoo!	beds, baths, size, price	30	1.00	1.00	1.00	0.15	0.02	-	-	-	15.66	-	0.77	0.97	0.86	246.95	0.89	1.00	1.00	1.00	16.32	86.55	0.83	0.83	0.83	15.62	16.75
Hart	address, beds, price	30	1.00	1.00	1.00	0.05	0.00	0.79	0.79	0.79	6.97	0.01	0.94	1.00	0.97	20.76	0.09	1.00	1.00	1.00	7.43	100.85	0.78	0.75	0.76	7.04	6.96
Trulia	address, beds, baths, size, price	30	0.63	1.00	0.77	0.79	0.05	0.00	0.00	0.00	46.78	0.03	-	-	-	-	-	0.88	0.92	0.90	25.55	284.94	1.00	0.89	0.94	25.18	25.37
Player Profiles	name, birth, hight, weight, club	30	1.00	1.00	1.00	0.11	0.23	-	-	-	-	-	0.36	0.99	0.52	14.68	0.06	0.83	0.13	0.23	5.51	17.43	1.00	1.00	1.00	5.34	4.96
NFL	name, birth, hight, weigth, age, college	30	1.00	1.00	1.00	0.19	0.00	0.98	0.98	0.98	34.30	0.03	0.53	0.81	0.64	159.39	0.39	0.71	0.71	0.71	10.14	39.91	0.86	0.86	0.86	9.64	9.56
Soccer Base	name, birth, age, hight, weigth, country, position, club	30	0.97	1.00	0.98	0.27	0.03	0.95	1.00	0.97	14.93	0.01	-	-	-	-	-	0.89	0.89	0.89	11.54	47.80	0.89	0.89	0.89	11.04	10.87
Amazon (cars)	make, model, price	21	0.93	0.73	0.82	0.01	0.00	0.00	0.00	0.00	1.36	0.01	0.60	0.67	0.63	2.29	0.05	0.98	1.00	0.99	1.06	10.09	0.97	1.00	0.98	19.47	8.10
UEFA (players)	name, country	20	1.00	0.90	0.95	0.02	0.02	1.00	1.00	1.00	0.62	0.00	0.91	0.94	0.92	10.81	0.05	0.92	0.90	0.91	1.34	11.64	0.92	0.51	0.66	3.53	3.70
Amazon (popartist)	name	19	1.00	0.98	0.99	0.02	0.00	0.00	0.00	0.00	2.45	0.00	1.00	1.00	1.00	246.97	0.11	0.94	0.72	0.82	3.42	23.35	0.92	0.58	0.71	15.66	10.22
UEFA (teams)	assoc.,country,fifa-aff.,founded,gen-secretary,president,press-officer,team,uefa-aff.	20	0.99	0.99	0.99	0.03	0.00	0.92	0.92	0.92	0.45	0.00	0.97	0.99	0.98	0.89	0.02	0.81	0.89	0.85	1.00	3.32	0.64	0.75	0.69	1.79	2.28
Aus Open	name, birth-date, birth-place, country, height, money, weight	29	1.00	1.00	1.00	0.26	0.02	0.70	0.85	0.77	4.84	0.01	0.24	0.82	0.37	132.24	0.27	0.65	0.28	0.39	1.98	22.06	0.67	0.32	0.43	9.67	16.96
E-Bay	price, bids, location	50	0.97	1.00	0.98	0.51	0.08	0.82	0.82	0.82	3.73	0.01	0.83	1.00	0.91	577.97	0.48	0.70	0.12	0.20	2.23	16.04	0.70	0.12	0.20	5.44	19.55
Major League	name, position, team	9	0.98	0.55	0.70	0.04	0.00	0.00	0.00	0.00	3.28	0.01	0.99	1.00	0.99	158.33	0.06	0.99	0.46	0.63	1.29	19.61	0.65	0.33	0.44	3.60	6.13
Netflix	title, director, length, year	50	0.99	0.99	0.99	0.18	0.02	0.76	0.79	0.78	11.62	0.01	0.82	0.80	0.81	706.64	0.76	0.94	0.82	0.88	3.68	34.15	0.99	0.99	0.99	7.47	31.54
RPM Find	name, description, os	20	0.95	0.97	0.96	0.03	0.05	0.10	0.00	0.00	1.28	0.01	-	-	-	-	-	0.72	0.03	0.06	1.40	26.22	0.99	0.99	0.99	1.29	10.42
Bigbook	name, city, phone, street	235	0.95	0.94	0.94	0.12	0.05	0.01	0.00	0.00	25.71	0.00	-	-	-	-	-	0.81	0.77	0.79	1.50	166.23	0.68	0.98	0.80	2.68	89.14
IAF	name, email, organisation, provider	252	0.84	0.38	0.52	0.12	0.00	0.90	0.09	0.17	0.70	0.00	0.53	0.69	0.60	14.41	0.06	0.37	0.43	0.40	1.39	13.34	1.00	1.00	1.00	2.00	2.40
Okra	name, email	10	1.00	0.82	0.90	0.08	0.00	0.12	0.01	0.02	20.51	0.00	0.49	0.34	0.40	849.27	0.37	0.83	0.82	0.82	1.73	216.36	0.60	0.67	0.63	1.64	35.08
LA Weekly	name, address, phone	28	0.97	0.92	0.94	0.01	0.00	0.00	0.00	0.00	0.47	0.00	0.83	0.57	0.68	4.24	0.05	0.87	0.49	0.63	2.18	23.45	0.90	0.80	0.85	0.90	2.73
Zagat	name, address, type	91	1.00	0.86	0.92	0.17	0.05	0.00	0.00	0.00	6.60	0.00	1.00	0.98	0.99	158.48	0.06	0.81	0.63	0.71	2.57	20.39	0.81	0.72	0.76	13.60	19.75

