Knowl Inf Syst (2018) 54:711–776
https://doi.org/10.1007/s10115-017-1097-2

REGULAR PAPER

The BigGrams: the semi-supervised information extraction system from HTML: an improvement in the wrapper induction

Marcin Michał Mirończuk[1]

Received: 2 August 2016 / Revised: 2 July 2017 / Accepted: 4 August 2017 /
Published online: 20 August 2017

Abstract The aim of this study is to propose an information extraction system, called BigGrams, which is able to retrieve relevant and structural information (relevant phrases, keywords) from semi-structural web pages, i.e. HTML documents. For this purpose, a novel semi-supervised wrappers induction algorithm has been developed and embedded in the BigGrams system. The wrappers induction algorithm utilizes a formal concept analysis to induce information extraction patterns. Also, in this article, the author (1) presents the impact of the configuration of the information extraction system components on information extraction results and (2) tests the boosting mode of this system. Based on empirical research, the author established that the proposed taxonomy of seeds and the HTML tags level analysis, with appropriate pre-processing, improve information extraction results. Also, the boosting mode works well when certain requirements are met, i.e. when well-diversified input data are ensured.

Keywords Information extraction · Information extraction system · Wrappers induction · Semi-supervised wrappers induction · Information extraction from HTML

1 Introduction

The information extraction (IE) was defined by Moens [48] as the identification, and consequent or concurrent classification and structuring into semantic classes, of specific information found in unstructured data sources, such as natural language text, making information more suitable for information processing tasks. The task of IE is to retrieve important information concerning named entities, time relations, noun phrases, semantic roles, or relations among entities from the text [48]. Often, this process consists of two steps: (1) finding the extraction patterns and (2) extraction of information with use of these patterns. There

✉ Marcin Michał Mirończuk
m.marcinmichal@gmail.com

[1] Department of Artificial Intelligence, Institute of Computer Science Polish Academy of Sciences, Jana Kazimierza 5, Warsaw, Poland

are three levels of the text structure degree [11]: free natural language text (free text, e.g. newspapers, books), semi-structured data in the XML or HTML format or fully structured data, e.g. databases. The literature sometimes considered semi-structured data, like HTML, as a container of free natural language text. There are many methods for pattern creation [11,63,74], e.g. manual or with use of machine learning techniques.

This study briefly presents a general framework of an information extraction system (IES) and its implementation—the BigGrams system. Moreover, it describes (1) a novel proposal of the semi-supervised wrappers induction (WI) algorithm that utilizes the whole Internet domain (website, site, domain's web pages) and creates a pattern to extract information in context with the entire Internet domain and (2) a novel taxonomic approach and its impact to the semi-supervised WI. This system and the WI are developed to support the information retrieval system called NEKST [17]. The NEKST utilizes the structured results coming from the BigGrams system to improve query suggestions and a ranking algorithm of web pages. Thanks to the BigGrams system, the relevant phrases (keywords) are extracted for each Internet domain.

The BigGrams system analyses HTML web pages to recognize and extract values of a single or multiple attributes of an information system (IS) [52] about, for example, films, cars, actors, pop stars, etc. On the other hand, the main aim of the WI is to create a set of patterns. These patterns are matched to the data, and in this way, new information is extracted and stored in the created IS. The proposed novel WI is based on application of formal concept analysis (FCA) [54,77] to create extraction patterns in a semi-supervised manner. Thanks to FCA, the hierarchy of chars sequence groups is created. These groups cover the selected parts of the Internet domains. Based on this hierarchy, the proposed WI algorithm (1) selects the appropriate groups from the hierarchy, i.e. the groups that sufficiently cover and generalize the domain's web pages, and (2) based on these groups, creates patterns that often occur in the Internet domain. The BigGrams system subsequently uses these patterns to extract information from semi-structured text documents (HTML documents). The proposed semi-supervised WI approach consists of the following steps: (1) the user defines a reference input set or a taxonomy of correct instances called *seeds* (values of the attributes of an IS), (2) the algorithm uses *seeds* to build the extraction patterns, and (3) the patterns are subsequently used to extract the new instances to extend the seed set [18,73,74]. For example, the input set of an actor name *Brad Pitt*, *Tom Hanks* can be next extended by the new instances, like a *Bruce Willis*, *David Duchovny*, *Matt Damon*.

The author did not find, in the available literature, similar methods (like the proposed BigGrams) to realize the deep semi-supervised approach to extract information from given websites. There are the shallow semi-supervised methods, such as Dual Iterative Pattern Relation Extraction (DIPRE) technique [5] and Set Expander For Any Language (SEAL) system [18,73,74] that obtain information from Internet web pages. These approaches use horizontal (shallow) scan and Internet web pages processing in order to extract appropriate new seeds and create an expanded global set of seeds. The aim of these systems is to expand the set of seeds for new seeds in the same category. In the end, these systems evaluate the global results, i.e. the quality of the extended global set of seeds. The proposed deep semi-supervised approach, as opposed to the shallow semi-supervised method, is based on the vertical (deeply) scans and processing of the entire Internet websites to extract information (relevant instances from these sites) and create an expanded local set of new seeds. In this approach, the number of proper seeds obtained from given websites is evaluated. In this article, the author shows empirically that shallow semi-supervised approach (SEAL is established as a baseline method) is inadequate to resolve the problem of deep semi-supervised extraction, i.e. the information extraction focuses on the websites. The shallow approaches cannot create all required and correct patterns to extract all important and relevant new instances from given sites.

The main objectives of this study are as follows:

- establish the good start point to explore IES and the proposed BigGrams system through the theoretical and practical description of the above systems,
- briefly describe the novel WI algorithm with the use case and theoretical preliminaries,
- establish the impact of the (1) input form (the seeds set and the taxonomy of seeds), (2) pre-processing domain's web pages, (3) matching techniques, and (4) a level of HTML documents representation to the WI algorithm results,
- find the best combination of the elements mentioned above to achieve the best results of the WI algorithm,
- check what kind of requirements must be satisfied to use the proposed WI in an iterative way, i.e. the boosting mode, where the output results are provided to the system input.

The author has determined (based on empirical research) the best combination and impact of the above-mentioned core information extraction elements to information extraction results. The conducted research shows that the best results are achieved when the proposed taxonomy approach is used to represent the input seeds and the pre-processing technique, which clears the values of HTML attributes, where the seeds are matched only between HTML tags, and if we use the tags level, rather than the chars level representation of HTML documents. Thanks to these findings, we can construct better WI algorithms producing better results. The proposed system and the WI method have been compared with the baseline SEAL system. Furthermore, the results of the conducted experiments show that we can use the output data (extracted information) as input data of the BigGrams system. It allows the system (when we can ensure well-diversified input data) to be used in an iterative manner.

The presented study is well grounded theoretically to give an in-depth understanding of the proposed method as well as to be easily reproduced. The paper is structured as follows. Section 2 describes various known IE systems. Section 3 presents the formal description of the IES, i.e. it contains the description of IS and the general framework of the IES. Section 4 describes the implementation of the system mentioned above. This section contains (1) the comparison of the baseline SEAL and the proposed BigGrams system, (2) the specification of the proposed BigGrams IES, and (3) the WI algorithm together with the historical and mathematical description of FCA background, and a case study. The experimental results are presented in Sect. 5. Finally, Sect. 6 concludes the findings.

2 State of the art and related work

Currently, there are numerous well-known reviews describing the IESs [53,66]. Usually they focus on free text or semi-structured text documents [4,48,58]. Optionally, the reviewers describe one of the selected components such as WI [11]. There are also many existing IESs. Typically, they are based on a *distributional hypothesis* (*"Words that occur in the same contexts tend to have similar meanings"*), and they use formal computational approach [30,38]. Researchers also constructed another hypothesis called *KnowItAll Hypothesis*. According to this hypothesis, *"Extractions drawn more frequently from distinct sentences in a corpus are more likely to be correct"*. [21]. IESs, such as Never-Ending Language Learner (NELL), Know It All, TextRunner, or Snowball represent this approach [1,3,6,9,10,22,23,56,59,68,78]. The systems mentioned above represent the trend called *open IE*. They extract information from semi-structured text (HTML documents considered to be containers of natural language text) or natural language text. Also, there are solutions that attempt to induce ontologies from natural language text [20,37,49]. The examples of IE for semi-structured texts are described in [26,32,34,50,55,61,76]. In the case of databases [11], the IE can be viewed as an element of data mining and knowledge discovery [45]. There are also many algorithms that implement the WI component of IES [11].

Schulz et al. [60] present the newest survey of the web data extraction aspects. Their paper describes and complements the most recent survey papers of authors like Ferrara et al. [24] or Sleiman and Corchuelo [61]. Furthermore, we can add three articles of Varlamov and Turdakov [72], Umamageswari and Kalpana [71], and Chiticariu et al. [13] as a complement to Schulz et al. survey. In these studies, the authors focus on the description of the methods, vendors, and products to IE or WI. Furthermore, all the papers mentioned above describe the IE problem using different perspectives, such as the level of human intervention, limitations, wrapper types, wrapper scope. In the author's research point of view, the best division of the IE approaches is based on the techniques used to learn WI component. We may distinguish three techniques: supervised, semi-supervised, and unsupervised. The supervised methods require manual effort of the user, i.e. the user must devote some time to label the web pages and mark the information to extraction [34–36,64,69]. The unsupervised methods, on the other hand, start from one or more unlabelled web documents and try to create patterns that extract as much prospective information as possible, and then the user gathers the relevant information from the results (Definition taken from Sleiman [64]) [12,15,39,43,62,63,65].

The semi-supervised technique is an intermediate form between supervised and unsupervised methods. In this approach, we only create a small dataset of seeds (a few values of the IS attribute) rather than create data set of labelled pages. There are three well-known IESs that are based on the semi-supervised approach to WI and processed on the web pages, i.e. Dual Iterative Pattern Relation Extraction (DIPRE) technique [5], Set Expander For Any Language (SEAL) system [18,73,74], and similar to SEAL the Set Expansion by Iterative Similarity Aggregation (SEISA) [31]. There are several advantages of these approaches, namely (1) they are language independent, (2) they can expand a set of small input instances (seeds) in an iterative way with sufficient precision, and (3) they discover the patterns with almost no human intervention. The SEAL represents a more general approach than DIPRE. The DIPRE extracts only information about books (title and author names). The SEAL can extract unary (e.g. *actor name(Bruce Willis)*) and binary (e.g. *born-in(Bruce Willis, Idar-Oberstein)*) relations from the HTML documents. In the first case, the extracted instance would be *Bruce Willis*, in the second *Bruce Willis/Idar-Oberstein*. Due to the more general form of the SEAL, as compared to the DIPRE, and thanks to its ability to reproduce, as compared to the SEISA, the author decided to compare the BigGrams system against the SEAL as a baseline.

Finally, it is worth mentioning one of the few obstacles that relate to the available, well-labelled and large gold-standard data sets and tools to IE [60]. The author can confirm the observation of the Schulz et al. [60] that it is difficult to compare the results of the IE solutions. The promising changes in this area are the large and well-labelled data sets created by the Bronzi et al. [7], Hao et al. [28] (this set was used in the additional benchmark, See "Appendix A"), as well as the original data set which was created by the author (see Sect. 5.1).

3 Formal description of the information extraction system

Usually, IES uses data that are received or have been transformed from the input of the IS. Also, the semi-supervised WI algorithms use information from some kind of the IS. For this reason, the author assumes that it is important to formally define the term IS to better understand the rest of this article and its role in IES. Sect. 3.1 describes theoretical basis of IS with the technical details. Section 3.2 explains the general framework of the IES.

3.1 Theoretical preliminaries

According to Pawlak [52] in each IS, there can be identified a finite set of objects X and finite set of attributes A. Each attribute a belonging to the A is related to its values collection V_a, which is also known as a *domain* attribute a. It is accepted that the domain of each attribute is at least a two-element, i.e. each attribute may take at least one of the two possible values. Clearly, some attributes may have common values, e.g. for the attribute *length* and *width* set of values are real numbers. The binary function ϱ is introduced to describe the properties of the system objects. This function assigns the value v belonging to the domain V_a for each object $x \in X$ and attribute $a \in A$. By *information system* is meant quadruple:

$$IS = <X, A, V, \varrho> \tag{1}$$

where X, a non-empty and finite set of objects; A, a non-empty and finite set of attributes; $V = \bigcup_{a \in A} V_a$, V_a domain attribute a, a set of values of the attribute a; ϱ, the entire function, $\varrho : X \times A \rightarrow V$, wherein $\varrho(x, a) \in V_a$ for each $x \in X$ and $a \in A$.

The *domain* V_a attribute a in IS is a set V_a described as follows:

$$V_a = \{v \in V\colon \textit{for each exist } x \in X,\ \textit{such as } \varrho(x, a) = v\} \tag{2}$$

3.1.1 Practical preliminaries

The name of the IS (films, cars, etc.) is usually a general concept that aggregates a combination of attributes and their values. For example, the name *movies* is a concept that may contain attributes like the film title, actor's name and production year.

In the remainder of this article, the author used a shortened notation to describe the IS, i.e. we can treat the IS as an n-tuple of attributes and their values: $IS{<}X = \{x_1, \ldots, x_{|X|}\}$, $\textit{attribute-name-}1 = \{\textit{value } 1 \textit{ of attribute } 1,\ \textit{value } 2 \textit{ ofattribute } 1, \ldots\}, \ldots, a \in A = V_a{>}$.

In the rest of this article, the following types of a tuple will be used: monad (singleton) and n-tuple. The monad tuple is an IS having one attribute $|A| = 1$, $IS{<}a \in A = V_a{>}$ and $\textit{is-a}(V_a, a \in A)$ or $[a \in A](V_a)$. For example, *IS<film title = {die hard, x files,…}>*, *IS<actor name = {bruce willis, david duchovny, …}>* and *is-a(die hard, film title)* or *film title(die hard)*. N-tuple is an IS with n-attributes $|A| > 1$.

Finally, describing the IS, it is worth noting that the attributes can be granulated. The attribute values can be generalized or closely specified so that attribute taxonomies can be built. Assume the attribute $a \in A$ and a set of its values V_a. This attribute can be decomposed into a_1 and a_2 such that $V_{a_1} \cap V_{a_2} = \emptyset$. Then, it is possible to connect the attributes' values $V_{a_1} \cup V_{a_2} = V_a$. The first action is defined as specification of attribute values. The second action is defined as generalization of attribute values. For example, the attribute *film title* can be split into two separate attributes *film-tile-pl* (Polish film title) and *film-title-en* (English film title), which then can be re-connected to receive a *film title* set of attribute values. Of course, it is not a general rule that $V_{a_1} \cap V_{a_2} = \emptyset$; for example, the attribute *person name* can be split into *musician name* and *actor name* attributes. And it is obvious that there are actors who are musicians and vice versa, for example *Will Smith*. However, the assumption on the input data set that $V_{a_1} \cap V_{a_2} = \emptyset$ has a positive effect on the results obtained from the proposed semi-supervised method of IE.

3.2 The general framework of the information extraction system

The author considered the whole process of an IE. This section describes all the components of an IES, regardless of the analysed data structure. It was assumed that an HTML document can be treated as a structure that stores free text (*<p>free text</p>*) or provides hidden semantic information. The HTML tag layouts (in short, HTML layouts) define this information (*<h1 style="actor name">Bruce Willis</h1>*). Natural language processing (NLP) algorithms are used in the first case to process free text. These tools are used to locate the end of a sentence and to grammatically analyse the sentences, etc. In the second case, the WI algorithms are used to analyse the structure of the HTML tag layouts. The created wrappers extract relevant information from these layouts. Figure 1 shows the basic components of the IES.

Figure 1 shows the IES pipeline. We can consider the task of the IE as a realization of reverse engineering. Usually, we try to restore a full or partial model of an IS based on free text or HTML tag layouts. We can divide this task into two subtasks. The first subtask relates to the creation of an information system scheme (defining the IS attributes and possible relationships between them). While the second subtask relates to the attribute values extraction of the created IS schema. The IS schema that contains the attributes and values assigned to them is in short called IS.

The presented process in Fig. 1 gets an input data set, which is a collection of documents (corpus) P. It contains HTML documents $p \in P$, which belong to a domain d. The domain comes from the set of domains D ($d \in D$). The unknown domain process $process_d$ has created these documents. The process connects information from an unknown IS_d with the HTML layout L_d and noise T_d. The HTML layout defines a presentation layer. The presentation layer displays information from the hidden IS_d. Noise is created by the dynamic and random elements generated in the presentation layer. For example, the domain of movies can contain a simple IS, which consists of the following attributes and their values IS_d *<film title = {x files,x files}, actors = {david duchovny, gila anderson}, comments = {comments body 1, comments body 2}>*. The HTML layout might look like *<h1 class="film title">\$film title</h1><br/> Actors<ul><li>\$actors*$_1$ *</li>< li>\$actors*$_2$ *</li></ul> ul>* $\cdots$ *<div class = "todays comments">random(comments)</div>*. Noise, in this case, is generated by the $random()$ function, which returns a random comment. An output document from this process will have the following form *<h1 class = 'film title'>x files</ h1><br/>Actors<ul>< li>david duchovny</li><li>gillian anderson</li></ul ><*ul*> …<div class = "todays comments'>comments body 2</div>*. It should be noted that the same information can be expressed using a free text embedded in HTML layout *<p>The film's title is "The X-Files". In this film, starring actors are david duchovny and gillian anderson. Selected a random comment at the premiere, which I heard was as follows: description of the body 2.</p>*.

In Fig. 1 an *Induction of the information system schema* component creates the IS schema. The schema contains attribute names and relationship between them. A software engineer who creates the IS can induce this schema based on manual analysis. The software engineer may also use other IES components to induce attribute names and relationships between them. Thanks to these components, we may extract the attribute names and values as well as relationship between attributes.

In Fig. 1 an *Identification of parts of the content of web pages to the information extraction* component is used to mark important sequences of HTML tags or text in a document.

