When Different is Wrong: Visual Unsupervised Validation for Web Information Extraction

Benoit Potvin and Roger Villemaire

Université du Québec à Montréal, Department of Computer Science,
Montréal, H3C 3P8, Canada

Abstract. This paper shows how visual information can be used to identify false positive entities from those returned by a state-of-the-art web information extraction algorithm and hence further improve extraction results. The proposed validation method is unsupervised and can be integrated into most web information extraction systems effortlessly without any impact on existing processes, system's robustness or maintenance. Instead of relying on visual patterns, we focus on identifying visual outliers, i.e. entities that visually differ from the norm. In the context of web information extraction, we show that visual outliers tend to be erroneous extracted entities. In order to validate our method, we post-processed the entities obtained by Boilerpipe, which is known as the best overall main content extraction algorithm for web documents. We show that our validation method improves Boilerpipe's initial precision by more than 10% while F_1 score is increased by at least 3% in all relevant cases.

1 Introduction

Visual information plays an important role in web pages that have been designed for humans. Web technologies have evolved to operate on powerful devices that can execute considerable front end calculation and enhance user experience. Visually rich documents such as web pages and PDF files contain visual information that supplements the meaning of textual information and facilitates comprehension. Without visual formatting, a website would be much more difficult to understand and navigate, perhaps even incomprehensible to the user. Accordingly, visual features that help human users to understand a document can also help data extraction.

Recent works have emphasized the importance of visual elements in web mining tasks [5,20,4,11,13]. Visually oriented web information extraction (WIE) methods use discriminant regularities (patterns) across visual information in order to improve extraction results. Visual characteristics can obviously be used with different motivations by web designers, sometimes in very creative ways. Consequently, most visual patterns are not expected to be consistent across the World Wide Web. In order to rely on consistent patterns, visually oriented WIE methods have a limited range of application. Accordingly, these methods can be classified in two broad categories, i.e. whether the visual regularities rely

on a limited set of documents (a corpus) or on an object with recurrent visual cues (e.g. a table or a content block). In the former case, good performances are obtained at the expense of generality (i.e. the possibility to process unseen documents) and robustness (i.e. the stability of the method following template modifications). In the latter case, most visual information is ignored in order to rely on generic regularities.

Many disadvantages are associated with the use of currently available visually oriented WIE methods :

1. Extraction time is often compromised. Instead of relying on the HTML response of the HTTP request, CSS properties must be computed for all document object model (DOM) nodes. On large sets of documents, extraction time can become impractical.
2. Visually oriented WIE methods are laborious to develop. Visual information is managed by means of ad hoc knowledge, i.e. rules or patterns that have been defined to fit a specific need. These extraction rules have to be crafted by experts or learned through (semi-)supervised algorithms on an annotated corpus.
3. Integration to existing systems can be arduous. In most cases, visually oriented WIE methods have to be combined with other extraction methods based on different types of patterns (e.g. across HTML tags). This can necessitate considerable efforts and/or reengineering of existing systems.
4. Visual patterns can compromise system's robustness. Template modifications can have consequences on exploited regulaties. In the industry, robustness (and subsequently maintenance) is a key issue of WIE systems.

Although visual information plays a key role in the meaning of web documents, the use of visually oriented WIE methods involves significant drawbacks. This gap between the importance of visual information in web documents and the possibility to optimally exploit this type of information in web mining tasks has motivated our research.

This paper shows how visual information from a set of formerly extracted entities can be used following a WIE task to identify false positive entities and hence further improve extraction results. The method is unsupervised and does not rely on any visual pattern. Instead, we focus on identifying visual outliers, i.e. entities that visually differ from the norm. In the context of WIE, we show that visual outliers tend to be erroneous extracted entities. We assume that 1) state-of-the-art WIE systems extract more true positive entities than false positive entities and 2) extracted entities are visually similar.

The advantages of the proposed validation method are the following:

1. The method is unsupervised and do not require any annotation, learning, or rule definition.
2. The method can be integrated into most state-of-the-art WIE systems effortlessly, as it is a validation process based on formerly extracted entities.
3. The method has no impact on the robustness or maintenance of the system because it does not rely on visual patterns and is used for validation.

4. Only computed visual properties of extracted entities are required, which can represent a substantial saving in computation time compared to extraction methods that rely on visual patterns.

In order to validate our method, we post-processed the entities obtained by the top state-of-the-art extraction algorithm Boilerpipe, which is the best overall main content extraction algorithm for web documents [24]. Main content extraction is an important WIE task for both research and industry, as it allows to remove the surplus "clutter" (boilerplate, templates) around the main textual content of a web page and improve subsequent extraction tasks. Our method improves Boilerpipe's initial precision by more than 10% while F_1 score is increased by at least 3% in all relevant cases.