Table I
EXPERIMENTAL RESULTS.

differences were found, except for its recall, regarding which it was ranked on the second place. Furthermore, despite of the fact that Proposal 2 seemed to be the worst one regarding effectiveness measures, no statistical differences were found to support this intuitive conclusion. Regarding learning time, Proposal 3 is the worst one as we stated in our previous conclusions; regarding extraction time, there were statistical differences that show that after Proposal 1 and Proposal 2 that are the best ones, Proposal 3 outperforms Proposal 4, which, in turn, outperforms Proposal 5.

The conclusion is that the experiments provide enough evidence to conclude that Proposal 1 outperforms the others; then comes Proposal 2, Proposal 3, Proposal 5, and, finally, Proposal 4.

VIII. CONCLUSIONS

We are very interested in practical web content mining, which is a sub-field of web mining that deals with applying knowledge discovery techniques to the information that is available in typical web documents. Unfortunately, these techniques are complicated by the fact that prior to using the data, the pages that contain it must be retrieved from the Web or a local repository, and then the information that they provide in user-friendly formats must be extracted and formatted according to a computer-procesable schema. There are many techniques available in the literature, which makes it difficult for practitioners to make an informed decision regarding which of them is the most appropriate for a particular web content mining problem. Evaluating and comparing information extraction proposals is not easy, and there's not a clear consensus in the literature.

We have reported on an evaluation method for information extraction proposals that work on semi-structured documents. Our method helps researchers make sure that their new proposals advance the state of the art not only conceptually, but from an empirical point of view; it also helps practitioners make informed decisions on which proposal is the most adequate for a particular information extraction problem. We have provided a few guidelines regarding repositories, datasets, and how to split them into training and evaluation datasets; we have also reported on the most common measures to assess the performance of a proposal and a method to compute them