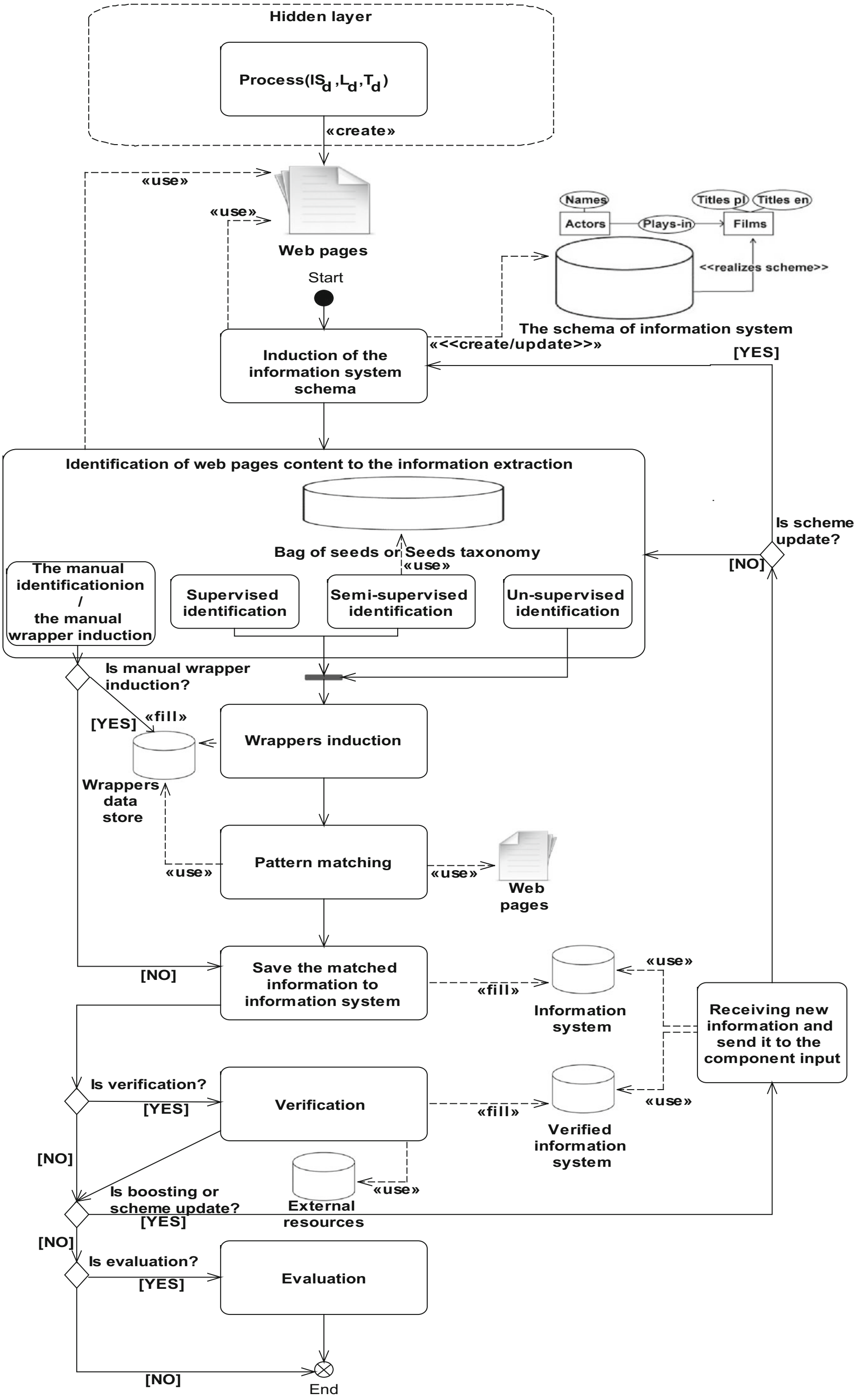
Hidden layer
Process(IS_d, L_d, T_d)
«create»
«use»
«use»
Web pages
Start
Names
Actors
Plays-in
Titles pl
Titles en
Films
<<realizes scheme>>
The schema of information system
«<<create/update>>»
[YES]
Induction of the information system schema
Identification of web pages content to the information extraction
Bag of seeds or Seeds taxonomy
«use»
Is scheme update?
[NO]
The manual identificationion / the manual wrapper induction
Supervised identification
Semi-supervised identification
Un-supervised identification
Is manual wrapper induction?
[YES]
«fill»
Wrappers induction
Wrappers data store
Pattern matching
«use»
«use»
Web pages
[NO]
Save the matched information to information system
«fill»
Information system
«use»
Receiving new information and send it to the component input
Is verification?
[YES]
Verification
«fill»
Verified information system
«use»
[NO]
Is boosting or scheme update?
[YES]
External resources
«use»
[NO]
Is evaluation?
[YES]
Evaluation
[NO]
End

Fig. 1 The information extraction system pipeline

The WI algorithm creates patterns based on these markings. We may mark documents using the supervised, semi-supervised, or unsupervised methods. Eventually, while viewing documents, we may manually identify important parts of the documents and immediately create the patterns (brute force method) or directly save them to the IS. Thanks to this, we can omit the WI component and we can go directly to the *Pattern matching* or *Save the matched information to information system* component. The supervised method is also based on manual marking. In this method, we can also mark important parts of documents. However, we do not create the patterns and after that we do not omit the WI component. The semi-supervised way involves creation of an input set of *seeds* (the attribute values of the IS). This set does not necessarily depend on the marked documents. However, this set must contain the seeds that can be matched to the analysed documents. It is the necessary condition for the WI. In the unsupervised identification, the same algorithm identifies important elements of the document. After marking the whole or some parts of the documents, we can go to the *Wrappers induction* component.

In Fig. 1 the *Wrappers induction* component is used to create the patterns. Based on the marked sequences of documents from the *Identification of the parts of the content of web pages to the information extraction* the WI algorithm creates the patterns. Next, the created patterns are saved in the data buffer (a memory, a database, etc.).

In Fig. 1 the *Pattern matching* component is used to match the created patterns. This component takes the patterns from the data buffer. After that, a matching algorithm matches these patterns to other documents. In this way, the component extracts new attribute values. The *Save the matched information to information system* component saves these new attribute values into the IS. Depending on the type of an analysis, i.e. induction of the attribute names or relations names of the IS, or extraction of the attribute values, information can be stored in an auxiliary or destination IS, which has the established IS schema. Of course, we may perform a manual identification of important information while viewing documents, and save this information directly to the selected IS (the *Wrappers induction* and the *Pattern matching* components are skipped).

In Fig. 1 the *Verification* component is optional. We may use it to validate the extracted attribute values, attribute names, or relation names. Such verification of facts may be based on external data sources, e.g. an external corpus of documents [8,16].

In Fig. 1 the *boosting* phase is an optional element. Extracted information (verified or not), depending on the type of analysis, can be redirected to the *Induction schema of the information system* or *Identification of the parts of the content of web pages to the information extraction* component.

In Fig. 1 the last *Evaluation* component is optional. We may use this component to verify the entire process or individual components. For example, we may evaluate the WI algorithm or the component to verify facts collected in the IS, etc.

4 BigGrams as the implementation of the information extraction system

Section 4.1 describes the comparison of the BigGrams and the SEAL systems. Section 4.2 explains the specification of the proposed IES. Section 4.3 describes the algorithm and its use case.

4.1 The comparison of BigGrams and SEAL systems

The BigGrams system is able to extract unary and binary relations, and it is partially similar to the SEAL. The main differences between the BigGrams and the SEAL are as follows. The data structure used in the BigGrams is a lattice instead of Trie data structure utilized in the SEAL. Also, the BigGrams system uses a different WI algorithm. The term *lattice* comes from FCA, which is also applied in many other disciplines, such as [77] psychology, sociology, anthropology, medicine, biology, linguistics, mathematics, etc. The author did not find approaches based on mathematical models called FCA to build the WI algorithm from HTML documents in the available literature. Another difference concerns the method of document analysis. In the SEAL, the WI algorithm creates the patterns on the level of a single page and a chars level. In the BigGrams, the WI algorithm recognizes the whole domain of documents as one huge document. Based on this document, the BigGrams creates a set of patterns and attempts to extract all important and relevant instances from this domain (high precision and recall inside Internet domain). In contrast, the SEAL retrieves instances by using every single HTML document from the whole Internet and attempts to achieve high precision. The SEAL also uses a rank function to rank the extracted instances. This function filters, for example, the noise instances. The BigGrams does not use any ranking function. Furthermore, from the point of view of the extraction task (the extraction of relevant instances from domains), the SEAL is not the appropriate tool to accomplish this task because of low recall.

Like in the SEAL, the WI algorithm of the BigGrams can use a sequence of characters (*raw chars, chars level, raw strings, strings level, or strings granularity*) [47]. This algorithm extracts these strings from the HTML document and uses them to create patterns. However, the author noticed that it is better to change these *raw strings* to the HTML tags level (HTML tags granularity). This article presents the results of this change.

In contrast to the SEAL, the WI algorithm of the BigGrams could use a more complex structure to the WI. The BigGrams may use the *taxonomy of seeds* rather than a simple set of seeds (*a bag of seeds*). The *bag of seeds* contains input instances (seeds) without semantic diversity. The taxonomy includes this semantic diversity.

Furthermore, the author introduced a weak assumption that based on the input values of the IS, the extracted patterns will extract new values belonging to the attribute of a given IS. This is a weak assumption because the created pattern can extract a value belonging to another IS. It occurs when based on values from IS_1 and values from IS_2 (disjunction values set), the WI algorithm will create the same pattern that covers values from IS_1 and IS_2. In the algorithm output, it cannot be recognized which values belong to which IS. Despite this drawback, this approach significantly improves the performance of the proposed WI algorithm. It has been proven experimentally and described in this article.

Finally, it is worth mentioning that the BigGrams, such as the SEAL, does not operate in "live DOM" where all CSS (e.g. CSS boxes) and JavaScript are applied to a page. Moreover, also the dynamic elements of HTML 5 are omitted in the WI phase. The BigGrams processes only the static rendered web pages.

4.2 Specification on high level of abstraction

The aim of the BigGram system, for a given particular domain, is to extract only information (new seeds, new values of attributes) connected with this domain. For example, for a domain connected with movies, the BigGrams should extract the actors' names, film titles, etc. To this end, the BigGrams analyses all the pages $p \in P$ from the domain $d \in D$. Based on

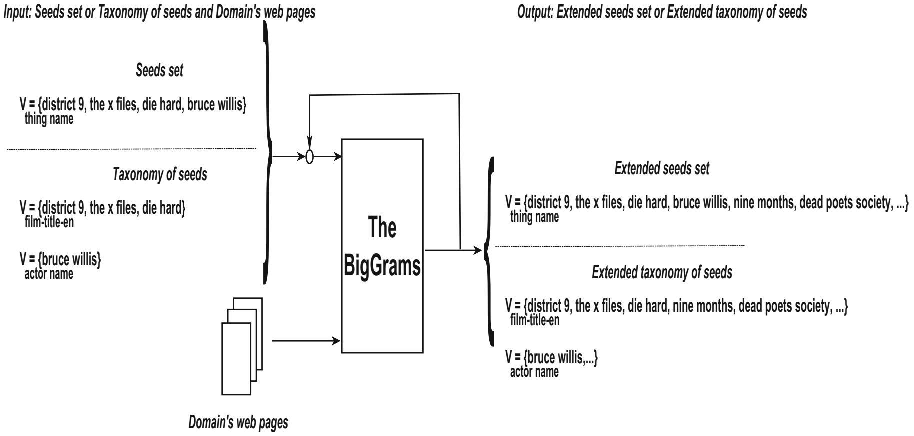
Input: Seeds set or Taxonomy of seeds and Domain's web pages
Seeds set
V = {district 9, the x files, die hard, bruce willis}
thing name
Taxonomy of seeds
V = {district 9, the x files, die hard}
film-title-en
V = {bruce willis}
actor name
The BigGrams
Domain's web pages
Output: Extended seeds set or Extended taxonomy of seeds
Extended seeds set
V = {district 9, the x files, die hard, bruce willis, nine months, dead poets society, ...}
thing name
Extended taxonomy of seeds
V = {district 9, the x files, die hard, nine months, dead poets society, ...}
film-title-en
V = {bruce willis,...}
actor name

Fig. 2 The general scheme of the BigGrams system

this analysis, the BigGrams creates patterns for the whole domain and extracts new seeds. Figure 2 shows the general scheme of the BigGrams system.

The input of the BigGrams system accepts two data sets (Fig. 2). The first data set may include a set of seeds (*bag of seeds*) or a taxonomy of seeds. The second data set contains the domain's web pages (three examples of web pages are shown in Figs. 4, 5, and 6). The *bag of seeds* contains input instances (seeds). Alternatively, all values (instances) are assigned to one attribute (*thing name*) of a singleton IS. We may split this attribute into several attributes that may store values that are more relevant to them (by a semantic view), i.e. we can create a taxonomy of seeds. The values are assigned to semantically appropriate attribute names. Let us consider the *the bag of seeds* in the following form *IS<(thing name = {district 9, the x files, die hard, bruce willis})*. The thing name attribute could be split into more specific attributes, such as *english film title* and *actor name*. In this way, we can create separable input data sets (singleton ISs). All the IS contains specific attributes that capture the well meaning (semantics) of their values. For *bag of seeds*, mentioned above, we can create separable ISs such as, *IS<english film title = {district 9, the x files, die hard}>* and *IS<actor name = {bruce willis}>*. The output of the BigGrams system contains an extended input set. The system realizes the following steps: (1) creates patterns based on the input pages and seeds, (2) extracts new instances from these pages by using the patterns, and (3) adds new instances to the appropriate data set. Furthermore, the system may work in the batch mode or the boosting mode. The batch mode does not operate in an iterative way and does not use the output results (the newly extracted seeds) to extend input seeds. The boosting mode includes these abilities.

4.2.1 Specification details with examples

Figure 3 presents the more elaborate scheme of the BigGrams system.

The process presented in Fig. 3 shows the basic steps of the BigGrams system. Firstly, a set of IS schemes has to be created manually. Each IS schema consists of common attributes, such as *id-domain* (a domain identifier), *id-webpage* (a web page identifier). Also, each IS schema contains an individual attribute (not shared/common between IS schemas), such as

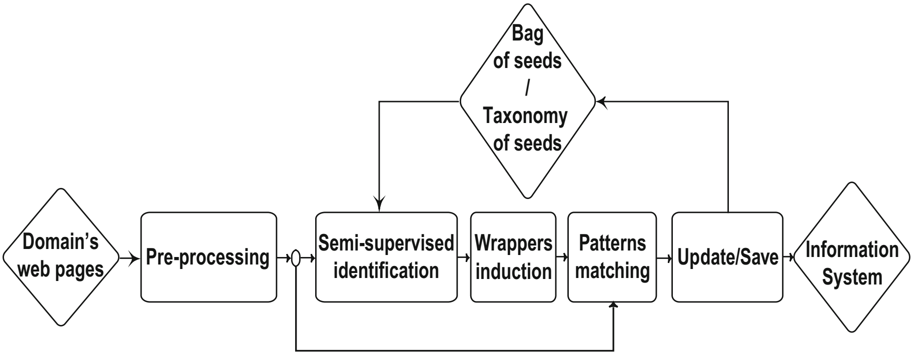

Fig. 3 The BigGrams' pipeline

<html> ... <body> ... <ul><li class=”film title”><br/>**the x files**</li><p> ... </p><li> ... </li><p class=”film title”><br/>**die hard**<br/></p> ... <ul><li class=”film title”><br/>**the avengers**</li><p> ... </p></ul> ... </body></html>

Fig. 4 Contents of document p_1

<html> ... <body> ... <ul><li class=”film title”><br/>**the x files**</li><p> ... </p><li> ... </li><p class=”film title”><br/>**district 9**<br/></p> ... <ul><li class=”film title”><br/>**die hard**</li><p> ... </p><ul> ... </li><p class=”film title”><br/>nine months<br/></p> ... <ul><li class=”film title”><br/>dead poets society<br/><p> ... </ul> ... </body></html>

Fig. 5 Contents of document p_2

<html> ... <body> ... <ul><li> ... <ul><li class=”film title”><br/>**the avengers**</li><p> ... <li> ... <ul><li class=”film title”><br/>good will hunting</li><p> ... <li> ... <ul><li class=”film title”><br/>the departed</li><p> ... <li> ... </li><p class=”film title”><br/>dead poets society<br/></p> ... </ul> ... </body></html>

Fig. 6 Contents of document p_3

film-title-pl, *film-title-en*, *actor name*. After the phase of patterns matching, the acquired information is saved in particular IS schemas.

In the second step, a collection of documents for each domain (*Domain's web pages*) is gathered from a distributed database (DDB). After this step, an initial document processing (*Pre-processing*) takes place. This processing:

- fixes the structure of an HTML document (closes the HTML tags, closes the attribute values using the chars ”, etc.),
- cleans an HTML document from unnecessary elements (header, JavaScript, css, comments, footers, etc.),
- changes the level of *granularity of HTML tags*.

The HTML tags contain attributes and their values, for example *<h1 attribute1 = “value1” attribute2 = “value2”>*. We may change the *granularity of HTML tags* by removing the value of the attributes or by removing the attributes and their values. For example, we can express the above-mentioned HTML tags as:

- *<h1 attribute1 = “ attribute2 = ”>* - the HTML tags without attribute values,

<ul><li class="film title"><br/>**the x files**</li><p></li><p class="film title"><br/>**die hard**<br/></p><ul><li class="film title"><br/>**the avengers**</li><p><ul><li class="film title"><br/>**the x files**</li><p></li><p class="film title"><br/>**district 9**<br/></p><ul><li class="film title"><br/>**die hard**</li><p><ul><li class="film title"><br/>**the avengers**</li><p>

Fig. 7 Contents of document p_4

- *<h1>* - the HTML tags without attributes and their values.

The *semi-supervised identification* performs matching the seeds (the attribute values of a singleton IS) for each processed HTML document. We create the input set of seeds for each created scheme of a singleton IS manually. After matching the seed to the document, the *n*-left and *m*-right HTML tags that surround the matched seed are collected. Based on these, we can create a set of triple data t_d *<n-left HTML tags, matched seed, m-right HTML tags>*. Based on this set, the algorithm creates one global *big document*. This document can be considered as a collection of aforementioned data triples t_d.

The *Wrappers induction* step contains the embedded element called *Pre-processing of wrappers induction*. This element processes the HTML documents before performing wrappers induction. This component is responsible for *outlier seeds detection and filtration*. It detects and removes the outlier data triples t_d from the *big document*. The previous experiments [47] have shown that t_d can be found in the data set that contains the seeds that contribute to the induction of patterns in a negative way. Usually, there are seeds and t_d that occur on the domain's HTML document too often or too rarely. This component removes too frequent or too rare seeds from *big document*, i.e. seeds of frequency below $Q3 - 1.5 \cdot (IQR)$ or above $Q1 + 1.5 \cdot (IQR)$ ($Q1$—first quartile, $Q3$—third quartile, IQR—interquartile range). Also, the approach of sampling only k random t_d from the *big document* is used, if it is too long, i.e. when it contains more than p triple data t_d. After the *outlier seeds detection and filtration* the WI algorithm creates the patterns based on the one global *big document*. The author defines a *pattern* as a pair which contains the left l and the right r contextual HTML tags.

The tags that surround the matched seeds from the triple data are defined as a left extension or a right extension. Respectively, the left and right extensions have fixed lengths. The length is expressed by the number of HTML tokens. For example, we may assume the input as the IS singleton *IS<film-title-en = {the x files, die hard, the avengers, district 9}>*. The input seeds set $V_{a=film\text{-}title\text{-}en}$ consists of four seeds (values of the attribute *film-title-en*) $|V_{a=film\text{-}title\text{-}en}| = 4$ and $V_{a=film\text{-}title\text{-}en} = \{\textit{the x files}, \textit{die hard}, \textit{the avengers}, \textit{district } 9\}$. Also, it is assumed that the domain $d \in D$ is represented by a set P of three documents $P = \{p_1, p_2, p_3\}$. The contents of the three pages are shown in Figs. 4, 5, and 6, respectively.

We can match the seeds $s \in V_{a=film\text{-}title\text{-}en}$ to the documents $p_1 - p_3$, and in this way we can retrieve the set of data triples t_d. Next, we create a *big document* that connects the all triple data t_d. The HTML tokens of the triple data have the fixed left k_l and the right k_r lengths. Figure 7 presents the *big document* for the $k_l = 3$ *and* $k_r = 2$ lengths.

After creation of the *big document*, each seed from the data triple t_d obtains its unique id o_i, $i = 1, \ldots, y$, *where* y *is a counts of all* t_d ($\{o_1 = \textit{the x files}, o_2 = \textit{die hard}, o_3 = \textit{the avengers}, o_4 = \textit{the x files}, o_5 = \textit{district } 9, o_6 = \textit{diehard}, o_7 = \textit{theavengers}\}$). In addition, each object o_i is associated with the identifiers (indexes) of web pages. Thanks to this, we know which page contains a specific object. The WI algorithm creates extraction patterns for the domain based on such created *big document* and with the use of FCA (Sect. 4.3).

In Fig. 3, it can be noticed that the *Wrappers induction* phase is followed by the *Pattern matching* phase and the *Update/Save* phase. In the *Patterns matching* phase, the patterns are subsequently used to extract new instances. In this phase, the instances between left and right HTML tokens are extracted. Based on the previously considered example, the WI algorithm may create a general pattern, like *<p class="film title"><br/>*(.+?)*<br/></p>*. After applying this pattern to documents, two new instances of the attribute of the film title will be extracted: *nine months* and *dead poet society*. Thus, the initial input set of seeds is extended by two new instances of the attribute of the film title. In the *Update/Save* phase, these new values of the attribute are saved into the singleton IS or the appropriate input data set is updated. Furthermore, we can use a *boosting mode* to improve the output collections of instances. The received output instances can be directed back to the input of the *semi-supervised identification* phase.