The contributions of this paper are:

1. We show that in a set of formerly extracted entities obtained by a state-of-the-art data extraction algorithm, visual outliers tend to be false positive entities.
2. We also show that visual outliers can be eliminated in order to improve precision and F_1 score (i.e. with minimal impact on recall).
3. We show that two established anomaly detection algorithms (k-NN and HBOS) can be used to identify relevant visual outliers.
4. We improve Boilerpipe main content extraction algorithm, which is the best overall main content extraction algorithm.
5. More generally, we show how visual information of web documents can be exploited in an unsupervised manner in web data extraction tasks without impacting on system's flexibility and robustness.
6. To our knowledge, we are the first authors to introduce *visual outliers*, i.e. point anomalies based on visual information, in WIE.

2 Background

Web wrappers (also called *web extractors*) are algorithms that extract data from unstructured or semi-structured web sources and map them to a suitable structured format for further processing [7]. However, wrappers rarely embed the full process executed by browsers [11,24].

When a human user visits a website the web browser creates a well-formed DOM tree from the (possibly broken) HTML code contained in the HTTP response. We will refer to this tree as the *first DOM tree*. The DOM is a W3C recommendation that allows programs and scripts to dynamically access and update the content, structure and style of documents. The browser then parses stylesheets and generates style boxes for all elements of the DOM tree according to the CSS box model and CSS visual formatting model [5]. JavaScript code is parsed and executed in order to update HTML elements, attributes, CSS style properties and events, yielding what we will call the *rendered DOM tree*. Finally, the browser renders the page on the user's screen.

Most web wrappers rely on the first DOM tree in order to extract information [11,24] despite the fact that it represents an approximation of the resulting page, i.e. a lone well-formed HTML document [5] where the inherent complexity of visually rich websites is mostly ignored [20,24]. Wrappers solely based on the first DOM tree are limited, particularly when:

1. HTML structure is highly variable or complex [9].
2. HTML code is not written properly (e.g. when a table is defined with *div* elements with absolute positioning) [12,11].
3. JavaScript code is present [9,20,24].
4. Web pages are visually rich [5,9,10,8,4,11,13].
5. Source code is hardly accessible [9].

Visually oriented WIE methods[1] usually overcome such limitations by dealing with the rendered representation of web documents.

3 Anomaly Detection

Anomaly detection is the process of identifying unexpected items or events in a dataset [14]. Unsupervised methods mostly deal with point anomalies, i.e. single anomalous instances that differ from the norm. These methods typically return, for each instance, a score based on the intrinsic properties of the dataset. This score is interpreted as a degree of abnormality [14]. For the detection of visual outliers in a set of extracted entities, we will focus on point anomalies and therefore use unsupervised methods.

Point anomalies can furthermore differ locally or globally from the norm, i.e. depending if each item is compared to the whole dataset or only to its closest neighborhood. In our case, visual outliers differ from all other elements, so we restrict ourselves to global anomaly detection methods.

Goldstein and Uchida [14] present a comparative evaluation of 19 unsupervised anomaly detection algorithms on ten different datasets from multiple application domains. For global anomaly detection, as in our case, they recommend k-nearest-neighbors based algorithms if computation time is not an issue and histogram-based algorithms when computation time is essential, especially for large datasets.

K-nearest-neighbors techniques assume that normal items occur in dense neighborhoods while anomalous instances are far from their closest neighbours [6]. The anomaly score of each item is computed relatively to the distance of its k-nearest-neighbors. This distance can be measured relatively to the k^{th}-nearest-neighbors or to the average distance of all of the k-nearest-neighbors. The first method is referred to as k^{th}-NN and the latter as k-NN. In practical applications, k-NN is often preferred [14].

[1] In the literature, visual web information extraction may refer to the use of a graphical user interface (GUI) that allows the user to generate wrappers. This is not the intended meaning here as we refer to the visual formatting of documents.

Histogram-based anomaly detection algorithms use histograms to maintain a profile of normal instances. Such approach is also referred to as frequency-based or counting-based [6]. For each feature of the dataset, a histogram is created based on the different values taken by that feature. Then each instance is evaluated according to the profile of its features. The most common approach for unsupervised histogram-based algorithms is to compute an anomaly score based on the height of the histogram in which each feature falls. The histogram-based outlier score (HBOS) algorithm proposed by Goldstein et al. [14] is obtained by multiplying the inverse heights of each histogram – representing the density estimation – in which the features resides. It is worth noting that HBOS assumes the independence of the features. This allows a fast processing speed. In some cases HBOS can process a dataset under a minute, whereas nearest-neighbors based algorithms take over 23 hours [15].