Measure	Sample ranking		Iman-Davenport's test	Bergmann-Hommels's test					Statistical ranking	
	Proposal	Rank	P-Value	P-Value	Proposal 1	Proposal 2	Proposal 3	Proposal 4	Proposal 5	Proposal	Rank
P	Proposal 1	1.47	8.30E-13	Proposal 1	-	9.14E-11	2.31E-05	7.47E-05	3.89E-07	Proposal 1	1
	Proposal 4	3.03		Proposal 2		-	6.69E-02	6.67E-02	3.14E-01	Proposal 4	2
	Proposal 3	3.12		Proposal 3			-	8.00E-01	7.82E-01	Proposal 3	2
	Proposal 5	3.43		Proposal 4				-	7.82E-01	Proposal 5	2
	Proposal 2	3.95		Proposal 5					-	Proposal 2	2
R	Proposal	Rank	P-Value	P-Value	Proposal 1	Proposal 2	Proposal 3	Proposal 4	Proposal 5	Proposal	Rank
	Proposal 1	1.59	6.24E-13	Proposal 1	-	1.38E-08	4.44E-02	4.74E-07	2.11E-07	Proposal 1	1
	Proposal 3	2.51		Proposal 2		-	2.60E-03	1.00E+00	1.00E+00	Proposal 3	2
	Proposal 4	3.51		Proposal 3			-	1.17E-02	8.81E-03	Proposal 4	3
	Proposal 5	3.59		Proposal 4				-	1.00E+00	Proposal 5	3
	Proposal 2	3.79		Proposal 5					-	Proposal 2	3
F1	Proposal	Rank	P-Value	P-Value	Proposal 1	Proposal 2	Proposal 3	Proposal 4	Proposal 5	Proposal	Rank
	Proposal 1	1.42	1.87E-14	Proposal 1	-	9.14E-11	7.47E-04	2.60E-07	3.89E-08	Proposal 1	1
	Proposal 3	2.78		Proposal 2		-	1.23E-02	4.71E-01	4.71E-01	Proposal 3	2
	Proposal 4	3.38		Proposal 3			-	1.90E-01	1.16E-01	Proposal 4	2
	Proposal 5	3.53		Proposal 4				-	6.90E-01	Proposal 5	2
	Proposal 2	3.89		Proposal 5					-	Proposal 2	2
LT	Proposal	Rank	P-Value	P-Value	Proposal 1	Proposal 2	Proposal 3	Proposal 4	Proposal 5	Proposal	Rank
	Proposal 1	1.00	5.64E-40	Proposal 1	-	1.05E-06	3.25E-24	2.64E-09	2.59E-08	Proposal 1	1
	Proposal 2	2.87		Proposal 2		-	1.05E-06	8.30E-01	8.30E-01	Proposal 2	2
	Proposal 5	3.11		Proposal 3			-	7.09E-05	1.46E-05	Proposal 5	2
	Proposal 4	3.26		Proposal 4				-	8.30E-01	Proposal 4	2
	Proposal 3	4.76		Proposal 5					-	Proposal 3	3
ET	Proposal	Rank	P-Value	P-Value	Proposal 1	Proposal 2	Proposal 3	Proposal 4	Proposal 5	Proposal	Rank
	Proposal 2	1.50	8.66E-88	Proposal 1	-	9.42E-01	1.94E-04	1.21E-20	1.98E-11	Proposal 1	1
	Proposal 1	1.53		Proposal 2		-	1.94E-04	1.01E-20	1.98E-11	Proposal 2	1
	Proposal 3	2.97		Proposal 3			-	1.41E-07	7.42E-03	Proposal 3	2
	Proposal 5	4.03		Proposal 4				-	1.80E-02	Proposal 5	3
	Proposal 4	4.97		Proposal 5					-	Proposal 4	4

Table II
STATISTICAL RANKING.

automatically in the case of unsupervised proposals; we have also surveyed the literature on statistical inference and we have selected the most adequate statistical tests to confirm or refute if the intuitive conclusions that we can draw from our empirical results can be sustained or not.

We expect that this method helps the many researchers in the field of web information extraction compare their proposals more homogeneously and sustain their conclusions so that they can prove that their new proposal actually outperform others in the literature. We expect that practitioners who are interested in web content mining have a useful tool to make informed decisions on which the most appropriate proposal is regarding a particular web content mining problem.

REFERENCES

[1] Álvarez, M., Pan, A., Raposo, J., Bellas, F., Cacheda, F.: Extracting lists of data records from semi-structured web pages. Data Knowl. Eng. 64(2), 491–509 (2008)

[2] Arasu, A., Garcia-Molina, H.: Extracting structured data from web pages. In: SIGMOD Conference. pp. 337–348 (2003)

[3] Bergmann, B., Hommel, G.: Improvements of general multiple test procedures for redundant systems of hypotheses. In: Multiple Hypotheses Testing, pp. 100–115. Springer (1988)

[4] Chang, C.H., Kayed, M., Girgis, M.R., Shaalan, K.F.: A survey of web information extraction systems. IEEE Trans. Knowl. Data Eng. 18(10), 1411–1428 (2006)

[5] Chinchor, N., Hirschman, L., Lewis, D.D.: Evaluating message understanding systems: An analysis of the third message understanding conference (muc-3). Computational Linguistics 19(3), 409–449 (1993)

[6] Conover, W.J.: Practical nonparametric statistics. Wiley series in probability and statistics, Wiley, 3. ed edn. (1999)

[7] Crescenzi, V., Mecca, G.: Automatic information extraction from large websites. J. ACM 51(5), 731–779 (2004)

[8] Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., genevieve Gorrell, Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M.A., horacio Saggion, Petrak, J., Li, Y., Peters, W.: Text Processing with GATE (Version 6). GATE (2011)

[9] Demetriou, G., Gaizauskas, R.J., Sun, H., Roberts, A.:

Annalist - annotation alignment and scoring tool. In: LREC (2008)
[10] Derrac, J., García, S., Molina, D., Herrera, F.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation 1(1), 3–18 (2011)
[11] Efron, B., Tibshirani, R.: An introduction to the Bootstrap (1993)
[12] Feilmayr, C., Pröll, B., Linsmayr, E.: EVALIEX: A proposal for an extended evaluation methodology for information extraction systems. In: LREC. pp. 2303–2310 (2012)
[13] Harman, D., Liberman, M.: TIPSTER complete (1993)
[14] Hirschman, L.: The evolution of evaluation: Lessons from the message understanding conferences. Computer Speech and Language 12(4), 281–305 (1998)
[15] Hodge, V.J., Austin, J.: A survey of outlier detection methodologies. Artif. Intell. Rev. 22(2), 85–126 (2004)
[16] Hommel, G.: A stagewise rejective multiple test procedure based on a modified Bonferroni test. Biometrika 75(2), 383–386 (1987)
[17] Iman, R.L., Davenport, J.M.: Approximations of the critical region of the Friedman statistic. Communications in Statistics A9(6), 571–595 (1980)
[18] Krishnamurthy, R., Li, Y., Raghavan, S., Reiss, F., Vaithyanathan, S., Zhu, H.: SystemT: a system for declarative information extraction. SIGMOD Record 37(4), 7–13 (2008)
[19] Kuhlins, S., Tredwell, R.: Toolkits for generating wrappers. In: NetObjectDays. pp. 184–198 (2002)
[20] Kushmerick, N., Thomas, B.: Adaptive information extraction: Core technologies for information agents. In: AgentLink. pp. 79–103 (2003)
[21] Laender, A.H.F., Ribeiro-Neto, B.A., da Silva, A.S., Teixeira, J.S.: A brief survey of web data extraction tools. SIGMOD Record 31(2), 84–93 (2002)
[22] Lavelli, A., Califf, M.E., Ciravegna, F., Freitag, D., Giuliano, C., Kushmerick, N., Romano, L., Ireson, N.: Evaluation of machine learning-based information extraction algorithms: criticisms and recommendations. Language Resources and Evaluation 42(4), 361–393 (2008)
[23] Meng, W., Yu, C.T.: Advanced Metasearch Engine Technology. Morgan and Claypool (2010)
[24] Minnotte, M.: Introduction to modern nonparametric statistics. The American Statistician 61, 184–184 (2007)
[25] Muslea, I.: RISE: repository of online information sources used in information extraction (1998)
[26] Olston, C., Najork, M.: Web crawling. Foundations and Trends in Information Retrieval 4(3), 175–246 (2010)
[27] Petasis, G., Karkaletsis, V., Paliouras, G., Androutsopoulos, I., Spyropoulos, C.D.: Ellogon: A new text engineering platform. In: LREC (2002)
[28] Sarawagi, S.: Information extraction. Foundations and Trends in Databases 1(3), 261–377 (2008)
[29] Shaffer, J.P.: Modified sequentially rejective multiple test procedures. Journal of the American Statistical Association 81(395), 826–831 (1986)
[30] Sheskin, D.J.: Handbook of Parametric and Nonparametric Statistical Procedures. Chapman and Hall/CRC, 5 edn. (2012)
[31] Sleiman, H.A., Corchuelo, R.: An information extraction framework. In: PAAMS. pp. 149–156 (2012)
[32] Sleiman, H.A., Corchuelo, R.: TEX: An efficient and effective unsupervised web information extractor. Knowl.-Based Syst. 39, 109–123 (2013)
[33] Sleiman, H.A., Corchuelo, R.: Trinity: On using trinary trees for unsupervised web data extraction. IEEE Trans. Knowl. Data Eng. 26(6), 1544–1556 (2014)
[34] Sprent, P.: Data driven statistical methods. Chapman and Hall (1998)
[35] Suchanek, F.M., Sozio, M., Weikum, G.: SOFIE: a self-organizing framework for information extraction. In: WWW. pp. 631–640 (2009)
[36] Sundheim, B.: TIPSTER/MUC-5: information extraction system evaluation. In: MUC. pp. 27–44 (1993)
[37] Turmo, J., Ageno, A., Català, N.: Adaptive information extraction. ACM Computing Surveys 38(2) (2006)
[38] Yamada, Y., Craswell, N., Nakatoh, T., Hirokawa, S.: Testbed for information extraction from the Deep Web. In: WWW (Alternate Track Papers and Posters). pp. 346–347 (2004)