4.3 Implementation

This section describes the FCA theory (Sect. 4.3.1) which is a core of the proposed WI algorithm. Moreover, the algorithm with the use case is described in Sect. 4.3.2.

4.3.1 Theoretical preliminaries

Rudolf Wille introduced FCA in 1984. FCA is based on a partial order theory. Birkhoff created this theory in the 1930s [54,77]. FCA serves, among others, to build a mathematical notion of a *concept* and provides a formal tool for data analysis and knowledge representation. Researchers use a *concept lattice* to visualize the relations among the discovered concepts. A Hasse diagram is another name of the concept lattice. This diagram consists of nodes and edges. Each node represents the concept, and each edge represents the generalization/specialization relation. FCA is one of the methods used in knowledge engineering. Researchers use FCA to discover and build ontologies (for example, from textual data) that are specific to particular domains [14,44].

FCA consists of three steps: defining the *objects* O, *attributes* C, and *incidence relations* R; defining a *formal context* K in terms of an *attribute*, *object*, and *incidence relation*; and defining a *formal concept* for a given *formal context*. The formal context K is a triple [27]:

$$K{<}O, C, R{>} \tag{3}$$

where O, the non-empty set of objects; C, the non-empty set of attributes; R, the binary relation between objects and attributes; orc, the relation r representing the fact that an object o has an attribute c.

From the *formal context* K the following dependencies can be derived: any subset of objects $A \subseteq O$ generates a set of attributes A' that can be assigned to all objects from A, e.g. $A = \{o2, o3\} \rightarrow A' = \{c2, c3\}$ and any subset of attributes $B \subseteq C$ generates a set of objects B' that have all attributes from B, e.g. $B = \{c2\} \rightarrow B' = \{o2, o3\}$.

The *formal concept* of the context $K(O, C, R)$ is a pair *(A, B)*, where [27]: $A = B' = \{o \in O : \forall c \in B\ orc\}$—extension of (A, B) and $B = A' = \{c \in C : \forall o \in A\ orc\}$—intension of (A, B).

With each concept there is a related *extension* and *intension*. The *extension* is the class of objects described by the concept. The *intension* is the set of attributes (properties) that are common for all objects from the extension. The concepts $(A1, B1)$ and $(A2, B2)$ of the context $K(O, C, R)$ are ordered by the relation that can be defined as follows [27]:

Table 1 An example of the cross-table

	c_1 <br/>	c_2 <li class = "film title"><br/>	c_3 <ul><li class = "film title"><br/>	c_4 <p class= "film title"><br/>	c_5 </li><p class = "film title"><br/>
o_1	1	1	1	0	0
o_2	1	0	0	1	1
o_3	1	1	1	0	0
o_4	1	1	1	0	0
o_5	1	0	0	1	1
o_6	1	1	1	0	0
o_7	1	1	1	0	0

$$(A_1, B_1) \leq (A_2, B_2) \iff (A_1 \subseteq A_2 \iff B_2 \subseteq B_1) \tag{4}$$

The set of all concepts of S of the context K together with the relation $\leq$ $(S(K), \leq)$ constitutes a lattice called *concept lattice* for the formal context $K(O, C, R)$ [27].

4.3.2 The wrapper induction algorithm and the use case

The algorithm presented below has three properties. Firstly, it suffices to scan the set of input pages and the set of seeds only once to construct the patterns. Secondly, the patterns are constructed with the use of concept lattice described in Sect. 4.3.1. The pattern construction consists of finding a combination of left l and right r HTML tokens surrounding the matched seeds that make it possible to extract new candidates for seeds. Thirdly, the algorithm has parameters to control its performance, e.g. precision, recall, and F-measure. One of such parameters is the minimum length of the pattern, which is defined by the minimum number of left l and right r HTML tokens that surround the seed.

Now, it will be described how the left and right lattices are constructed based on the *big document* (Sect. 4.2.1). There is a constructed appropriate relation matrix that next serves for constructing left and right concept lattices. The matrix for building the left (prefix) lattice is shown in Table 1. The resulting lattice is shown in Fig. 8. The right (suffix) matrix and lattice are built analogously.

Table 1 shows a matrix of incidence relation between objects (indexed by seeds in the *big document*) and HTML tokens that surround them from the left (that may be viewed as FCA attributes). In the matrix, there are seven objects and five attributes. The considered HTML tokens are restricted by the maximum numbers of HTML tokens. The string of HTML tokens can be expanded (starting from the right and moving to the left) and represented by 5 attributes: *<br/>*, *<li class = "film title"><br/>*, *<ul><li class = "film title"><br/>*,*<p class = "film title"><br/>* and *</li><p class = "film title"><br/>*. The relation between the object and attribute is present only if there is the possibility to match a given object with a given attribute (what is represented by "1" inside the matrix/Table 1). In this way, it is possible to derive an appropriate left lattice from the relation matrix, which is illustrated in Fig. 8. The lattice defines the partial order described in Eq. 4 in Sect. 4.3.1. We can see two concepts k_1 and k_2 (not counting the top and bottom nodes). The split was done due to the attribute *<br/>*, i.e. by extending the pattern *<br/>*, respectively, by the *<p class = "film title">* or *<li class = "film title">* HTML token. In this way, two new separate concepts are created.

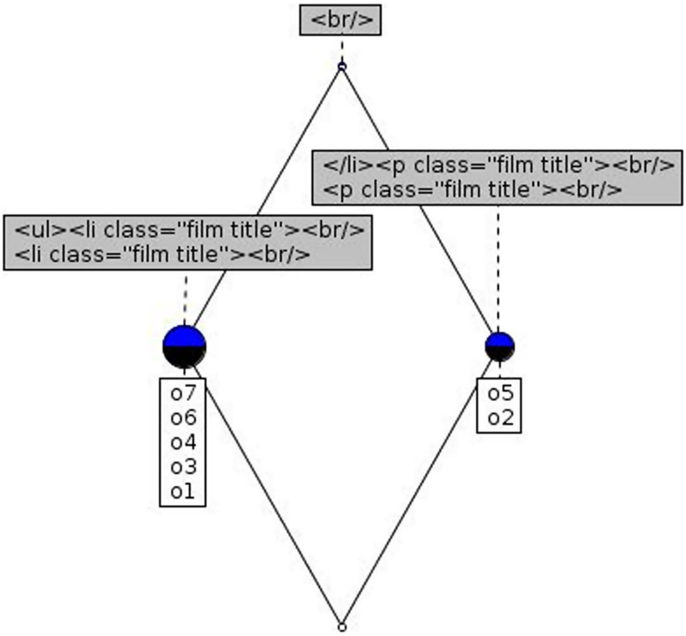
<br/>
</li><p class="film title"><br/>
<p class="film title"><br/>
<ul><li class="film title"><br/>
<li class="film title"><br/>
o7
o6
o4
o3
o1
o5
o2

Fig. 8 The concept lattice for data from Table 1

```
Require: $leftLattice$
Require: $rightLattice$
Require: $0 < minNumberOfLeftHtmlTags \leq MaxNumberOfLeftHtmlTags$
Require: $0 < minNumberOfRightHtmlTags \leq MaxNumberOfRightHtmlTags$
Require: $0 < supportConcept \leq 1$
Require: $0 < supportInterConcept \leq 1$
Require: $K_{left} \leftarrow \emptyset$ {Set of left concept}
Require: $W_{out} \leftarrow \emptyset$ {Set of building out wrappers}
1: $K_{left} \leftarrow$ receiveLeftLatticeConcept(leftLattice, minNumberOfLeftHtmlTags, supportConcept)
2: $W_{out} \leftarrow$ receiveWrappers($K_{left}$, rightLattice, minNumberOfRightHtmlTags, supportInterConcept)
```

Fig. 9 The pseudo-code of the proposed algorithm to wrapper induction

It can be noticed that the first concept k_1 aggregates in itself the information about the objects o_1, o_3, o_4, o_6, o_7 surrounded from the left by such prefixes as *<li class = "film title"><br/>* and *<ul><li class = "film title"><br/>* etc. The objects $o_i \in k_1$ are surrounded by HTML tokens expansions of lengths $conceptLength(k_1) = \{2, 3\}$. Additional information indirectly encoded inside the concepts concerns the distribution of seeds among pages that will be further used by the algorithm.

With the left and right concept lattices as an input, the pattern construction algorithm can be initiated. The pseudo-code of the algorithm is depicted in Fig. 9.

The algorithm from Fig. 9 creates the extraction patterns. Next, the patterns are used to extract new seeds. The instances are retrieved from documents belonging to the domain for which the patterns were created. The algorithm proceeds in two phases. The first phase consists of execution of the function *receiveLeftLatticeConcept()* (Fig. 9: line 1, body of

```
 1: receiveLeftLatticeConcept(leftLattice, minNumberOfLeftHtmlTags, supportConcept) {
 2: {iteration of the entire left lattice concepts}
 3: while leftLattice != end do
 4:    {$k_{i-th}$ current i-th concept from left lattice}
 5:
 6:    if conceptLength($k_{i-th}$) $\geq$ minNumberOfLeftHtmlTags then
 7:
 8:       if supportConcept($k_{i-th}$) $>$ supportConcept then
 9:          $K_{left} = K_{left} \cup \{k_{i-th}\}$
10:       end if
11:    end if
12: end while
13: return $K_{left}$
14: }
```

Fig. 10 The pseudo-code of the *receiveLeftLatticeConcept* function

Fig. 11 The pseudo-code of the *supportConcept* function

```
1: $support \leftarrow supportConcept(k_{i-th})\{$
2: return $\frac{|pagesCoverByConcept(k_{i-th})|}{countPagesCoverByAllConcept}$
3: }
```

function in Fig. 10). The second phase consists of execution of the function *receiveWrappers()* (Fig. 9: line 2, the body of the function presents in Fig. 12).

The function shown in Fig. 10 retrieves the candidates for the left patterns from the left lattice of concepts. We retrieve the patterns, which attributes (the left expansions) are of the length equal or bigger than the input parameter *minNumberOfLeftHtmlTags*. The inner function *conceptLength* (Fig. 10, line 6) is responsible for calculating the number of HTML tokens. The function from Fig. 10 also selects concepts from the left lattice that achieve the value of *support* higher than another parameter *supportConcept* (Fig. 10, line 8). This value is computed by the function *supportConcept()*. Figure 11 presents the pseudo-code of this function.

The *supportConcept()* function from Fig. 11 computes the *support*. The *support* is a ratio of a number of pages (identifiers) aggregated by a given concept and a number of pages in the domain covered by the concepts (the number of documents from the upper supremum of the lattice). The inner function *pagesCoverByConcept* from Fig. 11 (line 2) retrieves a set of identifiers of pages aggregated by a given concept k_i.

After computing the first phase, the second phase is initiated. The second phase of the algorithm executes the function *receiveWrappers()* (Fig. 9: line 2, the body of the function presents in Fig. 12).

The function from Fig. 12 is responsible for retrieving extraction patterns. During its execution, the left concepts from the left lattice and the right concepts from the right lattice are compared. The right concepts, which right expansions are not shorter than the value of input parameter, are selected *minNumberOfRightHtmlTags*. Next, if such condition is satisfied, the value (line 9) that estimates the support between the current left concept k_{left-i} and $k_{right-i}$ is computed. Its computation consists of checking what the percentage of pages is covered by the left and the right concepts. The pattern is accepted only if the computed value is not lower than *supportInterConcept*. The pattern consists of left and right expansions retrieved from the left and right concepts, $k_{left-i-th}$ and $k_{right-i-th}$.

Below, the illustration of the algorithm execution for data previously considered in Fig. 7 is presented. The following settings are assumed: *minNumberOfLeftHtmlTags* $= 3$, *minNumberOfRightHtmlTags* $= 2$, *supportConcept* $= 0.1$, *supportInterConcept* $= 0.55$ and *countPagesPerDomain* $= |D| = 3$. The first phase of the algorithm will return the

```
receiveWrappers($K_{left}$, rightLattice, minNumberOfRightHtmlTags, supportInterConcept) {
{iteration of the set of left concepts}
while $K_{left}$ != end do
    {$k_{left-i-th}$ current i-th left concept from $K_{left}$}
    while rightLattice != end do
        {$k_{right-i-th}$ current i-th concept from right lattice}

        if conceptLength($k_{right-i-th}$) $\geq$ minNumberOfRightHtmlTags then
            $L = |pagesCoverByConcept(k_{left-i-th})) \cap pagesCoverByConcept(k_{right-i-th}|$
            $M = |pagesCoverByConcept(k_{left-i-th})) \cup pagesCoverByConcept(k_{right-i-th})|$
            $mean = \frac{L}{M}$

            if mean $\geq$ supportInterConcept then
                pattern = contextualString($k_{left-i-th}$) + "(.+?)" + contextualString($k_{right-i-th}$)
                $W_{out} = W_{out} \cup \{pattern\}$
            end if
        end if
    end while
end while
return $W_{out}$
}
```

Fig. 12 The pseudo-code of the *receiveWrappers*() function

following set of left concepts $K_{left} = \{k1, k2\}$, where $k1 = \{o1, o3, o4, o6, o7\}$ and $k2 = \{o2, o5\}$. These objects cover the following documents: $pagesCoverByConcept(k1) = \{1, 2, 3\}$ and $pagesCoverByConcept(k1) = \{1, 2\}$. These concepts satisfy the following conditions $conceptLength(k_{i-th}) \geq minNumberOfLeftHtmlTags$ and $supportConcept(k_{i-th}, countPagesPerDomain) > supportConcept$.

The second phase of the algorithm returns the following set of patterns: $W_{out} =$ {*<ul><li class = “film title”><br/>(.+?)</li><p>*, *</li><p class = “film title”> <br/>(.+?)<br/></p>*}. After matching these patterns to the documents p_1, p_2, and p_3, the following new seeds will be extracted: *nine months, dead poets society, good will hunting, the departed, dead poets society*.

5 Empirical evaluation of the solution

Section 5.1 describes *a reference data set* (*a relevant set*) used to evaluate the WI algorithm. The evaluation was based on indicators described in Sect. 5.2. Section 5.3 explains the experiment's plan. Section 5.4 describes its realization and the results. Furthermore, the additional benchmark that is based on the another data set, and which is compared to another IE approach (the supervised method which was proposed by Hao et al. [28]), is presented in “Appendix A”.

5.1 The description of the reference data set

The author has not found an adequate reference data set to evaluate the proposed algorithm. In the literature, there are many references to data sets [19,51,75]. Unfortunately, these data sets are not proper to evaluate the proposed method. Well-labelled web documents from a certain domain are required. Each document from the domain must be labelled by a set of important instances (keywords). For this reason, the author created his own labelled reference data set. This data set contains 200 well-diversified documents for each existing Internet domain (*filmweb.pl*, *ptaki.info*, and *agatameble.pl*). In the rest of this section, the author uses the term "domain" rather than the Internet domain.

The test collection of HTML documents obtained (collected/crawled) from the *filmweb.pl* domain includes 200 documents. Among the 200 documents, 156 documents include information that should be extracted. The rest of the documents contain information irrelevant to the IE task, but they were not excluded. These documents emulate noise. The WI should not create patterns for these pages, and instances should not be extracted out from these pages. The author created the *reference data set* based on the 156 relevant documents. This set includes $V_{ref} = 4185$ instances of different types, i.e. different semantic classes (a film title, an actor name, etc.). The author also used 200 HTML documents from other domains, which have less complex layouts, to evaluate of the proposed WI algorithm. These domains are (1) *ptaki.info* (about birds), and (2) *agatameble.pl* (a trade offer store). The *reference data set* for these domains include $V_{ref} = 142$ and $V_{ref} = 1264$ instances, respectively.

5.1.1 Practical preliminaries

The creation of a good reference collection of seeds is a difficult, demanding and time-consuming task. This phase involves many problems, such as interpretation of the extracted (matched) information and adding it to the created *reference data set*. This is an important step, because based on these reference data, the effects of the proposed WI algorithm will be evaluated. The author decomposed the fundamental problem of information interpretation from the HTML document templates into several smaller sub-problems. This problem is quite general and can occur during analysis of most websites, which can be characterized by complicated HTML templates. In particular, the point is that the same information can be presented differently. The following brief analysis of potential problems was conducted based on observation of the IE from the *filmweb.pl* domain. This domain includes dozens of different templates. The author assumes (based on empirical observations) that the rich structure of this domain and its analysis is a good approximation to the analysis of the other domains' content. The described domain contains the HTML templates that display information about:

– a single movie/series/video game/theatre arts. The templates print the information from the following n-tuple <*polish title, english title, list of actors names, list of actors roles, music, photos*>. In addition, the author in the test set detected the templates that present a short and full version of the above n-tuple, e.g. the list of actors names can present all values (a full version) or k-first values (a short version).
– a set of movies. The templates print the information from the following n-tuples <*polish and english films titles*>,
– a set of movies. The templates print the information from the following n-tuples <*polish films titles, english films titles*>. Furthermore, in the test set, the author detected two different templates that represent above-mentioned tuples,
– a single actor. The templates print the information from the following n-tuples <*actor's name and surname, polish films titles, english films titles, names of films roles*>,
– the user's favourite films. The templates print the information from the following n-tuples <*prefix as the film year production and english films titles, polish films titles*>.

Furthermore, in the test set, the author detected two different templates that represent the above-mentioned n-tuple.

The author had the one fundamental problem during analysis of website templates and extracting information from them to the *reference data set*. This problem concerned the interpretation of the data. There might be a problem with the interpretation of (1) the HTML tags of a layout that surrounds the information, (2) the created extraction patterns, and (3) extracted information. The HTML tags layout and the created patterns may suggest some correct forms of the same information. For example, we may consider the following situations:

- there is an available layout of HTML tags *<tag1><tag2>[information to extraction]<tag3> [suffix associated with the information to extraction]<tag4>*. Based on the HTML tags we may create two patterns *{<tag1><tag2>*(.+?)*<tag3>*, *<tag1> <tag2>*(.+?)*<tag4>}*. Using these patterns, we may extract the following information *[information to extraction]* and *[information to extraction]<tag3>[suffix associated with the information to extraction]*. Using a simple pre-processing, we may filter out the unnecessary HTML tags from extracted information. This way, the correct form of instance, i.e. *[information to extraction] [suffix associated with the information to extraction]* is obtained. It often occurs, for instance, in case of displaying the cast. Usually, additional information is added to the film role name. This information indicates whether an actor lent their own voice or not, e.g. *barry "big bear" thorne* and *barry "big bear" thorne (voice)*, etc.,
- there is an available layout of HTML tags *<tag1>*(.+?)*<tag2>*, which covers, for example, name of an actor such as a *matthew perry i*. However, for the given page the WI may create another pattern, which extracts a similar semantic information, e.g. *matthew perry*,
- there is also an available layout of HTML tags *<tag1>*(.+?)*<tag2>*, which covers, for example, the film names in the following form *production year* | *english film title* (*1979* | *Apocalypse Now*). However, this layout may also cover only the prefixes, as a *production year* (*1979*) because there is no English version of the film title.

After consideration of the situations mentioned above, the author decided to add different variations of the same correct information to the designed *reference data set*. This means that the *Wrappers induction* and *Pattern matching* components should extract all possible forms of data that have been identified as correct. Furthermore, we may assume a simple pre-processing. This pre-processing removes the unnecessary HTML tags from the extracted data. Thus, the author assumes that, for example, *matthew perry i* and *matthew perry* or *apocalypse now*, *1979*| *apocalypse now* and *1979*, etc., from the HTML documents are correct.