We will hence use both k-NN and HBOS algorithms for visual outlier detection.

4 Proposed Method

The proposed method consists of three steps, as shown in figure 1:

1. *Information gathering*: Given a set of extracted DOM nodes by some WIE system, we use the XML Path Language (XPath) expressions of each node and their related URLs in order to obtain their visual characteristics from the rendered DOM tree, i.e., the computed style properties, as their are shown to the user.
2. *Anomaly Detection*: Given the set of CSS characteristics of all extracted nodes, we use an unsupervised anomaly detection algorithm to detect visual outliers. For each node, we hence obtain an *anomaly score* for which high scores denote entities that differ the most from the norm.
3. *Cleaning task*: Based on the anomaly scores, we rank the nodes in descending order and we successively delete the most visually abnormal ones. Visual outliers are eliminated from the initial set of extracted nodes and an improved set of nodes is returned.

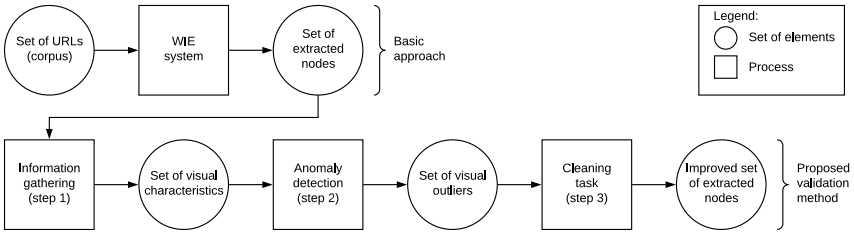

Fig. 1. The successive steps of the proposed method

4.1 Rationale

The rationale of our approach is that WIE systems aim at extracting entities of a specific kind, such as titles, prices, product descriptions, the main content of news articles, etc. These entities can either have a simple structure like titles (strings) and prices (numbers), or a complex one such as product descriptions (formed of specific fields) or main content of news articles (containing a title, a publication date, author's names, a location, subtitles, paragraphs and correction notes).

Assuming a WIE system with fair performance, most extracted DOM nodes should belong to the same type of entity or, in the case of complex entities, should be composed of a fixed set of types (e.g., all news articles should contain a title, an author, a publication date, etc). We should also expect that similar entities share similar visual characteristics. For example, a set of titles extracted from scientific articles should share some common visual patterns. However, we do not try to identify these visual similarities, but only leverage their existence in order to detect and remove visual outliers.

4.2 Data normalization

CSS values can be quantitative or qualitative. We normalize all data in order to deal exclusively with numeric values. Moreover, k-NN algorithm assumes that attributes are normalized and are of equal importance. We use standard normalization techniques for data mining as shown in [25]:

1. If value is numeric then we keep it as it is.
2. If value is a RGB color then R, G and B are separated in three different columns, each containing a numeric value.
3. If value is numeric but followed by some unit of measurement then we use the same unit and only keep the numerical part.
4. If value is qualitative then we create a new column for each categorial possibility (according to the W3C CSS specification) and store the value as binary numbers (i.e. "1" in the corresponding column, else "0").

Consequently, we obtain a set of multivariate tabular data where all CSS values are numeric. There are hundreds of CSS properties and most of them are in practice rarely used. Therefore, we retain the 32 CSS properties where variations are the most frequent.[2]

[2] Retained properties are the following: background-color; border-bottom-color; border-bottom-style; border-bottom-width; border-left-color; border-left-style; border-left-width; border-right-color; border-right-style; border-right-width; border-top-color; border-top-left-radius; border-top-right-radius; border-top-style; border-top-width; color; font-size; font-style; font-weight; margin-bottom; margin-left; margin-right; margin-top; outline-color; padding-bottom; padding-left; padding-right; padding-top; position; text-align; text-decoration; visibility;

5 Evaluation

In this section we evaluate our method based on the extraction results of Boilerpipe[3], a well-known main content extraction algorithm [18,24]. Boilerpipe aims to remove the surplus "clutter" (boilerplate, templates) around the main textual content of a web page (i.e. all content that is not related to the main content, e.g. navigational elements, advertisements, footers, etc). It uses a set of shallow text features – such as text density and link density – to classify the individual text elements in web pages. Boilerpipe is the overall best algorithm for main content extraction [19,24] and has a specific strategy that is tuned towards news articles.

5.1 Performance

Standard measures of performance for wrappers are precision, recall and F_1 score. Precision (P) is the quotient $\frac{TP}{TP+FP}$ of the number of true positive elements on the number of retrieved elements (true and false positives). Recall (R) is the quotient $\frac{TP}{TP+FN}$ of the number of true positive elements on the number of relevant elements (true positives and false negatives). F_1 score is the weighted harmonic mean $\frac{2P \cdot R}{P+R}$ of precision and recall.