5.2 The indicators to evaluate the proposed solutions

The following indicators were used for the evaluation of the proposed WI algorithm [25,46, 67]:

- Precision

$$Prec = \frac{|V_{ref} \cap V_{rec}|}{|V_{rec}|} \tag{5}$$

- Recall

$$Rec = \frac{|V_{ref} \cap V_{rec}|}{|V_{ref}|} \tag{6}$$

- F-measure

$$F = \frac{2 \cdot Prec \cdot Rec}{Prec + Rec} \tag{7}$$

- Macro-average precision

$$Prec_{mac\text{-}avg} = \frac{\sum_{k=1}^{n} Prec_{p_k}}{n} = \frac{1}{n} \sum_{k=1}^{n} \frac{\left| V_{ref_{p_k}} \cap V_{rec_{p_k}} \right|}{\left| V_{rec_{p_k}} \right|} \tag{8}$$

- Macro-average recall

$$Rec_{mac\text{-}avg} = \frac{\sum_{k=1}^{n} Rec_{p_k}}{n} = \frac{1}{n} \sum_{k=1}^{n} \frac{\left| V_{ref_{p_k}} \cap V_{rec_{p_k}} \right|}{\left| V_{ref_{p_k}} \right|} \tag{9}$$

- Macro-average F-measure

$$F_{mac\text{-}avg} = \frac{1}{n} \sum_{k=1}^{n} \frac{2 \cdot Prec_{p_k} \cdot Rec_{p_k}}{Prec_{p_k} + Rec_{p_k}} \tag{10}$$

where V_{ref} is the set of reference instances (the set of reference attribute values, for the given website) and V_{rec} is the set of received instances (the set of received attribute values/the retrieved set, for the given website); $|V_{ref}|$ is the size of the set of reference instances and $|V_{rec}|$ is the size of the set of received instances; n is the count of web page for the given website; $Prec_{p_k}$ is the precision and Rec_{p_k} is the recall of k-th document; $V_{ref_{p_k}}$ is the set of reference instances and $V_{rec_{p_k}}$ is the set of received instances of k-th document; $|V_{ref_{p_k}}|$ is the size of the set of reference instances and $|V_{rec_{p_k}}|$ is the size of the set of received instances of k-th document.

5.3 The plan of the experiment

The author conducted experiments with different configurations of components to evaluate the proposed algorithm/system in relation to the SEAL. Figure 13 presents the scheme of the experiment plan.

In Fig. 13 the *Seeds set* component includes the elements (the configuration names) such as *Set of seeds without semantic labels (S1)* and *Set of seeds with semantic labels (S2)*. The $S1$ set contains instances that belong to one general attribute *thing name*. The $S2$ set contains instances that are split between more specific attributes (the taxonomy of seeds, see Sect. 4). The *Pre-processing Domain's web pages* component includes the elements such as *HTML tags with attributes and values (H1)*, *HTML tags without the attribute values (H2)*, and *HTML tags without attributes and their values (H3)*. These elements may or may not remove some parts of HTML documents (see Sect. 4.2.1). The *Matching* component includes the elements such as *Matching seeds to the whole HTML document (M1)* and *Matching seeds between HTML tags (M2)*. The $M1$ matches each seed to the whole HTML document; for example, we can match $seed = x$ *files* to *<a href="…/**x files**">**x files**</a>*. The $M2$ matches each seed only to the between HTML tags; for example, we can match $seed = x$ *files* to *<a href="…/x files">**x files**</a>*. The *Wrapper induction* component includes the elements such as *Wrapper induction on chars level per document (W1)*, *Wrapper induction on chars level per domain (W2)*, and *Wrapper induction on HTML tags level per domain (W3)*. These elements induce the wrappers in three different ways.

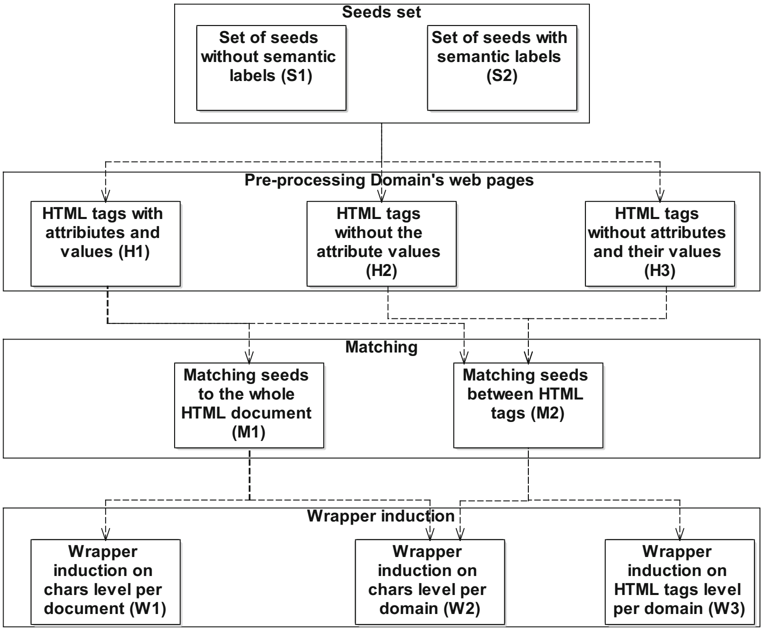
Seeds set
Set of seeds without semantic labels (S1)
Set of seeds with semantic labels (S2)
Pre-processing Domain's web pages
HTML tags with attribiutes and values (H1)
HTML tags without the attribute values (H2)
HTML tags without attributes and their values (H3)
Matching
Matching seeds to the whole HTML document (M1)
Matching seeds between HTML tags (M2)
Wrapper induction
Wrapper induction on chars level per document (W1)
Wrapper induction on chars level per domain (W2)
Wrapper induction on HTML tags level per domain (W3)

Fig. 13 The scheme of the experiment plan

As shown in Fig. 13, we can conduct the following experiments: $Experiment_1$: $S1 \rightarrow H1 \rightarrow M1 \rightarrow W1$, $Experiment_2$: $S2 \rightarrow H1 \rightarrow M1 \rightarrow W1$, $Experiment_3$: $S1 \rightarrow H1 \rightarrow M1 \rightarrow W2$, $Experiment_4$: $S2 \rightarrow H1 \rightarrow M1 \rightarrow W2$, $Experiment_5$: $S1 \rightarrow H1 \rightarrow M2 \rightarrow W2$, $Experiment_6$: $S2 \rightarrow H1 \rightarrow M2 \rightarrow W2$, $Experiment_7$: $S1 \rightarrow H2 \rightarrow M2 \rightarrow W2$, $Experiment_8$: $S2 \rightarrow H2 \rightarrow M2 \rightarrow W2$, $Experiment_9$: $S1 \rightarrow H3 \rightarrow M2 \rightarrow W2$, $Experiment_{10}$: $S2 \rightarrow H3 \rightarrow M2 \rightarrow W2$, $Experiment_{11}$: $S1 \rightarrow H1 \rightarrow M2 \rightarrow W3$, $Experiment_{12}$: $S2 \rightarrow H1 \rightarrow M2 \rightarrow W3$, $Experiment_{13}$: $S1 \rightarrow H2 \rightarrow M2 \rightarrow W3$, $Experiment_{14}$: $S2 \rightarrow H2 \rightarrow M2 \rightarrow W3$, $Experiment_{15}$: $S1 \rightarrow H3 \rightarrow M2 \rightarrow W3$ and $Experiment_{16}$: $S2 \rightarrow H3 \rightarrow M2 \rightarrow W3$.

The $Experiment_1$ and $Experiment_2$ refer to test of the SEAL algorithm (only the wrapper phase without the ranking phase). The $Experiment_3$–$Experiment_{10}$ refer to test of the BigGrams system, which works on the chars level. The $Experiment_{11}$–$Experiment_{16}$ refer to test of the BigGrams system, which works on the HTML tags level.

5.4 The realization of the experiment plan and the results

Section 5.4.1 presents the results of the experiment where the BigGrams system works in the batch mode. Section 5.4.2 shows the results of the experiment where the BigGrams system works in the boosting mode.

5.4.1 The batch mode

The evaluation of the proposed system to extraction of information is based on the comparison of the set of reference attribute values V_{ref} with the set of received attribute values V_{rec} from this system. The author evaluated each of three domains mentioned above. The author changed the values of the *numberOfChars* attribute from 1 to 9, and the best result for the SEAL algorithm was noted. The author changed the *support inter concept* parameter from 0.1 to 0.9 every 0.1 for the BigGrams system. The author set the following parameters $minNumberOfLeftChars = 2$, $minNumberOfRightChars = 12$, and $minNumberOfLeftChars = 4$, $minNumberOfRightChars = 4$, respectively, for the BigGrams system with the chars level mode. The author set the following parameters $minNumberOfLeftHtmlTags = 1$, $minNumberOfRightHtmlTags = 1$, and $minNumberOfLeftHtmlTags = 2$, $minNumberOfRightHtmlTags = 2$, respectively, for the BigGrams system with the HTML tags level mode. Also, the author assumed the constant parameters such as $support\ concept = 0.1$ and $filtered\ outlier\ seed = true$. The author created data sets of the following numbers of seeds $|S_{input}|$: filmweb.pl $|S_{input}| = \sum_{a \in A} |V_a| = 1020$ seeds (Table 4 shows the used attributes $a \in A$); ptaki.info $|S_{input}| = |V_{a=latin\text{-}bird\text{-}name}| + |V_{a=polish\text{-}bird\text{-}name}| = 6 + 6 = 12$ seeds and agatameble.pl $|S_{input}| = |V_{a=product\text{-}name}| = 65$ seeds.

Tables 2 and 3 contain the best results achieved in the experiments. These tables contain the comparison of the results from the SEAL system (Experiment 1–2) with the BigGrams system, which works on chars level (Experiment 3–10) and HTML tags level (Experiment 11–16), with different configurations of the components. Section 5.3 (Fig. 13) explains the whole research plan in detail.

Based on the results included in Tables 2 and 3, we may conclude that the proposed algorithm works the worst when the WI uses HTML tags with attributes and their values. On the contrary, it works best when the WI uses HTML tags with attributes without values. The HTML tag granularity significantly improves the proposed WI solution. Thanks to using the more complex structure of seeds (the taxonomy of seeds), rather than *bag of seeds*, we can achieve better results for a more complex domain, i.e. higher value of the $F_{mic\text{-}avg}$, etc. The SEAL is not an appropriate solution for direct extraction of the important instances (keywords) from the domain.

Based on the all the conducted experiments, some parameter values of the algorithm can be determined. The maximum values of these indicators for the filmweb.pl domain were obtained using the following input parameters of the algorithm: $|S_{input}| = 1020$, $minNumberOfLeftHtmlTags = 2$, $minNumberOfRightHtmlTags = 2$, $support\ concept = 0.1$, $support\ inter\ concept = 0.2$, $filtered\ attributes\ values = true$ and $filtered\ outlier\ seed = true$. For these parameters, the author obtained the following indicator values of the algorithm evaluations: $Prec = 0.9948$, $Rec = 0.9603$, $F = 0.9773$, $Prec_{mac\text{-}avg} = 0.9738 \pm 0.1565$, $Rec_{mac\text{-}avg} = 0.9362 \pm 0.1733$ and $F_{mac\text{-}avg} = 0.9523 \pm 0.1613$. In addition, for these parameters per-page the $V_{ref_{p_k}}$ and $V_{rec_{p_k}}$ sets were compared. Figure 14 shows this comparison.

Figure 14 (the top plot) shows the per-page comparison of the $V_{ref_{p_k}}$ set with the $V_{rec_{p_k}}$ set. This figure shows an almost perfect overlap between these two data sets. Due to this fact, the values of evaluation indicators are high. Moreover, the experiment shows that such value of parameters can be established that the algorithm can produce almost a perfect overlap, i.e. $|V_{ref_{p_k}}| \cong |V_{rec_{p_k}}|$. Furthermore, Fig. 14 (the bottom plot) shows the boxplots of precision ($Prec_{p_k}$), Recall (Rec_{p_k}) and F-measure ($F - measure_{p_k}$) values in terms of median. We may see that (1) almost all values of each indicator are nearly 1 and (2) there are few outlier points that increase the value of standard deviation (s).

Table 2 The comparison of the results from the SEAL (SEAL’—the chars level and the seed set, SEAL”—the chars level and the taxonomy of seeds) and the BigGrams system (BigGrams*—the chars level with the different configurations, BigGrams**—the HTML tags level with the different configurations) for the filmweb.pl and ptaki.info domains and s is a standard deviation

Domain name	Experiment name	Tested system	Indicators					
			$Prec$	Rec	F	$Prec_{mac\text{-}avg} \pm s$	$Rec_{mac\text{-}avg} \pm s$	$F_{mac\text{-}avg} \pm s$
filmweb.pl	$Experiment_1$	SEAL’	0.2579	0.2320	0.2443	0.0893 ± 0.2489	0.2360 ± 0.3809	0.1035 ± 0.2557
	$Experiment_2$	SEAL”	0.4174	0.2397	0.3045	0.1893 ± 0.3134	0.1904 ± 0.3391	0.1369 ± 0.2583
	$Experiment_3$	BigGrams*	0.8043	0.2258	0.3526	0.6092 ± 0.4286	0.3777 ± 0.3827	0.4218 ± 0.3907
	$Experiment_4$	BigGrams*	0.4417	0.2533	0.3219	0.3224 ± 0.2560	0.3549 ± 0.3458	0.2990 ± 0.2641
	$Experiment_5$	BigGrams*	0.7567	0.4029	0.5258	0.4990 ± 0.4238	0.3503 ± 0.3685	0.3848 ± 0.3743
	$Experiment_6$	BigGrams*	0.7114	0.3658	0.4832	0.3256 ± 0.2832	0.4495 ± 0.3953	0.3286 ± 0.3051
	$Experiment_7$	BigGrams*	0.6143	0.6234	0.6188	0.0944 ± 0.1437	0.3355 ± 0.3840	0.1374 ± 0.1922
	$Experiment_8$	BigGrams*	0.6999	0.5935	0.6424	0.1808 ± 0.1911	0.4335 ± 0.3709	0.2361 ± 0.2321
	$Experiment_9$	BigGrams*	0.4622	0.0131	0.0256	0.3408 ± 0.4479	0.2341 ± 0.4092	0.2447 ± 0.4063
	$Experiment_{10}$	BigGrams*	0.7255	0.2994	0.4239	0.1849 ± 0.2641	0.2737 ± 0.3899	0.2138 ± 0.3017
	$Experiment_{11}$	BigGrams**	0.9990	0.2432	0.3912	0.5748 ± 0.4954	0.4555 ± 0.4291	0.4959 ± 0.4467
	$Experiment_{12}$	BigGrams**	0.9722	0.3589	0.5243	0.8950 ± 0.2932	0.5987 ± 0.4038	0.6482 ± 0.3947
	$Experiment_{13}$	BigGrams**	0.9993	0.7016	0.8244	0.7946 ± 0.4045	0.6719 ± 0.3878	0.7155 ± 0.3862
	$Experiment_{14}$	BigGrams**	0.9948	0.9603	0.9773	0.9738 ± 0.1565	0.9362 ± 0.1733	0.9523 ± 0.1613
	$Experiment_{15}$	BigGrams**	0.9958	0.7864	0.8788	0.7894 ± 0.4026	0.6620 ± 0.3841	0.7072 ± 0.3823
	$Experiment_{16}$	BigGrams**	0.9962	0.8691	0.9283	0.9484 ± 0.1998	0.8807 ± 0.2357	0.9007 ± 0.2111
ptaki.info	$Experiment_1$	SEAL’	1	0.0490	0.0933	0.6700 ± 0.4714	0.6525 ± 0.4681	0.6583 ± 0.4672
	$Experiment_2$	SEAL”	0.5	0.0559	0.1006	0.6609 ± 0.4737	0.6525 ± 0.4681	0.6532 ± 0.4699

Table 2 continued

Domain name	Experiment name	Tested system	Indicators					
			Prec	*Rec*	*F*	$Prec_{mac\text{-}avg} \pm s$	$Rec_{mac\text{-}avg} \pm s$	$F_{mac\text{-}avg} \pm s$
	$Experiment_3$	BigGrams*	1	0.4965	0.6636	0.9950 ± 0.0707	0.8150 ± 0.2471	0.8750 ± 0.1718
	$Experiment_4$	BigGrams*	1	0.993	0.9965	1 ± 0	0.9975 ± 0.0354	0.9983 ± 0.0236
	$Experiment_5$	BigGrams*	1	0.5035	0.6698	1 ± 0	0.8175 ± 0.2413	0.8783 ± 0.1609
	$Experiment_6$	BigGrams*	1	0.5874	0.7401	1 ± 0	0.8500 ± 0.2297	0.9000 ± 0.1531
	$Experiment_7$	BigGrams*	1	0.5035	0.6698	1 ± 0	0.8175 ± 0.2413	0.8783 ± 0.1609
	$Experiment_8$	BigGrams*	1	0.585	0.7401	1 ± 0	0.8500 ± 0.2297	0.9000 ± 0.1531
	$Experiment_9$	BigGrams*	1	0.5874	0.6698	1 ± 0	0.8175 ± 0.2413	0.8783 ± 0.1609
	$Experiment_{10}$	BigGrams*	1	0.5874	0.7401	1 ± 0	0.8500 ± 0.2297	0.9000 ± 0.1531
	$Experiment_{11}$	BigGrams**	1	1	1	1 ± 0	1 ± 0	1 ± 0
	$Experiment_{12}$	BigGrams**	1	1	1	1 ± 0	1 ± 0	**1 ± 0**
	$Experiment_{13}$	BigGrams**	1	1	1	1 ± 0	1 ± 0	1 ± 0
	$Experiment_{14}$	BigGrams**	1	1	1	1 ± 0	1 ± 0	**1 ± 0**
	$Experiment_{15}$	BigGrams**	1	1	1	1 ± 0	1 ± 0	1 ± 0
	$Experiment_{16}$	BigGrams**	1	1	1	1 ± 0	1 ± 0	**1 ± 0**

Table 3 The comparison of the results from the SEAL (SEAL'—the chars level and the seed set, SEAL''—the chars level and the taxonomy of seeds) and the BigGrams system (BigGrams*—the chars level with the different configurations, BigGrams**—the HTML tags level with the different configurations) for the agatameble.pl domain and s is a standard deviation

Domain name	Experiment name	Tested system	Indicators					
			Prec	*Rec*	*F*	$Prec_{mac\text{-}avg} \pm s$	$Rec_{mac\text{-}avg} \pm s$	$F_{mac\text{-}avg} \pm s$
agatameble.pl	$Experiment_1$	SEAL'	0.8533	0.0506	0.0955	0.1542 ± 0.3610	0.1049 ± 0.2756	0.1171 ± 0.2791
	$Experiment_2$	SEAL''	0.8533	0.0506	0.0955	0.1642 ± 0.3703	0.1105 ± 0.2791	0.1243 ± 0.3005
	$Experiment_3$	BigGrams*	0.8113	0.7241	0.7652	0.7608 ± 0.3236	0.7073 ± 0.3168	0.7266 ± 0.3159
	$Experiment_4$	BigGrams*	0.8113	0.7241	0.7652	0.7605 ± 0.3235	0.7073 ± 0.3168	0.7264 ± 0.3158
	$Experiment_5$	BigGrams*	0.6534	0.7779	0.7102	0.6125 ± 0.3074	0.7145 ± 0.3354	0.6497 ± 0.3148
	$Experiment_6$	BigGrams*	0.6534	0.7779	0.7102	0.6125 ± 0.3074	0.7145 ± 0.3354	0.6497 ± 0.3148
	$Experiment_7$	BigGrams*	0.8113	0.7241	0.7652	0.7608 ± 0.3236	0.7073 ± 0.3168	0.7266 ± 0.3159
	$Experiment_8$	BigGrams*	0.7773	0.7779	0.7776	0.7341 ± 0.3332	0.7145 ± 0.3354	0.7192 ± 0.3310
	$Experiment_9$	BigGrams*	0.8113	0.7241	0.7652	0.7608 ± 0.3236	0.7073 ± 0.3168	0.7266 ± 0.3159
	$Experiment_{10}$	BigGrams*	0.7773	0.7779	0.7776	0.7341 ± 0.3332	0.7145 ± 0.3354	0.7192 ± 0.3310
	$Experiment_{11}$	BigGrams**	0.7059	0.1802	0.2872	0.4137 ± 0.3039	0.1354 ± 0.2408	0.1606 ± 0.2319
	$Experiment_{12}$	BigGrams**	0.7037	0.1802	0.2870	0.4139 ± 0.3042	0.1363 ± 0.2417	0.1617 ± 0.2335
	$Experiment_{13}$	BigGrams**	0.9247	1	0.9609	0.9087 ± 0.1557	1 ± 0	0.9424 ± 0.1221
	$Experiment_{14}$	BigGrams**	0.9240	1	0.9605	0.9084 ± 0.1555	1 ± 0	0.9422 ± 0.1220
	$Experiment_{15}$	BigGrams**	0.9247	1	0.9609	0.9087 ± 0.1557	1 ± 0	0.9424 ± 0.1221
	$Experiment_{16}$	BigGrams**	0.9240	1	0.9605	0.9084 ± 0.1555	1 ± 0	0.9422 ± 0.1220

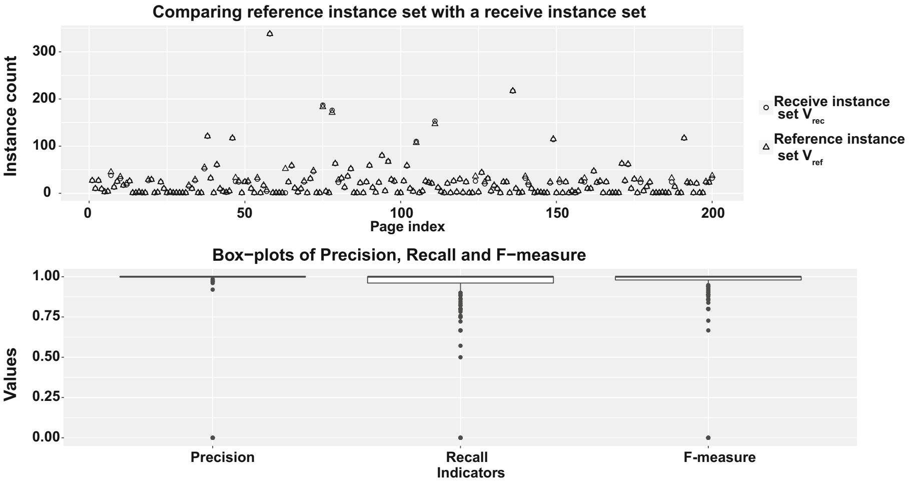

Fig. 14 The per-page comparison of the reference instances set $V_{ref_{p_k}}$ with the set of received instances $V_{rec_{p_k}}$ (the *top plot*), and *boxplots* of precision ($Prec_{p_k}$), recall (Rec_{p_k}) and F-measure ($F\text{-}measure_{p_k}$) values in terms of median (the *bottom plot*)

5.4.2 The boosting mode

The author conducted two experiments for the boosting mode. Both experiments assumed two things: (1) we have the output results of the information extraction, and (2) we can set these results to the input of the BigGrams system.

In the first case, the author assumed that the BigGrams system can perfectly extract new values for each attribute $a \in A$ of the taxonomy of seeds, i.e. the extracted values are semantically related with the attributes, e.g. the BigGrams system will retrieve only the new true names of actors for the *actor names* attribute. The author called this experiment *the perfect boosting*. The author created seven taxonomies with different numbers of input seeds $|V_a|$. Each taxonomy in each iteration was set as the input of the BigGrams system. Table 4 presents the results obtained in this experiment.

In the second case, the author assumed that the BigGrams system might extract the imperfect new values for each attribute $a \in A$ of the taxonomy of seeds, i.e. the extracted values may not be semantically related with the attributes. For example, the BigGrams system will retrieve new false values, such as bmw, *x files* for the *actor names* attribute. The author called this experiment as *the non-perfect boosting*. The author created three initial taxonomies with different numbers of input seeds $|V_a|$. Each taxonomy in each iteration was extended by newly extracted instances, and they were set in the input of the BigGrams system. Table 5 presents the results obtained in this experiment.

The author used only the *filmweb.pl* domain to test, since this domain is constructed on a complex taxonomy of seeds (six different attributes) and it has a complex layout. The domain *agatameble.pl* has only one attribute, the *ptaki.info* has two attributes, but it also has a simple layout, and only twelve seeds are required to achieve the max value of the indicators. In the experiment, the author employed exactly the same configuration that was previously used in the batch mode and produced the best results.

Table 4 presents results of the experiment where *the perfect boosting* was used. In this case, in each iteration, the BigGrams system created the valid sets of data input based on

Table 4 The results of the experiment: the perfect boosting

Iteration number	$\lvert V_a \rvert$ for each attribute name a						$\sum \lvert V_a \rvert$	$\frac{\sum \lvert V_a \rvert}{\lvert V_{ref} \rvert}$ (%)	F	$F_{mac\text{-}avg}$
	Actor names	Film titles pl	Film titles en	Film titles pl/en	Role names	Another				
1	4	4	4	2	2	2	18	0.43	0.333	0.657 ± 0.393
2	10	15	15	5	5	5	55	1.3	0.884	0.87 ± 0.278
3	26	28	30	10	10	10	114	2.7	0.953	0.943 ± 0.175
4	26	60	50	15	15	15	181	4.3	0.964	0.95 ± 0.161
5	41	112	90	15	22	19	299	7.1	0.968	0.95 ± 0.161
6	71	216	170	29	43	19	548	13	0.974	0.952 ± 0.161
7	131	425	330	29	86	19	1020	24.4	0.977	0.953 ± 0.161

Table 5 The results of the experiment: the non-perfect boosting

$\sum \lvert V_a \rvert$	Iteration number	F	$F_{mac\text{-}avg}$
18	1	0.33	0.657 ± 0.393
	2	0.853	0.95 ± 0.161
	3	0.853	0.95 ± 0.161
55	1	0.884	0.87 ± 0.278
	2	0.839	0.882 ± 0.162
	3	0.822	0.878 ± 0.164
	4	0.822	0.878 ± 0.164
548	1	0.974	0.952 ± 0.161
	2	0.896	0.914 ± 0.182
	3	0.853	0.917 ± 0.142
	4	0.853	0.92 ± 0.127
	5	0.853	0.92 ± 0.127

the output. Thus, the maximum values of indicators are achieved in the incremental fashion (after a maximum of 7 iterations). Furthermore, in each iteration each V_a set is extended by new instances and the indicators are increased.

Table 5 presents the results of the experiment where *the non-perfect boosting* was used. As we can see, the values of indicators saturate too quickly, which is followed by their decrease after some iterations. It occurs because the generic pattern is created despite the separation of the *film title* attribute in the input attributes, such as *polish film title* and *english film title*. This pattern extracts both titles. As a result, in the next iteration, the algorithm loses the ability to create general patterns. The algorithm in each iteration keeps rediscovering the same information. In addition, the algorithm begins to emit noise (false values, the case 55 seeds). As a result, the algorithm cannot diversify the patterns or extract new seeds. The algorithm becomes overfitting and it fits the data overly. When the output data and the algorithm indicators (precision and recall) were reviewed, it was noted that the large values of recall (0.9–0.95) corresponded to lower values of precision (0.8–0.95). Thus, the algorithm extracts a new value not present in the reference set.

6 Conclusion

The most important findings of this work are as follows:

- the empirical research shows that we can improve and achieve a high quality of the WI output results by using the described techniques,
- the empirical research shows that the quality of information extraction depends on the (1) form of input data, (2) pre-processing domain's web pages, (3) matching techniques, and (4) the level of HTML documents representation (the granularity of HTML tags),
- the worst results are obtained when the HTML tags contain attributes and their values. In this case, the algorithm creates very detailed patterns with a low degree of generalization,
- the best results are achieved when the proposed taxonomy approach is used as the input of the WI algorithm, and when the pre-processing technique clearing the values of HTML attributes, where the seeds are matched only between HTML tags, and if we use the

tags level rather than the chars level representation of HTML documents. Thanks to this configuration, the WI created generic patterns covering the most of the expected instance,
- if we can ensure well-diversified input data, the WI may be used in the boosting mode,
- the weak assumption made about the fact that *on the basis of seeds belonging to semantic classes patterns, that will extract new semantically consistent instances, will be created* is useful, but it is also only partly right. Adoption of this assumption in the first iteration of the proposed algorithm produces good results,
- the BigGrams system is suitable for extracting relevant keywords from Internet domains.

During the evaluation phase, the author received a set of new instances, which coincides with the set of reference instances. We should remember that the newly extracted instances have not been evaluated in terms of semantics. However, as shown by the following experiments based on boosting without verification of the semantic instance, the next iteration of the algorithm may worsen initial results. The created wrappers generate instances of different classes of semantics. For this reason, the author intends to add an automatic mechanism for defining the semantics of the new instances (*Verification* component). Experiments involving the perfect boosting yielded promising results.

The presented BigGrams system has achieved promising experimental results. It seems that the method still has some potential and allows further optimization. The algorithm still has four parameters (*minNumberOfLeftHtmlTags*, *minNumberOfRightHtmlTags*, *supportConcept*, *supportInterConcept*) that give an opportunity to control the results (precision, recall, F-measure). In the next optimization step of the algorithm, the author wants to reduce the numbers of these parameters or determine their values automatically. So far, the author conducted an experiment in this direction. The experiment confirmed that such algorithm can be constructed. The created initial model based only on the parameters *minNumberOfLeftHtmlTags* and *minNumberOfRightHtmlTags* is able to induce wrappers that for the most complex domain *filmweb.pl* give worse results for 5% F-measure. Also, this issue will be further researched and developed.

Acknowledgements The study is cofounded by the European Union from resources of the European Social Fund. Project PO KL "Information technologies: Research and their interdisciplinary applications", Agreement UDA-POKL.04.01.01-00-051/10-00.

A Another empirical evaluation of the solution

In this appendix, the author has presented (1) another empirical evaluation of the BigGrams system and (2) the comparison of BigGrams system and the IE solution that was proposed by Hao et al. [28] IE. The benchmarks are based on the new data set that is presented in Sect. A.1. The evaluation was based on indicators that are described in Sect. 5.2. Furthermore, the new indicators, ranking methods, and statistics tests were proposed. Section A.2 describes these properties. Section A.3 explains the created experiment's plan. Section A.4 describes its realization and the results. The additional conclusions and findings are presented in Sect. A.5.

Table 6 Overview of the experimental data set [28]

Vertical	#Sites	#Pages	Attributes
Autos	10	17,923	Model, price, engine, fuel economy
Books	10	20,000	Title, author, ISBN 13, publisher, publication date
Cameras	10	5258	Model, price, manufacturer
Jobs	10	20,000	Title, company, location, date posted
Movies	10	20,000	Title, director, genre, mpaa rating
NBA Players	10	4405	Name, team, height, weight
Restaurants	10	20,000	Name, address, phone, cuisine
Universities	10	16,705	Name, phone, website, type

A.1 The description of the reference data set

The new evaluation process is based on SWDE data set with ground-truth information [28]. Table 6 shows the basic statistics of the SWDE data set.

The data set presented in Table 6 contains around 124,000 pages collected from 80 websites. Vertical, a semantically diverse reference data set of values (N-tuple IS), was constructed for each website. Each vertical consists of a set of (3–5) common attributes. The websites are related to 8 verticals, including Autos, Books, Cameras, Jobs, Movies, NBA Players, Restaurants, and Universities. For each website, 200–2000 pages, each containing structured data of one entity, Hao Q. et al. had downloaded and extracted value of the appropriate attribute.

A.2 The indicators and methods to evaluate and comparison of the solutions

Hao et al. [28] defined in the different ways the precision (P_1), recall (R_1), and F-measure (F_1) indicators. For each attribute, they defined the precision (P_1) of a method as the number of pages whose ground-truth attribute values are correctly extracted, called page hits, divided by the number of pages from which the method extracts values. Recall (R_1) was the page hits divided by the number of pages containing ground-truth attribute values. F-Measure (F_1) was the harmonic mean of precision (P_1) and recall (R_1). Furthermore, they reported, “As a side note, it is possible that a page contains more than one ground-truth values of an attribute (e.g. co-authors of a book), while the current solution in this paper is designed to detect only one attribute value. For this case, an extracted value is considered to be correct if it matches any labelled value in the ground-truth”.

In addition, the author used the ranking method and statistics test to compare the IE methods, i.e. the BigGrams system which worked on the HTML tags level with the different configurations, and Hao et al. [28] IE (see Sect. A.3). The procedure of evaluation or ranking creation to compare several methods is a common and well-described problem in the scientific literature. In particular, in the machine learning field, we may enumerate several kinds of solutions to resolve the problems mentioned above. We may justify a statistic-based approach [57], and more general approach to build ranking method (a ranking-based approach) [2,40,41,79].

In the presented case, the author has utilized a mixed strategies to evaluate and compare the IE methods. The author has used the Hellinger-TOPSIS multi-criteria evaluation method [41,42] to create the ranking of IE methods. In the evaluation phase (Sect. A.4), the author utilized four different F-measure indicators and their values to build rankings and compare the IE methods. Based on these indicators, four different rankings of IE methods were created. The author has used the Spearman's rank (r_s) [33] to compare obtained rankings. This method was employed by Ali et al. [2] in the similar comparison of ranking given by the different multi-criteria methods. The Hmisc R-project package [29] was used to compute a value of Spearman's rank. This package includes a function to calculate the values of Spearman's rank with the value of statistical significance tests, i.e. p-value. Thanks to the statistics test, we can check that there is no monotonic association between created rankings based on different indicators (the null hypothesis H_{S0}). Furthermore, the Wilcoxon–Mann–Whitney statistic test [70] was used to check if the values of two experiments (a couple of experiments) ranked by the Hellinger-TOPSIS multi-criteria evaluation method were independent. Using this test, we can decide whether the population distributions are identical, without assuming them to follow the normal distribution. In this case, the null hypothesis (H_{W0}) is that the two different experiments have identical populations. For the statistic tests mentioned above, a statistical significance $\alpha = 0.05$ was assumed.

A.3 The plan of the experiment

The author in the current experiments had chosen the BigGrams mode, which worked on the HTML tags level. The author created six different configurations for this mode. This level was set because the previously conducted experiments and results (Sect. 5) had shown that this level gave the best results. The IE method proposed by Hao et al. [28] was set as a baseline method.

In this experiment plan, the author did not change the *support inter concept* parameter, and the value of this parameter was set to 0.1. Also, the author assumed the constant parameters such as *support concept* $= 0.1$ and *filtered outlier seed* $=$ *true*. In five experiments the value of *minNumberOfLeftHtmlTags* and *minNumberOfRightHtmlTags* was set to 1. In the last sixth experiment, the value of *minNumberOfLeftHtmlTags* and *minNumberOfRightHtmlTags* was set to 2. Furthermore, the author used the following pre-processing strategies (1) HTML tags with attributes and values, (2) HTML tags without the attribute values, and (3) A HREF HTML tags without a value. The last strategy is new. The author based on the empirical research and review of the reference data set established that the most important values are placed in web page links. The author has observed that IE process that use the pre-processing which cleans all attributes and their values gives the worst results rather than clean values of chosen HTML tags attributes such as *a href*. For this reason, this type of pre-processing was excluded from the experiment plan. Also, in current experiment plan, the author tested another property of the BigGrams system such as the ability to select and use the unique patterns, i.e. we may block the created patterns that cover/extract more than one value from a given website. Finally, for each website, 10–30 input seeds, was set. All strategies/experiments are listed below:

- $Experiment_1$ used the first pre-processing strategy, i.e. HTML tags with attribute and values, and utilized all created IE patterns,
- $Experiment_2$ used the first pre-processing strategy, i.e. HTML tags with attribute and values, and utilized the unique IE patterns,
- $Experiment_3$ used the second pre-processing strategy, i.e. HTML tags without the attribute values, and utilized all created IE patterns,
- $Experiment_4$ used the second pre-processing strategy, i.e. HTML tags without the attribute values, and utilized unique IE patterns,

- *Experiment*$_5$ used the third pre-processing strategy, i.e. A HREF HTML tags without a value, and utilized all created IE patterns,
- *Experiment*$_6$ used the third pre-processing strategy, i.e. A HREF HTML tags without a value, and utilized all created IE patterns (different numbers of left and right HTML token are used compared to *Experiment*$_5$).

In the presented experiment plan, the author used the shorter notation of used indicators, i.e. P_1, R_1, and F_1 are the indicators described in Sect. A.2, $P_2 = Prec_{mac\text{-}avg}$, $R_2 = Rec_{mac\text{-}avg}$, and $F_2 = F_{mac\text{-}avg}$ are the indicators described in Sect. 5.2, and their values are measured by attribute level (each attribute is considered separately), the indicators $P_3 = Prec_{mac\text{-}avg}$, $R_3 = Rec_{mac\text{-}avg}$, $F_3 = F_{mac\text{-}avg}$, $P_4 = Prec$, $R_4 = Rec$, $F_4 = F - measure$ are also shown in Sect. 5.2. In this case, their values are measured for a group of attributes (the author considered $V_{ref_{p_k}}$ of a given page or V_{ref} of a given website as connected sets of the value of each attribute, for example, $V_{ref_{p_k}} = V_{ref,title_{p_k}} \cup V_{ref,price_{p_k}} \cup V_{ref,manufacturer_{p_k}}$ for Cameras vertical).

It is worth noticing that Hao et al. [28] reported only values of P_1, R_1, and F_1. Also, there is no available software to reconstruct or conduct new experiments based on their solution. For these reasons, the author noticed only their available values of the indicators mentioned above. In the rest of this article, the *Experiment*$_7$ relates to Hao et al. [28] approach.

A.4 The realization of the experiment plan and the results

This section presents the results obtained from conducted experiments. Tables 7, 9, 11, 13, 15, 17, 19, 21 contain the mean (μ) and standard deviation (s) values of precision (P_1, P_2), recall (R_1, R_2), and F-measure (F_1, F_2) that were achieved based on 10 websites per given vertical. The presented results come from analysis based on the attribute level, i.e. separate statistics for each attribute. Hao et al. [28] reported only values of P_1, R_1, and F_1 indicators. For this reason, the author put a sign "-" in Tables mentioned above for P_2, R_2, and F_2 indicators.

Tables 8, 10, 12, 14, 16, 18, 20, 22 contain the mean (μ) and standard deviation (s) values of precision (P_3, P_4), recall (R_3, R_4), and F-measure (F_3, F_4) that were achieved based on 10 websites per given vertical. The presented results come from analysis based on the group of attributes, i.e. statistics for the group of attributes. Hao et al. [28] did not consider this type of analysis and did not report such values.