Since our method consists in filtering out some elements extracted by an existing WIE system, the number of retrieved true positives can only decrease while the absolute number of true positives and false negatives remains constant. Recall can hence only decrease. However, retaining very few elements (low recall) could dramatically improve precision. We will hence use the F_1 score to evaluate our method.

Finally, in order to compute precision, recall and F_1 score, the number of elements can be calculated either from the number of bytes or from the number of DOM nodes. However, in content extraction the number of bytes is more relevant, hence we use this measure. One could also argue that there could be a large number of irrelevant nodes containing few bytes. Counting nodes would then artificially boost the impact of our method.

5.2 Dataset

There are two fairly well known datasets for the evaluation of content extraction algorithms. Cleaneval evaluation dataset[4], whose documents mostly date to 2006 and L3S-GN1 from the authors of Boilerpipe, which contains Google news articles dating mostly from 2008.

However, documents in these datasets consist of basic HTML files without CSS and Javascript files. Consequently, we created our own dataset. Similarly to L3S-GN1, we used Google News to obtain the first one hundred news articles

[3] `https://boilerpipe-web.appspot.com/`
[4] `https://cleaneval.sigwac.org.uk/`

from three different sources (300 news articles in total): The New York Times[5], The Guardian[6], and Le Devoir[7]. The size of our dataset is comparable to similar datasets as no learning task is required and all documents are used for evaluation (L3S-GN1 has 740 documents and L3S-GN1 has 621 documents).

Since our goal is not to re-evaluate Boilerpipe, we did not proceed to the annotation of the whole corpus but rather identified all false positives resulting from the Boilerpipe tool on our corpus. For annotation we used the CleanEval guidelines[8] for boilerplate removal.

On the corpus on which it has been trained (L3S-GN1), Boilerpipe's articles extractor obtains a precision of 0.9312, a recall of 0.9550, and a F_1 score of 0.9388 [19]. On another well-known corpus (CleanEval), it obtains a precision of 0.9485, a recall of 0.7643, and a F_1 score of 0.8041. Weninger et al. [24] evaluated the precision of Boilerpipe's articles extractor on a corpus of recent websites (2015) of all kinds (i.e. not just news articles). They obtained a precision of 0.8579, a recall of 0.6321, and a F_1 score of 0.7279.

On our corpus, Boilerpipe's articles extractor obtains a precision of 0.8442 on 10696 extracted entities. Boilerpipe's recall is however unknown because we only annotated false positives from retrieved entities. In order to tackle this uncertainty of recall we will compute F_1 values from estimated recall values of 0.65, 0.75, 0.85, and 0.95 and show a consistent improvement of F_1 scores across all these values.

5.3 Experimental Setup

We used the Boilerpipe Java Library[9] with the "ArticleExtractor" strategy, i.e. the best strategy for extracting the main content of news articles. Boilerpipe relies on the first DOM tree instance of web pages and it is therefore necessary to map the nodes of the first DOM tree to the nodes of the rendered DOM tree.

We use XPath expressions in order to localize the nodes and obtain their computed style properties. XPath is the W3C recommended and preferred tool to address nodes of the DOM and has been largely used in WIE systems [11]. Although it is not a difficulty for most WIE systems to associate extracted nodes to XPath expressions, computed style properties are generated according to the nodes of the rendered DOM tree. Consequently, XPath expressions must be valid on the rendered DOM tree. When extracted nodes rely on the first DOM tree, our method requires to map the XPath expressions of extracted nodes (of the first DOM tree) to XPath expressions of the corresponding nodes in the rendered DOM tree.

Instead of modifying Boilerpipe's library in order to obtain the XPath expressions of extracted nodes, we used an already implemented debug function

[5] `https://www.nytimes.com/`
[6] `https://www.theguardian.com/`
[7] `http://www.ledevoir.com/`
[8] `https://cleaneval.sigwac.org.uk/annotation_guidelines.html`
[9] `https://github.com/kohlschutter/boilerpipe`

that shows how Boilerpipe segments the page in different sections. Each section is numbered by Boilerpipe according to its order of appearance in the HTML document. Consequently, it is possible to associate each section to a set of nodes in the rendered DOM tree. After the extraction process the same debug function can be used to get the list of all sections that have been identified as a part of the main content. Consequently, we obtain the set of all XPath expression of extracted nodes by Boilerpipe.