Table 7 The comparison of the mean (μ) and standard deviation (s) values of precision (P_1, P_2), recall (R_1, R_2), and F-measure (F_1, F_2) obtained from the Hao et al. [28] IE, and the BigGrams system which worked on the HTML tags level with the different configurations ($Experiment_1$–$Experiment_6$) for the *IS* about autos (verticle: autos)

Method name	$a \in A$	$\mu_{P_1} \pm s_{P_1}$	$\mu_{R_1} \pm s_{R_1}$	$\mu_{F_1} \pm s_{F_1}$	$\mu_{P_2} \pm s_{P_2}$	$\mu_{R_2} \pm s_{R_2}$	$\mu_{F_2} \pm s_{F_2}$
$Experiment_1$	Engine	0.999 ± 0.003	0.999 ± 0.003	0.999 ± 0.003	0.761 ± 0.386	0.977 ± 0.07	0.796 ± 0.331
	Fuel economy	0.993 ± 0.022	0.908 ± 0.292	0.91 ± 0.27	0.715 ± 0.392	0.896 ± 0.289	0.753 ± 0.363
	Model	1 ± 0	1 ± 0	1 ± 0	0.895 ± 0.209	0.999 ± 0.003	0.929 ± 0.139
	Price	0.9 ± 0.316	0.9 ± 0.316	0.9 ± 0.316	0.62 ± 0.421	0.9 ± 0.316	0.67 ± 0.387
$Experiment_2$	Engine	0.75 ± 0.422	0.584 ± 0.44	0.611 ± 0.448	0.573 ± 0.429	0.584 ± 0.44	0.576 ± 0.432
	Fuel economy	0.9 ± 0.316	0.808 ± 0.406	0.814 ± 0.393	0.808 ± 0.406	0.757 ± 0.4	0.774 ± 0.397
	Model	1 ± 0	1 ± 0	1 ± 0	0.896 ± 0.209	0.999 ± 0.003	0.93 ± 0.139
	Price	0.797 ± 0.42	0.699 ± 0.479	0.7 ± 0.476	0.639 ± 0.458	0.699 ± 0.479	0.658 ± 0.459
$Experiment_3$	Engine	0.999 ± 0.003	0.999 ± 0.003	0.999 ± 0.003	0.71 ± 0.385	0.977 ± 0.07	0.761 ± 0.326
	Fuel economy	0.984 ± 0.034	1 ± 0	0.992 ± 0.018	0.716 ± 0.377	0.979 ± 0.035	0.769 ± 0.318
	Model	1 ± 0	1 ± 0	1 ± 0	0.608 ± 0.448	0.999 ± 0.003	0.649 ± 0.433
	Price	1 ± 0	0.9 ± 0.315	0.901 ± 0.313	0.697 ± 0.422	0.9 ± 0.315	0.73 ± 0.399
$Experiment_4$	Engine	0.75 ± 0.422	0.584 ± 0.44	0.611 ± 0.448	0.575 ± 0.432	0.584 ± 0.44	0.578 ± 0.434
	Fuel economy	0.859 ± 0.328	0.712 ± 0.464	0.723 ± 0.448	0.712 ± 0.464	0.661 ± 0.446	0.678 ± 0.448
	Model	1 ± 0	1 ± 0	1 ± 0	0.832 ± 0.229	0.999 ± 0.003	0.888 ± 0.153
	Price	0.897 ± 0.315	0.699 ± 0.478	0.701 ± 0.475	0.639 ± 0.457	0.651 ± 0.472	0.626 ± 0.445
$Experiment_5$	Engine	0.999 ± 0.003	0.999 ± 0.003	0.999 ± 0.003	0.761 ± 0.386	0.977 ± 0.07	0.796 ± 0.331
	Fuel economy	0.985 ± 0.031	1 ± 0	0.992 ± 0.016	0.734 ± 0.362	0.981 ± 0.032	0.786 ± 0.306
	Model	1 ± 0	1 ± 0	1 ± 0	0.608 ± 0.447	0.999 ± 0.003	0.65 ± 0.432
	Price	1 ± 0	1 ± 0	1 ± 0	0.72 ± 0.373	1 ± 0	0.77 ± 0.318
$Experiment_6$	Engine	0.939 ± 0.152	0.91 ± 0.172	0.923 ± 0.158	0.821 ± 0.292	0.91 ± 0.172	0.832 ± 0.266
	Fuel economy	0.992 ± 0.025	1 ± 0	0.996 ± 0.013	0.783 ± 0.361	0.939 ± 0.153	0.788 ± 0.305
	Model	1 ± 0	0.991 ± 0.027	0.996 ± 0.014	0.653 ± 0.451	0.991 ± 0.027	0.68 ± 0.433

Table 7 continued

Method name	$a \in A$	$\mu_{P_1} \pm s_{P_1}$	$\mu_{R_1} \pm s_{R_1}$	$\mu_{F_1} \pm s_{F_1}$	$\mu_{P_2} \pm s_{P_2}$	$\mu_{R_2} \pm s_{R_2}$	$\mu_{F_2} \pm s_{F_2}$
	Price	0.998 ± 0.007	0.997 ± 0.008	0.998 ± 0.008	0.749 ± 0.33	0.997 ± 0.008	0.803 ± 0.263
Hao et al. [28] IE	Engine	0.82 ± 0.14	0.82 ± 0.14	0.82 ± 0.14	–	–	–
	Fuel economy	0.81 ± 0.2	0.73 ± 0.18	0.77 ± 0.19	–	–	–
	Model	0.46 ± 0.27	0.41 ± 0.26	0.43 ± 0.26	–	–	–
	Price	0.8 ± 0.19	0.79 ± 0.19	0.8 ± 0.19	–	–	–

The analysis is based on the attribute level, i.e. separate statistics for each attribute

Table 8 The comparison of the mean (μ) and standard deviation (s) values of precision (P_3, P_4), recall (R_3, R_4), and F-measure (F_3, F_4) obtained from the BigGrams system which worked on the HTML tags level with the different configurations ($Experiment_1$–$Experiment_6$) for the *IS* about autos (verticle: autos)

Method name	$a \in A$	$\mu_{P_3} \pm s_{P_3}$	$\mu_{R_3} \pm s_{R_3}$	$\mu_{F_3} \pm s_{F_3}$	$\mu_{P_4} \pm s_{P_4}$	$\mu_{R_4} \pm s_{R_4}$	$\mu_{F_4} \pm s_{F_4}$
$Experiment_1$	Engine	0.748 ± 0.222	0.943 ± 0.112	0.787 ± 0.191	0.755 ± 0.27	0.934 ± 0.145	0.803 ± 0.189
	Fuel economy						
	Model						
	Price						
$Experiment_2$	Engine	0.729 ± 0.268	0.76 ± 0.287	0.734 ± 0.275	0.895 ± 0.119	0.824 ± 0.236	0.834 ± 0.154
	Fuel economy						
	Model						
	Price						
$Experiment_3$	Engine	0.683 ± 0.197	0.964 ± 0.077	0.728 ± 0.167	0.648 ± 0.29	0.935 ± 0.184	0.708 ± 0.212
	Fuel economy						
	Model						
	Price						
$Experiment_4$	Engine	0.69 ± 0.174	0.724 ± 0.227	0.693 ± 0.196	0.862 ± 0.128	0.752 ± 0.277	0.772 ± 0.184
	Fuel economy						
	Model						
	Price						
$Experiment_5$	Engine	0.706 ± 0.207	0.989 ± 0.018	0.75 ± 0.176	0.64 ± 0.289	0.994 ± 0.016	0.745 ± 0.22
	Fuel economy						
	Model						
	Price						
$Experiment_6$	Engine	0.751 ± 0.187	0.959 ± 0.049	0.776 ± 0.152	0.709 ± 0.276	0.975 ± 0.043	0.792 ± 0.2
	Fuel economy						
	Model						
	Price						

The analysis is based on the group of attributes, i.e. statistics for the group of attributes

Table 9 The comparison of the mean (μ) and standard deviation (s) values of precision (P_1, P_2), recall (R_1, R_2), and F-measure (F_1, F_2) obtained from the Hao et al. [28] IE, and the BigGrams system which worked on the HTML tags level with the different configurations ($Experiment_1$–$Experiment_6$) for the *IS* about books (verticle: books)

Method name	$a \in A$	$\mu_{P_1} \pm s_{P_1}$	$\mu_{R_1} \pm s_{R_1}$	$\mu_{F_1} \pm s_{F_1}$	$\mu_{P_2} \pm s_{P_2}$	$\mu_{R_2} \pm s_{R_2}$	$\mu_{F_2} \pm s_{F_2}$
$Experiment_1$	Author	0.665 ± 0.469	0.056 ± 0.139	0.076 ± 0.166	0.024 ± 0.04	0.054 ± 0.133	0.028 ± 0.051
	ISBN 13	1 ± 0	0.999 ± 0.004	0.999 ± 0.002	0.995 ± 0.012	0.999 ± 0.004	0.996 ± 0.008
	Publication date	1 ± 0	0.998 ± 0.007	0.999 ± 0.004	0.898 ± 0.209	0.998 ± 0.007	0.931 ± 0.139
	Publisher	1 ± 0	0.805 ± 0.411	0.81 ± 0.401	0.757 ± 0.413	0.805 ± 0.411	0.772 ± 0.406
	Title	0.999 ± 0.002	0.992 ± 0.009	0.996 ± 0.005	0.789 ± 0.334	0.992 ± 0.009	0.817 ± 0.294
$Experiment_2$	Author	0.679 ± 0.472	0.046 ± 0.107	0.07 ± 0.149	0.043 ± 0.097	0.044 ± 0.102	0.043 ± 0.097
	ISBN 13	0.792 ± 0.418	0.769 ± 0.41	0.78 ± 0.413	0.769 ± 0.41	0.769 ± 0.41	0.769 ± 0.41
	Publication date	0.797 ± 0.42	0.764 ± 0.409	0.779 ± 0.413	0.746 ± 0.403	0.764 ± 0.409	0.752 ± 0.404
	Publisher	0.9 ± 0.316	0.582 ± 0.494	0.597 ± 0.494	0.582 ± 0.494	0.582 ± 0.494	0.582 ± 0.494
	Title	0.991 ± 0.028	0.983 ± 0.037	0.987 ± 0.033	0.911 ± 0.146	0.983 ± 0.037	0.934 ± 0.11
$Experiment_3$	Author	0.992 ± 0.015	0.999 ± 0.002	0.996 ± 0.008	0.243 ± 0.349	0.98 ± 0.035	0.286 ± 0.358
	ISBN 13	1 ± 0	0.999 ± 0.004	0.999 ± 0.002	0.9 ± 0.299	0.999 ± 0.004	0.906 ± 0.285
	Publication date	1 ± 0	0.998 ± 0.007	0.999 ± 0.004	0.851 ± 0.309	0.998 ± 0.007	0.877 ± 0.271
	Publisher	1 ± 0	0.997 ± 0.008	0.998 ± 0.004	0.872 ± 0.234	0.997 ± 0.008	0.908 ± 0.169
	Title	0.994 ± 0.008	0.992 ± 0.007	0.993 ± 0.008	0.467 ± 0.453	0.992 ± 0.007	0.502 ± 0.433
$Experiment_4$	author	0.86 ± 0.31	0.86 ± 0.309	0.86 ± 0.31	0.749 ± 0.306	0.839 ± 0.305	0.777 ± 0.3
	ISBN 13	0.792 ± 0.418	0.769 ± 0.41	0.78 ± 0.413	0.761 ± 0.405	0.769 ± 0.41	0.763 ± 0.406
	Publication date	0.797 ± 0.42	0.659 ± 0.448	0.683 ± 0.452	0.64 ± 0.438	0.659 ± 0.448	0.647 ± 0.441
	Publisher	0.8 ± 0.422	0.675 ± 0.469	0.686 ± 0.474	0.675 ± 0.469	0.675 ± 0.469	0.675 ± 0.469
	Title	0.977 ± 0.065	0.956 ± 0.084	0.966 ± 0.072	0.815 ± 0.211	0.956 ± 0.084	0.859 ± 0.163

Table 9 continued

Method name	$a \in A$	$\mu_{P_1} \pm s_{P_1}$	$\mu_{R_1} \pm s_{R_1}$	$\mu_{F_1} \pm s_{F_1}$	$\mu_{P_2} \pm s_{P_2}$	$\mu_{R_2} \pm s_{R_2}$	$\mu_{F_2} \pm s_{F_2}$
Experiment$_5$	Author	0.992 ± 0.015	0.999 ± 0.002	0.996 ± 0.008	0.247 ± 0.351	0.98 ± 0.035	0.292 ± 0.358
	ISBN 13	1 ± 0	0.999 ± 0.004	0.999 ± 0.002	0.9 ± 0.299	0.999 ± 0.004	0.906 ± 0.285
	Publication date	1 ± 0	0.998 ± 0.007	0.999 ± 0.004	0.898 ± 0.209	0.998 ± 0.007	0.931 ± 0.139
	Publisher	1 ± 0	0.997 ± 0.008	0.998 ± 0.004	0.949 ± 0.15	0.997 ± 0.008	0.964 ± 0.103
	Title	0.994 ± 0.009	0.992 ± 0.009	0.993 ± 0.009	0.468 ± 0.452	0.992 ± 0.009	0.503 ± 0.432
Experiment$_6$	Author	0.979 ± 0.044	0.982 ± 0.041	0.98 ± 0.042	0.682 ± 0.327	0.945 ± 0.045	0.731 ± 0.297
	ISBN 13	0.993 ± 0.023	0.991 ± 0.023	0.992 ± 0.023	0.942 ± 0.153	0.991 ± 0.023	0.959 ± 0.103
	Publication date	1 ± 0	0.902 ± 0.288	0.912 ± 0.266	0.853 ± 0.311	0.902 ± 0.288	0.869 ± 0.294
	Publisher	0.9 ± 0.316	0.897 ± 0.315	0.898 ± 0.316	0.849 ± 0.333	0.897 ± 0.315	0.864 ± 0.32
	Title	0.964 ± 0.093	0.921 ± 0.135	0.939 ± 0.108	0.619 ± 0.352	0.921 ± 0.135	0.663 ± 0.302
Hao et al. [28] IE	Author	0.95 ± 0.04	0.89 ± 0.04	0.92 ± 0.04	–	–	–
	ISBN 13	0.84 ± 0.19	0.84 ± 0.18	0.84 ± 0.18	–	–	–
	Publication date	0.88 ± 0.08	0.88 ± 0.08	0.88 ± 0.08	–	–	–
	Publisher	0.81 ± 0.06	0.81 ± 0.06	0.81 ± 0.06	–	–	–
	Title	0.89 ± 0.13	0.87 ± 0.14	0.88 ± 0.14	–	–	–

The analysis is based on the attribute level, i.e. separate statistics for each attribute

Table 10 The comparison of the mean (μ) and standard deviation (s) values of precision (P_3, P_4), recall (R_3, R_4), and F-measure (F_3, F_4) obtained from the BigGrams system which worked on the HTML tags level with the different configurations ($Experiment_1$–$Experiment_6$) for the *IS* about books (verticle: books)

Method name	$a \in A$	$\mu_{P_3} \pm s_{P_3}$	$\mu_{R_3} \pm s_{R_3}$	$\mu_{F_3} \pm s_{F_3}$	$\mu_{P_4} \pm s_{P_4}$	$\mu_{R_4} \pm s_{R_4}$	$\mu_{F_4} \pm s_{F_4}$
$Experiments_1$	Author ISBN 13 Publication date Publisher Title	0.693 ± 0.087	0.769 ± 0.09	0.709 ± 0.08	0.776 ± 0.286	0.725 ± 0.064	0.714 ± 0.166
$Experiments_2$	Author ISBN 13 Publication date Publisher Title	0.61 ± 0.225	0.629 ± 0.222	0.616 ± 0.224	0.923 ± 0.111	0.622 ± 0.174	0.728 ± 0.157
$Experiments_3$	Author ISBN 13 Publication date Publisher Title	0.667 ± 0.172	0.993 ± 0.007	0.696 ± 0.169	0.434 ± 0.284	0.99 ± 0.018	0.555 ± 0.263
$Experiments_4$	Author ISBN 13 Publication date Publisher Title	0.728 ± 0.261	0.779 ± 0.273	0.744 ± 0.265	0.853 ± 0.112	0.82 ± 0.238	0.809 ± 0.177

Table 10 continued

Method name	$a \in A$	$\mu_{P_3} \pm s_{P_3}$	$\mu_{R_3} \pm s_{R_3}$	$\mu_{F_3} \pm s_{F_3}$	$\mu_{P_4} \pm s_{P_4}$	$\mu_{R_4} \pm s_{R_4}$	$\mu_{F_4} \pm s_{F_4}$
$Experiments_5$	Author	0.692 ± 0.165	0.993 ± 0.007	0.719 ± 0.162	0.462 ± 0.275	0.99 ± 0.018	0.586 ± 0.251
	ISBN 13						
	Publication date						
	Publisher						
	Title						
$Experiments_6$	Author	0.789 ± 0.129	0.931 ± 0.118	0.817 ± 0.124	0.683 ± 0.266	0.929 ± 0.061	0.754 ± 0.199
	ISBN 13						
	Publication date						
	Publisher						
	Title						

The analysis is based on the group of attributes, i.e. statistics for the group of attributes

Table 11 The comparison of the mean (μ) and standard deviation (s) values of precision (P_1, P_2), recall (R_1, R_2), and F-measure (F_1, F_2) obtained from the Hao et al. [28] IE, and the BigGrams system which worked on the HTML tags level with the different configurations (*Experiment*$_1$–*Experiment*$_6$) for the *IS* about cameras (verticle: cameras)

Method name	$a \in A$	$\mu_{P_1} \pm s_{P_1}$	$\mu_{R_1} \pm s_{R_1}$	$\mu_{F_1} \pm s_{F_1}$	$\mu_{P_2} \pm s_{P_2}$	$\mu_{R_2} \pm s_{R_2}$	$\mu_{F_2} \pm s_{F_2}$
Experiment$_1$	Manufacturer	0.83 ± 0.366	0.818 ± 0.387	0.822 ± 0.379	0.434 ± 0.459	0.717 ± 0.382	0.424 ± 0.411
	Model	0.981 ± 0.035	0.981 ± 0.035	0.981 ± 0.035	0.799 ± 0.229	0.864 ± 0.201	0.794 ± 0.176
	Price	0.992 ± 0.021	1 ± 0.001	0.996 ± 0.011	0.529 ± 0.332	0.992 ± 0.021	0.609 ± 0.294
Experiment$_2$	Manufacturer	0.53 ± 0.483	0.418 ± 0.483	0.423 ± 0.48	0.361 ± 0.44	0.315 ± 0.387	0.327 ± 0.402
	Model	0.895 ± 0.252	0.795 ± 0.376	0.807 ± 0.354	0.641 ± 0.366	0.666 ± 0.383	0.622 ± 0.339
	Price	0.94 ± 0.165	0.845 ± 0.221	0.88 ± 0.184	0.652 ± 0.257	0.842 ± 0.227	0.705 ± 0.23
Experiment$_3$	Manufacturer	0.9 ± 0.316	0.9 ± 0.316	0.9 ± 0.316	0.373 ± 0.445	0.809 ± 0.34	0.372 ± 0.394
	Model	0.993 ± 0.013	0.993 ± 0.013	0.993 ± 0.013	0.695 ± 0.298	0.932 ± 0.155	0.739 ± 0.261
	Price	0.992 ± 0.021	1 ± 0.001	0.995 ± 0.011	0.273 ± 0.386	0.992 ± 0.021	0.329 ± 0.363
Experiment$_4$	Manufacturer	0.599 ± 0.497	0.497 ± 0.505	0.499 ± 0.506	0.377 ± 0.432	0.387 ± 0.424	0.358 ± 0.399
	Model	0.893 ± 0.267	0.868 ± 0.284	0.879 ± 0.277	0.696 ± 0.325	0.693 ± 0.333	0.647 ± 0.279
	Price	0.912 ± 0.174	0.826 ± 0.217	0.857 ± 0.183	0.614 ± 0.283	0.822 ± 0.222	0.672 ± 0.246
Experiment$_5$	Manufacturer	0.9 ± 0.316	0.9 ± 0.316	0.9 ± 0.316	0.425 ± 0.454	0.797 ± 0.334	0.432 ± 0.405
	Model	0.99 ± 0.022	0.989 ± 0.023	0.989 ± 0.023	0.696 ± 0.294	0.872 ± 0.205	0.701 ± 0.243
	Price	0.992 ± 0.021	1 ± 0.001	0.996 ± 0.011	0.525 ± 0.337	0.992 ± 0.021	0.606 ± 0.299
Experiment$_6$	Manufacturer	0.9 ± 0.316	0.899 ± 0.316	0.9 ± 0.316	0.538 ± 0.448	0.797 ± 0.335	0.527 ± 0.414
	Model	0.989 ± 0.023	0.919 ± 0.21	0.937 ± 0.157	0.633 ± 0.285	0.803 ± 0.261	0.639 ± 0.242
	Price	0.996 ± 0.006	0.997 ± 0.006	0.997 ± 0.006	0.619 ± 0.332	0.996 ± 0.006	0.69 ± 0.29
Hao et al. [28] IE	Manufacturer	0.96 ± 0.06	0.93 ± 0.06	0.94 ± 0.06	–	–	–
	Model	0.93 ± 0.07	0.88 ± 0.06	0.9 ± 0.07	–	–	–
	Price	0.98 ± 0.04	0.9 ± 0.05	0.94 ± 0.05	–	–	–

The analysis is based on the attribute level, i.e. separate statistics for each attribute

Table 12 The comparison of the mean (μ) and standard deviation (s) values of precision (P_3, P_4), recall (R_3, R_4), and F-measure (F_3, F_4) obtained from the BigGrams system which worked on the HTML tags level with the different configurations ($Experiment_1$–$Experiment_6$) for the *IS* about cameras (verticle: cameras)

Method name	$a \in A$	$\mu_{P_3} \pm s_{P_3}$	$\mu_{R_3} \pm s_{R_3}$	$\mu_{F_3} \pm s_{F_3}$	$\mu_{P_4} \pm s_{P_4}$	$\mu_{R_4} \pm s_{R_4}$	$\mu_{F_4} \pm s_{F_4}$
$Experiments_1$	Manufacturer	0.587 ± 0.234	0.858 ± 0.152	0.609 ± 0.217	0.501 ± 0.235	0.883 ± 0.179	0.614 ± 0.211
	Model						
	Price						
$Experiments_2$	Manufacturer	0.551 ± 0.293	0.607 ± 0.231	0.551 ± 0.258	0.693 ± 0.211	0.736 ± 0.28	0.669 ± 0.22
	Model						
	Price						
$Experiments_3$	Manufacturer	0.447 ± 0.213	0.911 ± 0.121	0.48 ± 0.208	0.371 ± 0.211	0.927 ± 0.165	0.507 ± 0.211
	Model						
	Price						
$Experiments_4$	Manufacturer	0.562 ± 0.243	0.634 ± 0.22	0.559 ± 0.226	0.638 ± 0.208	0.752 ± 0.258	0.656 ± 0.211
	Model						
	Price						
$Experiments_5$	Manufacturer	0.548 ± 0.211	0.887 ± 0.132	0.579 ± 0.205	0.456 ± 0.218	0.891 ± 0.18	0.58 ± 0.208
	Model						
	Price						
$Experiments_6$	Manufacturer	0.597 ± 0.204	0.865 ± 0.127	0.618 ± 0.201	0.474 ± 0.205	0.843 ± 0.205	0.578 ± 0.187
	Model						
	Price						

The analysis is based on the group of attributes, i.e. statistics for the group of attributes

Table 13 The comparison of the mean (μ) and standard deviation (s) values of precision (P_1, P_2), recall (R_1, R_2), and F-measure (F_1, F_2) obtained from the Hao et al. [28] IE, and the BigGrams system which worked on the HTML tags level with the different configurations (*Experiment*$_1$–*Experiment*$_6$) for the *IS* about jobs (verticle: jobs)

Method name	$a \in A$	$\mu_{P_1} \pm s_{P_1}$	$\mu_{R_1} \pm s_{R_1}$	$\mu_{F_1} \pm s_{F_1}$	$\mu_{P_2} \pm s_{P_2}$	$\mu_{R_2} \pm s_{R_2}$	$\mu_{F_2} \pm s_{F_2}$
Experiment$_1$	Company	0.82 ± 0.382	0.7251 ± 0.444	0.731 ± 0.435	0.635 ± 0.475	0.725 ± 0.444	0.651 ± 0.462
	Date posted	1 ± 0	1 ± 0	1 ± 0	0.985 ± 0.048	1 ± 0	0.99 ± 0.032
	Location	1 ± 0	0.999 ± 0.003	0.999 ± 0.002	0.696 ± 0.407	0.999 ± 0.003	0.739 ± 0.359
	Title	1 ± 0.001	0.9993 ± 0.001	0.999 ± 0.001	0.942 ± 0.172	0.999 ± 0.001	0.959 ± 0.12
Experiment$_2$	Company	0.76 ± 0.405	0.4343 ± 0.488	0.454 ± 0.475	0.431 ± 0.491	0.434 ± 0.488	0.432 ± 0.49
	Date posted	0.7 ± 0.483	0.6743 ± 0.472	0.685 ± 0.475	0.659 ± 0.463	0.674 ± 0.472	0.664 ± 0.465
	Location	0.501 ± 0.526	0.4844 ± 0.511	0.492 ± 0.517	0.48 ± 0.507	0.484 ± 0.511	0.482 ± 0.508
	Title	1 ± 0.001	0.998 ± 0.005	0.999 ± 0.003	0.984 ± 0.045	0.983 ± 0.048	0.978 ± 0.041
Experiment$_3$	Company	0.997 ± 0.007	0.9593 ± 0.12	0.974 ± 0.074	0.549 ± 0.415	0.959 ± 0.12	0.604 ± 0.367
	Date posted	1 ± 0	1 ± 0	1 ± 0	0.985 ± 0.048	1 ± 0	0.99 ± 0.032
	Location	1 ± 0	0.999 ± 0.003	0.999 ± 0.002	0.613 ± 0.423	0.999 ± 0.003	0.668 ± 0.372
	Title	0.999 ± 0.001	0.9993 ± 0.001	0.999 ± 0.001	0.813 ± 0.307	0.999 ± 0.001	0.857 ± 0.245
Experiment$_4$	Company	0.897 ± 0.315	0.7714 ± 0.389	0.797 ± 0.367	0.706 ± 0.378	0.771 ± 0.389	0.728 ± 0.375
	Date posted	0.7 ± 0.483	0.6743 ± 0.472	0.685 ± 0.475	0.659 ± 0.463	0.674 ± 0.472	0.664 ± 0.465
	Location	0.501 ± 0.526	0.4841 ± 0.511	0.492 ± 0.517	0.484 ± 0.511	0.484 ± 0.511	0.484 ± 0.511
	Title	1 ± 0.001	0.9965 ± 0.01	0.998 ± 0.005	0.982 ± 0.045	0.981 ± 0.048	0.977 ± 0.041
Experiment$_5$	Company	0.997 ± 0.007	0.9593 ± 0.12	0.974 ± 0.074	0.633 ± 0.412	0.959 ± 0.12	0.677 ± 0.365
	Date posted	1 ± 0	1 ± 0	1 ± 0	0.985 ± 0.048	1 ± 0	0.99 ± 0.032
	Location	1 ± 0	0.999 ± 0.003	0.999 ± 0.002	0.696 ± 0.407	0.999 ± 0.003	0.739 ± 0.359
	Title	1 ± 0.001	0.9993 ± 0.001	0.999 ± 0.001	0.863 ± 0.29	0.999 ± 0.001	0.891 ± 0.238

Table 13 continued

Method name	$a \in A$	$\mu_{P_1} \pm s_{P_1}$	$\mu_{R_1} \pm s_{R_1}$	$\mu_{F_1} \pm s_{F_1}$	$\mu_{P_2} \pm s_{P_2}$	$\mu_{R_2} \pm s_{R_2}$	$\mu_{F_2} \pm s_{F_2}$
Experiment$_6$	Company	0.995 ± 0.014	0.9554 ± 0.12	0.971 ± 0.075	0.819 ± 0.262	0.955 ± 0.12	0.854 ± 0.215
	Date posted	0.983 ± 0.054	0.9571 ± 0.093	0.968 ± 0.068	0.95 ± 0.105	0.957 ± 0.093	0.953 ± 0.1
	Location	1 ± 0	0.999 ± 0.003	0.999 ± 0.002	0.783 ± 0.363	0.999 ± 0.003	0.816 ± 0.318
	Title	1 ± 0.001	0.9967 ± 0.009	0.998 ± 0.005	0.896 ± 0.254	0.953 ± 0.096	0.884 ± 0.215
Hao et al. [28] IE	Company	0.84 ± 0.24	0.8 ± 0.22	0.82 ± 0.22	–	–	–
	Date posted	0.79 ± 0.2	0.77 ± 0.19	0.78 ± 0.2	–	–	–
	Location	0.87 ± 0.07	0.84 ± 0.07	0.85 ± 0.07	–	–	–
	Title	0.99 ± 0.03	0.93 ± 0.04	0.95 ± 0.04	–	–	–

The analysis is based on the attribute level, i.e. separate statistics for each attribute

Table 14 The comparison of the mean (μ) and standard deviation (s) values of precision (P_3, P_4), recall (R_3, R_4), and F-measure (F_3, F_4) obtained from the BigGrams system which worked on the HTML tags level with the different configurations (*Experiment*$_1$–*Experiment*$_6$) for the *IS* about jobs (verticle: jobs)

Method name	$a \in A$	$\mu_{P_3} \pm s_{P_3}$	$\mu_{R_3} \pm s_{R_3}$	$\mu_{F_3} \pm s_{F_3}$	$\mu_{P_4} \pm s_{P_4}$	$\mu_{R_4} \pm s_{R_4}$	$\mu_{F_4} \pm s_{F_4}$
Experiments$_1$	Company Date posted Location Title	0.815 ± 0.185	0.931 ± 0.111	0.835 ± 0.173	0.73 ± 0.28	0.96 ± 0.075	0.805 ± 0.216
Experiments$_2$	Company Date posted Location Title	0.638 ± 0.292	0.644 ± 0.285	0.639 ± 0.288	0.939 ± 0.108	0.789 ± 0.159	0.853 ± 0.126
Experiments$_3$	Company Date posted Location Title	0.74 ± 0.19	0.989 ± 0.03	0.78 ± 0.163	0.729 ± 0.278	0.982 ± 0.051	0.81 ± 0.209
Experiments$_4$	Company Date posted Location Title	0.708 ± 0.259	0.728 ± 0.254	0.713 ± 0.255	0.942 ± 0.106	0.849 ± 0.129	0.886 ± 0.097
Experiments$_5$	Company Date posted Location Title	0.794 ± 0.192	0.989 ± 0.03	0.824 ± 0.166	0.731 ± 0.279	0.982 ± 0.051	0.811 ± 0.21
Experiments$_6$	Company Date posted Location Title	0.862 ± 0.185	0.966 ± 0.057	0.877 ± 0.163	0.837 ± 0.254	0.946 ± 0.08	0.865 ± 0.193

The analysis is based on the group of attributes, i.e. statistics for the group of attributes

Table 15 The comparison of the mean (μ) and standard deviation (s) values of precision (P_1, P_2), recall (R_1, R_2), and F-measure (F_1, F_2) obtained from the Hao et al. [28] IE, and the BigGrams system which worked on the HTML tags level with the different configurations ($Experiment_1$–$Experiment_6$) for the *IS* about movies (verticle: movies)

Method name	$a \in A$	$\mu_{P_1} \pm s_{P_1}$	$\mu_{R_1} \pm s_{R_1}$	$\mu_{F_1} \pm s_{F_1}$	$\mu_{P_2} \pm s_{P_2}$	$\mu_{R_2} \pm s_{R_2}$	$\mu_{F_2} \pm s_{F_2}$
$Experiment_1$	Director	0.705 ± 0.417	0.2429 ± 0.404	0.251 ± 0.401	0.163 ± 0.326	0.242 ± 0.404	0.178 ± 0.342
	Genre	0.985 ± 0.041	0.8002 ± 0.285	0.85 ± 0.226	0.591 ± 0.342	0.717 ± 0.324	0.587 ± 0.333
	Mpaa rating	0.956 ± 0.14	0.9537 ± 0.139	0.955 ± 0.14	0.894 ± 0.221	0.954 ± 0.139	0.911 ± 0.183
	Title	0.891 ± 0.313	0.8909 ± 0.313	0.891 ± 0.313	0.808 ± 0.386	0.891 ± 0.313	0.82 ± 0.366
$Experiment_2$	Director	0.702 ± 0.439	0.1052 ± 0.301	0.114 ± 0.299	0.105 ± 0.301	0.105 ± 0.301	0.105 ± 0.301
	Genre	0.969 ± 0.079	0.6159 ± 0.414	0.663 ± 0.384	0.569 ± 0.409	0.524 ± 0.423	0.536 ± 0.416
	Mpaa rating	0.9 ± 0.316	0.7065 ± 0.468	0.715 ± 0.456	0.647 ± 0.464	0.706 ± 0.468	0.663 ± 0.458
	Title	0.891 ± 0.313	0.8909 ± 0.313	0.891 ± 0.313	0.89 ± 0.313	0.891 ± 0.313	0.89 ± 0.313
$Experiment_3$	Director	0.797 ± 0.396	0.8026 ± 0.399	0.8 ± 0.398	0.114 ± 0.299	0.796 ± 0.397	0.128 ± 0.295
	Genre	0.993 ± 0.015	0.9991 ± 0.002	0.996 ± 0.008	0.434 ± 0.424	0.992 ± 0.015	0.488 ± 0.421
	Mpaa rating	0.945 ± 0.139	0.9537 ± 0.139	0.949 ± 0.139	0.752 ± 0.347	0.943 ± 0.139	0.786 ± 0.309
	Title	0.987 ± 0.018	0.9868 ± 0.018	0.987 ± 0.018	0.392 ± 0.405	0.987 ± 0.018	0.438 ± 0.408
$Experiment_4$	Director	0.711 ± 0.346	0.2621 ± 0.407	0.276 ± 0.403	0.231 ± 0.367	0.258 ± 0.404	0.239 ± 0.378
	Genre	0.978 ± 0.056	0.6555 ± 0.467	0.669 ± 0.458	0.566 ± 0.415	0.553 ± 0.443	0.545 ± 0.424
	Mpaa rating	0.9 ± 0.316	0.7065 ± 0.468	0.715 ± 0.456	0.647 ± 0.464	0.706 ± 0.468	0.663 ± 0.458
	Title	0.987 ± 0.018	0.9868 ± 0.018	0.987 ± 0.018	0.798 ± 0.183	0.987 ± 0.018	0.861 ± 0.124
$Experiment_5$	Director	0.798 ± 0.397	0.7037 ± 0.462	0.702 ± 0.459	0.108 ± 0.3	0.698 ± 0.459	0.128 ± 0.295
	Genre	0.993 ± 0.015	0.9991 ± 0.002	0.996 ± 0.008	0.439 ± 0.426	0.992 ± 0.015	0.49 ± 0.423
	Mpaa rating	0.956 ± 0.14	0.9537 ± 0.139	0.955 ± 0.14	0.861 ± 0.311	0.954 ± 0.139	0.867 ± 0.294
	Title	0.987 ± 0.018	0.9869 ± 0.018	0.987 ± 0.018	0.427 ± 0.436	0.987 ± 0.018	0.373 ± 0.41

Table 15 continued

Method name	$a \in A$	$\mu_{P_1} \pm s_{P_1}$	$\mu_{R_1} \pm s_{R_1}$	$\mu_{F_1} \pm s_{F_1}$	$\mu_{P_2} \pm s_{P_2}$	$\mu_{R_2} \pm s_{R_2}$	$\mu_{F_2} \pm s_{F_2}$
$Experiment_6$	Director	0.876 ± 0.307	0.7895 ± 0.412	0.788 ± 0.408	0.366 ± 0.374	0.778 ± 0.406	0.425 ± 0.369
	Genre	0.986 ± 0.037	0.9874 ± 0.037	0.987 ± 0.037	0.716 ± 0.343	0.985 ± 0.036	0.771 ± 0.31
	Mpaa rating	0.956 ± 0.14	0.9535 ± 0.139	0.955 ± 0.14	0.86 ± 0.311	0.954 ± 0.139	0.866 ± 0.294
	Title	0.992 ± 0.012	0.9798 ± 0.038	0.985 ± 0.022	0.698 ± 0.366	0.98 ± 0.038	0.738 ± 0.328
Hao et al. [28] IE	Director	0.75 ± 0.11	0.8 ± 0.12	0.77 ± 0.12	–	–	–
	Genre	0.96 ± 0.04	0.91 ± 0.04	0.93 ± 0.04	–	–	–
	Mpaa rating	0.78 ± 0.23	0.75 ± 0.23	0.76 ± 0.23	–	–	–
	Title	0.71 ± 0.25	0.68 ± 0.25	0.69 ± 0.25	–	–	–

The analysis is based on the attribute level, i.e. separate statistics for each attribute

Table 16 The comparison of the mean (μ) and standard deviation (s) values of precision (P_3, P_4), recall (R_3, R_4), and F-measure (F_3, F_4) obtained from the BigGrams system which worked on the HTML tags level with the different configurations ($Experiment_1$–$Experiment_6$) for the *IS* about movies (verticle: movies)

Method name	$a \in A$	$\mu_{P_3} \pm s_{P_3}$	$\mu_{R_3} \pm s_{R_3}$	$\mu_{F_3} \pm s_{F_3}$	$\mu_{P_4} \pm s_{P_4}$	$\mu_{R_4} \pm s_{R_4}$	$\mu_{F_4} \pm s_{F_4}$
$Experiments_1$	Director	0.614 ± 0.118	0.701 ± 0.126	0.624 ± 0.1	0.772 ± 0.292	0.693 ± 0.157	0.671 ± 0.134
	Genre						
	Mpaa rating						
	Title						
$Experiments_2$	Director	0.553 ± 0.169	0.557 ± 0.176	0.549 ± 0.169	0.983 ± 0.018	0.61 ± 0.12	0.746 ± 0.097
	Genre						
	Mpaa rating						
	Title						
$Experiments_3$	Director	0.423 ± 0.206	0.93 ± 0.096	0.46 ± 0.198	0.219 ± 0.168	0.922 ± 0.143	0.317 ± 0.173
	Genre						
	Mpaa rating						
	Title						
$Experiments_4$	Director	0.56 ± 0.202	0.626 ± 0.218	0.577 ± 0.201	0.783 ± 0.16	0.69 ± 0.196	0.71 ± 0.119
	Genre						
	Mpaa rating						
	Title						
$Experiments_5$	Director	0.436 ± 0.2	0.932 ± 0.097	0.464 ± 0.2	0.229 ± 0.18	0.922 ± 0.143	0.328 ± 0.188
	Genre						
	Mpaa rating						
	Title						
$Experiments_6$	Director	0.66 ± 0.16	0.924 ± 0.099	0.7 ± 0.146	0.424 ± 0.239	0.904 ± 0.146	0.537 ± 0.212
	Genre						
	Mpaa rating						
	Title						

The analysis is based on the group of attributes, i.e. statistics for the group of attributes

Table 17 The comparison of the mean (μ) and standard deviation (s) values of precision (P_1, P_2), recall (R_1, R_2), and F-measure (F_1, F_2) obtained from the Hao et al. [28] IE, and the BigGrams system which worked on the HTML tags level with the different configurations ($Experiment_1$–$Experiment_6$) for the *IS* about NBA players (verticle: NBA players)