In order to access the rendered DOM tree, we use PhantomJs[10], a well-known headless browser [24]. PhantomJs allows manipulation to rendered web pages through a JavaScript API. Computed style properties are obtained with the *getComputedStyle()* javascript function[11]. As a result, PhantomJs returns the CSS properties of requested DOM nodes and saves them in a CSV file. Then a simple script normalizes all values in the CSV file according to the previously discussed normalization strategy.

Anomaly detection algorithms take this normalized CSV file as an input. For HBOS and k-NN we used Goldstein et al.'s RapidMiner[12] library[14]. An anomaly score is computed for each node and Rapidminer adds this score to the original CSV file.

Finally, extracted nodes are ranked in descending order according to their anomaly score and are successively deleted one by one. Through the deletion process we compute precision, recall, and F_1 scores.

5.4 Results

Figure 2 shows the distribution of Boilerpipes's retrieved elements (true positives in blue and false positives in red) according to their computed visual anomaly scores. The distribution of false positives in function of their anomaly scores for 100-NN, 200-NN, 500-NN, and HBOS is similar, which confirms our hypothesis that visual outliers – i.e. entities with high visual anomaly scores – tend to be false positives (red). It is worth noting that there is a surprisingly high quantity of false positives for the main content extraction task despite a precision of 0.8442. Accordingly, these nodes contain significantly less bytes of information. For example, a subscription form in the core text of an article can generate dozens of nodes while a relevant paragraph is only associated to one node. It remains to be shown to which extent filtering out elements with the HBOS or k-NN algorithms can improve F_1 score.

In order to evaluate our method we remove nodes in descending visual anomaly scores stopping when the F_1 score is maximal.

As shown on the right of Table 1, the number of removed bytes ranges between 2% and 20%, while the number of removed nodes ranges between 5% and 50%. With the exception of 10-NN and 50-NN, which improve precision by less

[10] `http://phantomjs.org/`

[11] Most developer tools included in browsers, such as Firebug for Firefox or Chrome DevTools, allow to access computed style properties of DOM nodes.

[12] `https://rapidminer.com/`

than 10% (see Table 2), the number of removed bytes ranges between 14%-20% and the number of removed nodes ranges between 35%-50%. The extent to which these values would be adequate for other corpora fall outside the scope of this paper and is left to further investigations.

As for the appropriate value of k for the k-NN algorithms, we experimented with different values, showing that precision and F_1 score improve up to around $k = 200$, with more minor gains afterwards (see Table 2).

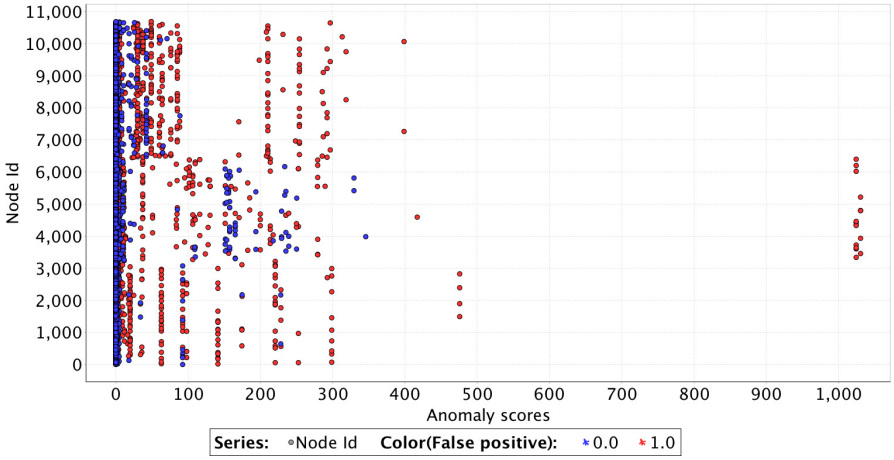

Fig. 2. Boilerpipe's retrieved entities; k-NN anomaly scores for $k = 200$.

As said before, we identified all true and false positives hence we are able to compute exact values for precision (as shown on the left of Table 1). Furthermore, Table 2 shows that our method offers substantial precision improvement (more than 10%) in all cases except 10-NN and 50-NN.

Table 1. F_1 scores and related results for k-NN with different values of k and HBOS.

	Precision	Recall measures								% deleted nodes	% deleted bytes
		0.65		0.75		0.85		0.95			
		R	F_1	R	F_1	R	F_1	R	F_1		
Boilerpipe	0.8442	0.65	0.7345	0.75	0.7943	0.85	0.8471	0.95	0.8940	0	0
10-NN	0.8575	0.6449	0.7361	0.7441	0.7968	0.8433	0.8503	0.9425	08980	4.78%	2.32%
50-NN	0.8786	0.6407	0.7410	0.7392	0.8029	0.8378	0.8577	0.9364	0.9066	10.79%	5.29%
100-NN	0.9389	0.6213	0.7478	0.7169	0.8130	0.8126	0.8712	0.9081	0.9233	34.92%	14.05%
200-NN	0.9883	0.6129	0.7566	0.7072	0.8245	0.8015	0.8852	0.8958	0.9398	48.18%	19.46%
500-NN	0.9963	0.6129	0.7589	0.7072	0.8272	0.8015	0.8883	0.8958	0.9434	52.62%	20.10%
HBOS	0.9450	0.6345	0.7592	0.7321	0.8250	0.8297	0.8836	0.9273	0.9361	30.40%	12.80%

Table 2. Increase in % compared to Boilerpipe.

	Precision	Recall measures							
		0.65		0.75		0.85		0.95	
		R	F_1	R	F_1	R	F_1	R	F_1
10-NN	1.57%	-0.79%	0.22%	-0.79%	0.31%	-0.79%	0.38%	-0.79%	0.45%
50-NN	4.07%	-1.43%	0.89%	-1.43%	1.08%	-1.43%	1.25%	-1.43%	1.41%
100-NN	11.22%	-4.41%	1.81%	-4.41%	2.36%	-4.41%	2.84%	-4.41%	3.27%
200-NN	17.07%	-5.71%	3.01%	-5.71%	3.79%	-5.71%	4.49%	-5.71%	5.12%
500-NN	18.00%	-5.70%	3.33%	-5.70%	4.14%	-5.71%	4.86%	-5.71%	5.52%
HBOS	11.94%	-2.39%	3.37%	-2.39%	3.87%	-2.39%	4.31%	-2.39%	4.71%

However, we don't have exact values for recall. Using estimated recall values of 0.65, 0.75, 0.85, and 0.95 for Boilerpipe, we can compute F_1 scores for our own method. This allows us to show that in all cases except 10-NN, 50-NN and 100-NN, our method improves F_1 score by at least 3%, irrespective of Boilerpipe's estimated recall (see Table 2). While our approach cannot improve recall, this significant improvement in F_1 score shows that the increase in precision largely compensates the decrease of recall. Figure 3 shows how F_1 scores variate across the deletion of visual outliers for 200-NN, 500-NN, and HBOS algorithms and initial recall measures of 0.65 (left) and 0.95 (right).

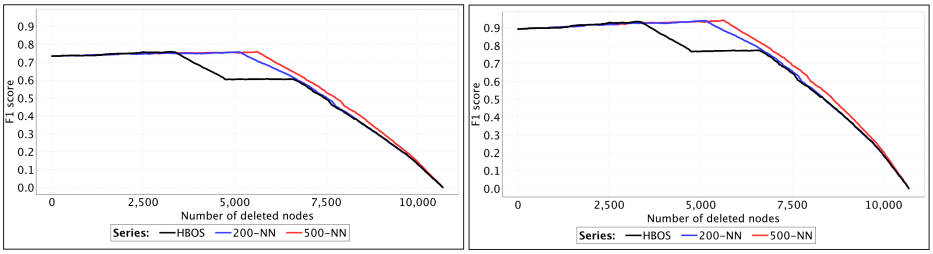

Fig. 3. Variation of F_1 scores across node deletion for 200-NN, 500NN, and HBOS for recall values of 0.65 (left) and 0.95 (right).

As shown in Table 2, the best improvements for precision are obtained with the 200-NN and 500-NN algorithms. While HBOS is outperformed by k-NN, this is at the expense of setting k to considerably high values. This tends to confirm Goldstein and Uchida's recommendation to use HBOS on large datasets [14].

Another interesting fact is that 200-NN and 500-NN must delete considerably more bytes (and nodes) than HBOS in order to reach similar F_1 scores. This is related to the distribution of false positives according to their anomaly score computed by each algorithm (see Figure 2). Although most true positives have a low anomaly score, HBOS gives higher scores to more values than k-NN. From

this point of view, HBOS is more *efficient* as it achieves similar results with less actions.

On our dataset 500-NN has a running time of approximately 10 seconds while HBOS running time is below a second. However, one would expect that on very large datasets HBOS would outperform 500-NN in terms of execution time.

6 Related Works

To our knowledge, we are the first authors to use *visual outliers*, i.e. point anomalies based on visual information for web data extraction. Agyemang et al. [2,3,1] introduced the concept of *web outlier* in data mining. While data mining is the process of discovering patterns in data [25], web outlier mining is defined as "the discovery and analysis of rare and interesting patterns from the web" [2]. Agyemang [1] presents a web content outlier mining framework that uses textual data of web pages in order to find web documents with varying contents from a set of similar documents. The discovery of web outliers from Agyemang's definition still relies on document patterns. Agyemang et al.'s original ideas have been extended in recent web text outlier mining applications [16,17]. Visual information of web documents is however not considered.

Recent works demonstrate the importance of visual information for data extraction. Apostolova et al. [4] evaluated the performance of SVM classifiers on the task of identifying 12 types of named entities in online commercial real estate flyers. They show that the addition of visual features increase their overall F_1 score from 0.83 to 0.87 (4%), and up to 19% for visually salient features. Apostolova et al. give an excellent example of how corpus-specific visual patterns can improve extraction results.

Gogar et al. [13] uses convolutional neural networks in order to create web wrappers that can extract information in non-trivial cases. They propose a method for combining textual and visual information into a single neural net (called *Text Maps*). They evaluate their method on a task of product information extraction and show that the resulting wrapper can extract information on previously unseen websites with an overall accuracy of 93.7%. This method shows how important visual information is for web information extraction. Their results suggest that visual data itself can outperform textual data. Moreover, they show that the combination of both inputs (textual data and visual data) do not have a significant difference from the results achieved with only visual data (at least using convolutional neural networks [21]).

Visual information is also of great interest for anomaly detection, especially when dealing with media resources such as images and videos. Li et al. [22] show that burn injury diagnostic imaging devices can be improved by outlier detection. The deletion of outliers allows to reduce the variance of training data and improve device's accuracy from 63% to 76%. Vu et al. [23] present a unified framework for anomaly detection in video surveillance based on restricted Boltzmann machines (RBM). Their system works directly on image pixels rather than handcrafted features and is unsupervised as it does not require labels. Other examples

include satellite imagery, spectroscopy, medical imagery and video surveillance [6].

There are also works in WIE that use anomaly detection. For example, FluxFlow [26] is a system designed for detecting, exploring, and interpreting anomalous conversational threads in Twitter. It integrates a visualization module that displays anomalous threads and their contextual information with various views in order to facilitate deeper analysis. 239 features are used to compute anomalous threads, which include user profile and user network features (individual level) and temporal and content features (thread level). FluxFlow does not include visual features.

Our motivation to improve state-of-the-art algorithm Boilerpipe has been influenced by a recent meta-analysis on web content extraction algorithms. Weninger et al. [24] evaluate 11 different algorithms for web content extraction on 4 news corpora from 2000 to 2015, each corpus containing 250 websites from 10 news sources spread over 5 year periods (i.e. 2000-2004, 2005-2009, 2010-2014, 2015). The study aims to evaluate how web content extractors are impacted by the changing web in order to make recommendations for future content extraction algorithms. Their results show that Boilerpipe has the best performance on most sets of documents but also underscore a robustness problem in web content extraction. In fact, most of the worst extraction results are obtained on the 2015 corpus. Weninger et al. argue that this is due to web's increasing reliance on external sources for content and data via JavaScript, iframes, etc. They give as an example the fact that the most frequent last-word found by many content extractors on New York Times articles is "loading...". Consequently, they recommend to perform content extraction on the rendered DOM tree (with tools like PhantomJs) in order to manage external scripts. They also suggest that visual information may improve extraction effectiveness, notwithstanding the fact that visual patterns can impact system's robustness. However, for most algorithms, these recommendations would require substantial modifications and may even be inconsistent with existing approaches.

In this paper, we showed that our validation method fulfills Weninger et al.'s recommendations for web content extraction, while maintaining robustness and minimizing integration efforts.

7 Conclusion

In this paper we presented a novel unsupervised validation method that uses visual information of formerly extracted entities in order to eliminate false positive entities and improve extraction results. We introduced the concept of *visual outliers*, i.e. point anomalies based on visual information in web information extraction. We showed that two established anomaly detection algorithms (k-NN and HBOS) can be used in order to identify relevant visual outliers. We applied our method to top state-of-the-art main content extraction algorithm Boilerpipe and showed that visual outliers can be eliminated in order to improve precision and F_1 score. The proposed validation method can be integrated effortlessly

into most WIE systems without impacting on system's flexibility, robustness, and maintenance. Moreover, only computed visual properties of extracted entities are required, which can represent a substantial economy in computation time compared to extraction methods that rely on visual patterns. Future research projects will extend the scope of the proposed method in order to validate its application on large corpora, different extraction tasks, and other WIE methods.

Acknowledgements. The authors gratefully acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

1. Agyemang, M.: Web content outlier mining: motivation, framework, and algorithms. University of Calgary (2006)
2. Agyemang, M., Barker, K., Alhajj, R.: Framework for mining web content outliers. In: Proceedings of the 2004 ACM symposium on Applied computing. pp. 590–594. ACM (2004)
3. Agyemang, M., Barker, K., Alhajj, R.: Web outlier mining: Discovering outliers from web datasets. Intelligent Data Analysis **9**(5), 473–486 (2005)
4. Apostolova, E., Tomuro, N.: Combining visual and textual features for information extraction from online flyers. In: EMNLP. pp. 1924–1929 (2014)
5. Burget, R., Rudolfova, I.: Web page element classification based on visual features. In: Intelligent Information and Database Systems, 2009. ACIIDS 2009. First Asian Conference on. pp. 67–72. IEEE (2009)
6. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM computing surveys (CSUR) **41**(3), 15 (2009)
7. Chang, C.H., Kayed, M., Girgis, M.R., Shaalan, K.F.: A survey of web information extraction systems. IEEE transactions on knowledge and data engineering **18**(10), 1411–1428 (2006)
8. Chenthamarakshan, V., Varadarajan, R., Deshpande, P.M., Krishnapuram, R., Stolze, K.: Wysiwye: An algebra for expressing spatial and textual rules for information extraction. In: International Conference on Web-Age Information Management. pp. 419–433. Springer (2012)
9. Della Penna, G., Magazzeni, D., Orefice, S.: Visual extraction of information from web pages. Journal of Visual Languages & Computing **21**(1), 23–32 (2010)
10. Della Penna, G., Magazzeni, D., Orefice, S.: A spatial relation-based framework to perform visual information extraction. Knowledge and information systems **30**(3), 667 (2012)
11. Ferrara, E., De Meo, P., Fiumara, G., Baumgartner, R.: Web data extraction, applications and techniques: A survey. Knowledge-Based Systems **70**, 301–323 (2014)
12. Gatterbauer, W., Bohunsky, P.: Table extraction using spatial reasoning on the css2 visual box model, proceedings of the 21st national conference on artificial intelligence (2006)
13. Gogar, T., Hubacek, O., Sedivy, J.: Deep neural networks for web page information extraction. In: IFIP International Conference on Artificial Intelligence Applications and Innovations. pp. 154–163. Springer (2016)
14. Goldstein, M., Uchida, S.: A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. PloS one **11**(4), e0152173 (2016)

15. Goldstein, M.B.: Anomaly detection in large datasets. Verlag Dr. Hut (2014)
16. Huosong, X., Zhaoyan, F., Liuyan, P.: Chinese web text outlier mining based on domain knowledge. In: Intelligent Systems (GCIS), 2010 Second WRI Global Congress on. vol. 2, pp. 73–77. IEEE (2010)
17. Khan, M.R.R., Ahmed, M.I., Riyad, M.A.: A novel analytical approach for identifying outliers from web documents. International Journal of Applied Engineering Research **12**(22), 12156–12161 (2017)
18. Kohlschütter, C., Fankhauser, P., Nejdl, W.: Boilerplate detection using shallow text features. In: Proceedings of the third ACM international conference on Web search and data mining. pp. 441–450. ACM (2010)
19. Kovacic, T.: Evaluating web content extraction algorithms. University of Ljubljana (2012)
20. Krüpl-Sypien, B., Fayzrakhmanov, R.R., Holzinger, W., Panzenböck, M., Baumgartner, R.: A versatile model for web page representation, information extraction and content re-packaging. In: Proceedings of the 11th ACM symposium on Document engineering. pp. 129–138. ACM (2011)
21. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature **521**(7553), 436 (2015)
22. Li, W., Mo, W., Zhang, X., Lu, Y., Squiers, J.J., Sellke, E.W., Fan, W., DiMaio, J.M., Thatcher, J.E.: Burn injury diagnostic imaging devices accuracy improved by outlier detection and removal. In: SPIE Defense+ Security. pp. 947206–947206. International Society for Optics and Photonics (2015)
23. Vu, H., Nguyen, T.D., Travers, A., Venkatesh, S., Phung, D.: Energy-based localized anomaly detection in video surveillance. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 641–653. Springer (2017)
24. Weninger, T., Palacios, R., Crescenzi, V., Gottron, T., Merialdo, P.: Web content extraction: a metaanalysis of its past and thoughts on its future. ACM SIGKDD Explorations Newsletter **17**(2), 17–23 (2016)
25. Witten, I.H., Frank, E., Hall, M.A., Pal, C.J.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann (2016)
26. Zhao, J., Cao, N., Wen, Z., Song, Y., Lin, Y.R., Collins, C.: # fluxflow: Visual analysis of anomalous information spreading on social media. IEEE Transactions on Visualization and Computer Graphics **20**(12), 1773–1782 (2014)