Method name	$a \in A$	$\mu_{P_1} \pm s_{P_1}$	$\mu_{R_1} \pm s_{R_1}$	$\mu_{F_1} \pm s_{F_1}$	$\mu_{P_2} \pm s_{P_2}$	$\mu_{R_2} \pm s_{R_2}$	$\mu_{F_2} \pm s_{F_2}$
$Experiment_1$	Height	1 ± 0	1 ± 0	1 ± 0	0.832 ± 0.36	1 ± 0	0.849 ± 0.333
	Name	1 ± 0	0.9718 ± 0.089	0.984 ± 0.052	0.901 ± 0.233	0.972 ± 0.089	0.915 ± 0.192
	Team	0.631 ± 0.448	0.6302 ± 0.448	0.63 ± 0.448	0.128 ± 0.307	0.63 ± 0.448	0.152 ± 0.301
	Weight	1 ± 0.001	0.9998 ± 0.001	1 ± 0.001	0.663 ± 0.454	1 ± 0.001	0.689 ± 0.436
$Experiment_2$	Height	0.6 ± 0.516	0.6 ± 0.516	0.6 ± 0.516	0.6 ± 0.516	0.6 ± 0.516	0.6 ± 0.516
	Name	0.9 ± 0.316	0.8718 ± 0.319	0.884 ± 0.315	0.858 ± 0.316	0.872 ± 0.319	0.863 ± 0.316
	Team	0.617 ± 0.449	0.617 ± 0.449	0.617 ± 0.449	0.128 ± 0.307	0.617 ± 0.449	0.152 ± 0.301
	Weight	0.8 ± 0.422	0.5986 ± 0.504	0.607 ± 0.501	0.508 ± 0.464	0.599 ± 0.504	0.534 ± 0.468
$Experiment_3$	Height	1 ± 0	1 ± 0	1 ± 0	0.82 ± 0.381	1 ± 0	0.834 ± 0.355
	Name	1 ± 0	1 ± 0	1 ± 0	0.851 ± 0.336	1 ± 0	0.868 ± 0.319
	Team	1 ± 0	1 ± 0	1 ± 0	0.408 ± 0.509	1 ± 0	0.416 ± 0.503
	Weight	1 ± 0.001	0.9998 ± 0.001	1 ± 0.001	0.66 ± 0.46	1 ± 0.001	0.684 ± 0.444
$Experiment_4$	Height	0.6 ± 0.516	0.6 ± 0.516	0.6 ± 0.516	0.6 ± 0.516	0.6 ± 0.516	0.6 ± 0.516
	Name	0.9 ± 0.316	0.8395 ± 0.351	0.857 ± 0.33	0.825 ± 0.347	0.84 ± 0.351	0.83 ± 0.347
	Team	1 ± 0	1 ± 0	1 ± 0	0.473 ± 0.478	1 ± 0	0.505 ± 0.471
	Weight	0.8 ± 0.422	0.5986 ± 0.504	0.607 ± 0.501	0.531 ± 0.462	0.599 ± 0.504	0.554 ± 0.472
$Experiment_5$	Height	1 ± 0	1 ± 0	1 ± 0	0.825 ± 0.373	1 ± 0	0.841 ± 0.345
	Name	1 ± 0	1 ± 0	1 ± 0	0.901 ± 0.315	1 ± 0	0.901 ± 0.313
	Team	1 ± 0	1 ± 0	1 ± 0	0.41 ± 0.508	1 ± 0	0.414 ± 0.504
	Weight	1 ± 0.001	0.9998 ± 0.001	1 ± 0.001	0.663 ± 0.454	1 ± 0.001	0.689 ± 0.436

Table 17 continued

Method name	$a \in A$	$\mu_{P_1} \pm s_{P_1}$	$\mu_{R_1} \pm s_{R_1}$	$\mu_{F_1} \pm s_{F_1}$	$\mu_{P_2} \pm s_{P_2}$	$\mu_{R_2} \pm s_{R_2}$	$\mu_{F_2} \pm s_{F_2}$
$Experiment_6$	Height	0.9 ± 0.316	0.9 ± 0.316	0.9 ± 0.316	0.725 ± 0.447	0.9 ± 0.316	0.741 ± 0.429
	Name	1 ± 0	0.9395 ± 0.191	0.957 ± 0.137	0.853 ± 0.316	0.94 ± 0.191	0.855 ± 0.312
	Team	0.884 ± 0.315	0.884 ± 0.315	0.884 ± 0.315	0.311 ± 0.476	0.884 ± 0.315	0.322 ± 0.469
	Weight	1 ± 0.001	0.9027 ± 0.307	0.906 ± 0.298	0.66 ± 0.459	0.903 ± 0.307	0.681 ± 0.449
Hao et al. [28] IE	Height	0.76 ± 0.19	0.67 ± 0.17	0.71 ± 0.18	–	–	–
	Name	0.84 ± 0.24	0.82 ± 0.23	0.83 ± 0.23	–	–	–
	Team	0.82 ± 0.09	0.82 ± 0.09	0.82 ± 0.09	–	–	–
	Weight	0.91 ± 0.1	0.91 ± 0.1	0.91 ± 0.1	–	–	–

The analysis is based on the attribute level, i.e. separate statistics for each attribute

Table 18 The comparison of the mean (μ) and standard deviation (s) values of precision (P_3, P_4), recall (R_3, R_4), and F-measure (F_3, F_4) obtained from the BigGrams system which worked on the HTML tags level with the different configurations (*Experiment*$_1$–*Experiment*$_6$) for the *IS* about NBA players (verticle: NBA players)

Method name	$a \in A$	$\mu_{P_3} \pm s_{P_3}$	$\mu_{R_3} \pm s_{R_3}$	$\mu_{F_3} \pm s_{F_3}$	$\mu_{P_4} \pm s_{P_4}$	$\mu_{R_4} \pm s_{R_4}$	$\mu_{F_4} \pm s_{F_4}$
Experiments$_1$	Height	0.631 ± 0.183	0.9 ± 0.108	0.651 ± 0.17	0.606 ± 0.418	0.959 ± 0.067	0.663 ± 0.353
	Name						
	Team						
	Weight						
Experiments$_2$	Height	0.524 ± 0.242	0.672 ± 0.282	0.537 ± 0.243	0.824 ± 0.258	0.808 ± 0.237	0.768 ± 0.24
	Name						
	Team						
	Weight						
Experiments$_3$	Height	0.685 ± 0.246	1 ± 0	0.701 ± 0.236	0.549 ± 0.437	1 ± 0	0.607 ± 0.401
	Name						
	Team						
	Weight						
Experiments$_4$	Height	0.607 ± 0.239	0.76 ± 0.236	0.622 ± 0.24	0.743 ± 0.369	0.804 ± 0.259	0.666 ± 0.322
	Name						
	Team						
	Weight						
Experiments$_5$	Height	0.699 ± 0.23	1 ± 0	0.711 ± 0.224	0.556 ± 0.426	1 ± 0	0.62 ± 0.389
	Name						
	Team						
	Weight						
Experiments$_6$	Height	0.637 ± 0.184	0.907 ± 0.164	0.649 ± 0.175	0.6 ± 0.396	0.931 ± 0.148	0.631 ± 0.327
	Name						
	Team						
	Weight						

The analysis is based on the group of attributes, i.e. statistics for the group of attributes

Table 19 The comparison of the mean (μ) and standard deviation (s) values of precision (P_1, P_2), recall (R_1, R_2), and F-measure (F_1, F_2) obtained from the Hao et al. [28] IE, and the BigGrams system which worked on the HTML tags level with the different configurations ($Experiment_1$–$Experiment_6$) for the *IS* about restaurants (verticle: restaurants)

Method name	$a \in A$	$\mu_{P_1} \pm s_{P_1}$	$\mu_{R_1} \pm s_{R_1}$	$\mu_{F_1} \pm s_{F_1}$	$\mu_{P_2} \pm s_{P_2}$	$\mu_{R_2} \pm s_{R_2}$	$\mu_{F_2} \pm s_{F_2}$
$Experiment_1$	Address	0.999 ± 0.004	0.8063 ± 0.404	0.814 ± 0.39	0.579 ± 0.42	0.593 ± 0.376	0.536 ± 0.362
	Cuisine	0.8 ± 0.422	0.613 ± 0.5	0.624 ± 0.487	0.454 ± 0.465	0.613 ± 0.5	0.483 ± 0.458
	Name	1 ± 0	0.9666 ± 0.104	0.98 ± 0.062	0.64 ± 0.295	0.967 ± 0.104	0.727 ± 0.236
	Phone	1 ± 0.001	0.9588 ± 0.087	0.977 ± 0.049	0.828 ± 0.295	0.959 ± 0.087	0.855 ± 0.25
$Experiment_2$	Address	0.899 ± 0.316	0.7064 ± 0.469	0.714 ± 0.459	0.689 ± 0.459	0.497 ± 0.398	0.551 ± 0.399
	Cuisine	0.488 ± 0.515	0.1922 ± 0.394	0.197 ± 0.392	0.192 ± 0.394	0.192 ± 0.394	0.192 ± 0.394
	Name	1 ± 0	0.9666 ± 0.104	0.98 ± 0.062	0.737 ± 0.242	0.967 ± 0.104	0.812 ± 0.17
	Phone	0.777 ± 0.416	0.7346 ± 0.411	0.753 ± 0.41	0.726 ± 0.405	0.735 ± 0.411	0.729 ± 0.407
$Experiment_3$	Address	1 ± 0	0.9978 ± 0.006	0.999 ± 0.003	0.51 ± 0.336	0.795 ± 0.265	0.54 ± 0.281
	Cuisine	0.989 ± 0.025	0.9996 ± 0.001	0.994 ± 0.013	0.472 ± 0.434	0.989 ± 0.025	0.531 ± 0.402
	Name	1 ± 0	0.9995 ± 0.001	1 ± 0.001	0.399 ± 0.371	0.999 ± 0.001	0.483 ± 0.368
	Phone	0.997 ± 0.008	0.9995 ± 0.001	0.998 ± 0.004	0.856 ± 0.275	0.997 ± 0.008	0.886 ± 0.232
$Experiment_4$	Address	0.9 ± 0.316	0.8978 ± 0.316	0.899 ± 0.316	0.839 ± 0.326	0.606 ± 0.354	0.654 ± 0.311
	Cuisine	0.684 ± 0.474	0.3904 ± 0.488	0.399 ± 0.484	0.339 ± 0.428	0.358 ± 0.45	0.345 ± 0.435
	Name	1 ± 0	0.9995 ± 0.001	1 ± 0.001	0.736 ± 0.293	0.999 ± 0.001	0.811 ± 0.214
	Phone	0.7 ± 0.483	0.5696 ± 0.472	0.603 ± 0.46	0.563 ± 0.471	0.569 ± 0.471	0.565 ± 0.471
$Experiment_5$	Address	1 ± 0	0.9978 ± 0.006	0.999 ± 0.003	0.708 ± 0.325	0.71 ± 0.298	0.631 ± 0.27
	Cuisine	0.997 ± 0.01	0.9996 ± 0.001	0.998 ± 0.005	0.535 ± 0.452	0.996 ± 0.01	0.582 ± 0.422
	Name	1 ± 0	0.9995 ± 0.001	1 ± 0.001	0.426 ± 0.347	0.999 ± 0.001	0.525 ± 0.328
	Phone	1 ± 0.001	0.9996 ± 0.001	1 ± 0.001	0.887 ± 0.264	1 ± 0.001	0.908 ± 0.227
$Experiment_6$	Address	0.8 ± 0.422	0.7978 ± 0.42	0.799 ± 0.421	0.659 ± 0.41	0.518 ± 0.384	0.529 ± 0.348
	Cuisine	0.995 ± 0.011	0.9978 ± 0.006	0.996 ± 0.007	0.674 ± 0.398	0.988 ± 0.022	0.723 ± 0.354
	Name	0.999 ± 0.002	0.999 ± 0.002	0.999 ± 0.002	0.523 ± 0.36	0.999 ± 0.002	0.619 ± 0.315
	Phone	1 ± 0.001	0.9961 ± 0.011	0.998 ± 0.005	0.883 ± 0.263	0.996 ± 0.011	0.904 ± 0.226

Table 19 continued

Method name	$a \in A$	$\mu_{P_1} \pm s_{P_1}$	$\mu_{R_1} \pm s_{R_1}$	$\mu_{F_1} \pm s_{F_1}$	$\mu_{P_2} \pm s_{P_2}$	$\mu_{R_2} \pm s_{R_2}$	$\mu_{F_2} \pm s_{F_2}$
Hao et al. [28] IE	Address	0.97 ± 0.02	0.96 ± 0.02	0.96 ± 0.02	–	–	–
	Cuisine	0.98 ± 0.07	0.94 ± 0.06	0.96 ± 0.06	–	–	–
	Name	0.95 ± 0.08	0.89 ± 0.07	0.92 ± 0.07	–	–	–
	Phone	1 ± 0	0.98 ± 0.01	0.99 ± 0	–	–	–

The analysis is based on the attribute level, i.e. separate statistics for each attribute

Table 20 The comparison of the mean (μ) and standard deviation (s) values of precision (P_3, P_4), recall (R_3, R_4), and F-measure (F_3, F_4) obtained from the BigGrams system which worked on the HTML tags level with the different configurations ($Experiment_1$–$Experiment_6$) for the *IS* about restaurants (verticle: restaurants)

Method name	$a \in A$	$\mu_{P_3} \pm s_{P_3}$	$\mu_{R_3} \pm s_{R_3}$	$\mu_{F_3} \pm s_{F_3}$	$\mu_{P_4} \pm s_{P_4}$	$\mu_{R_4} \pm s_{R_4}$	$\mu_{F_4} \pm s_{F_4}$
$Experiments_1$	Address	0.625 ± 0.161	0.783 ± 0.216	0.65 ± 0.156	0.769 ± 0.182	0.828 ± 0.191	0.771 ± 0.125
	Cuisine						
	Name						
	Phone						
$Experiments_2$	Address	0.586 ± 0.208	0.598 ± 0.221	0.571 ± 0.197	0.869 ± 0.155	0.704 ± 0.247	0.74 ± 0.156
	Cuisine						
	Name						
	Phone						
$Experiments_3$	Address	0.559 ± 0.157	0.945 ± 0.069	0.61 ± 0.143	0.547 ± 0.209	0.923 ± 0.113	0.657 ± 0.182
	Cuisine						
	Name						
	Phone						
$Experiments_4$	Address	0.619 ± 0.246	0.633 ± 0.194	0.594 ± 0.201	0.808 ± 0.177	0.728 ± 0.2	0.735 ± 0.146
	Cuisine						
	Name						
	Phone						
$Experiments_5$	Address	0.639 ± 0.115	0.926 ± 0.076	0.662 ± 0.128	0.585 ± 0.16	0.915 ± 0.109	0.696 ± 0.13
	Cuisine						
	Name						
	Phone						
$Experiments_6$	Address	0.685 ± 0.131	0.875 ± 0.096	0.694 ± 0.131	0.625 ± 0.208	0.828 ± 0.184	0.677 ± 0.136
	Cuisine						
	Name						
	Phone						

The analysis is based on the group of attributes, i.e. statistics for the group of attributes

Table 21 The comparison of the mean (μ) and standard deviation (s) values of precision (P_1, P_2), recall (R_1, R_2), and F-measure (F_1, F_2) obtained from the Hao et al. [28] IE, and the BigGrams system which worked on the HTML tags level with the different configurations (*Experiment*$_1$–*Experiment*$_6$) for the *IS* about universities (verticle: universities)

Method name	$a \in A$	$\mu_{P_1} \pm s_{P_1}$	$\mu_{R_1} \pm s_{R_1}$	$\mu_{F_1} \pm s_{F_1}$	$\mu_{P_2} \pm s_{P_2}$	$\mu_{R_2} \pm s_{R_2}$	$\mu_{F_2} \pm s_{F_2}$
Experiment$_1$	Name	0.954 ± 0.146	0.9539 ± 0.146	0.954 ± 0.146	0.606 ± 0.427	0.954 ± 0.146	0.661 ± 0.39
	Phone	0.999 ± 0.005	0.9985 ± 0.005	0.999 ± 0.005	0.85 ± 0.317	0.938 ± 0.131	0.872 ± 0.272
	Type	1 ± 0	1 ± 0	1 ± 0	0.908 ± 0.159	1 ± 0	0.938 ± 0.106
	Website	0.2 ± 0.422	0.2 ± 0.422	0.2 ± 0.422	0.15 ± 0.337	0.2 ± 0.422	0.167 ± 0.36
Experiment$_2$	Name	0.737 ± 0.431	0.7329 ± 0.429	0.735 ± 0.43	0.591 ± 0.44	0.733 ± 0.429	0.63 ± 0.424
	Phone	0.7 ± 0.483	0.556 ± 0.488	0.575 ± 0.497	0.556 ± 0.488	0.556 ± 0.489	0.556 ± 0.488
	Type	0.692 ± 0.478	0.5723 ± 0.49	0.587 ± 0.491	0.566 ± 0.488	0.572 ± 0.49	0.568 ± 0.488
	Website	0.2 ± 0.422	0.2 ± 0.422	0.2 ± 0.422	0.2 ± 0.422	0.2 ± 0.422	0.2 ± 0.422
Experiment$_3$	Name	1 ± 0	1 ± 0	1 ± 0	0.494 ± 0.456	1 ± 0	0.545 ± 0.429
	Phone	1 ± 0	1 ± 0	1 ± 0	0.787 ± 0.354	0.947 ± 0.114	0.825 ± 0.302
	Type	1 ± 0	1 ± 0	1 ± 0	0.829 ± 0.317	1 ± 0	0.858 ± 0.292
	Website	1 ± 0	1 ± 0	1 ± 0	0.872 ± 0.181	1 ± 0	0.909 ± 0.126
Experiment$_4$	Name	0.831 ± 0.363	0.8273 ± 0.362	0.829 ± 0.362	0.55 ± 0.399	0.827 ± 0.362	0.619 ± 0.364
	Phone	0.8 ± 0.422	0.656 ± 0.464	0.675 ± 0.468	0.656 ± 0.464	0.609 ± 0.449	0.623 ± 0.448
	Type	0.592 ± 0.51	0.4719 ± 0.494	0.487 ± 0.499	0.466 ± 0.491	0.472 ± 0.494	0.468 ± 0.492
	Website	0.9 ± 0.316	0.8085 ± 0.404	0.816 ± 0.391	0.782 ± 0.396	0.808 ± 0.404	0.791 ± 0.398
Experiment$_5$	Name	1 ± 0	1 ± 0	1 ± 0	0.524 ± 0.43	1 ± 0	0.588 ± 0.394
	Phone	1 ± 0	1 ± 0	1 ± 0	0.845 ± 0.33	0.947 ± 0.114	0.866 ± 0.287
	Type	1 ± 0	1 ± 0	1 ± 0	0.908 ± 0.159	1 ± 0	0.938 ± 0.106
	Website	1 ± 0	1 ± 0	1 ± 0	0.822 ± 0.208	1 ± 0	0.876 ± 0.142

Table 21 continued

Method name	$a \in A$	$\mu_{P_1} \pm s_{P_1}$	$\mu_{R_1} \pm s_{R_1}$	$\mu_{F_1} \pm s_{F_1}$	$\mu_{P_2} \pm s_{P_2}$	$\mu_{R_2} \pm s_{R_2}$	$\mu_{F_2} \pm s_{F_2}$
Experiment$_6$	Name	0.931 ± 0.218	0.9273 ± 0.217	0.929 ± 0.217	0.554 ± 0.387	0.927 ± 0.217	0.628 ± 0.336
	Phone	0.817 ± 0.389	0.8167 ± 0.389	0.817 ± 0.389	0.816 ± 0.389	0.808 ± 0.405	0.811 ± 0.399
	Type	1 ± 0	1 ± 0	1 ± 0	0.908 ± 0.159	1 ± 0	0.938 ± 0.106
	Website	1 ± 0	1 ± 0	1 ± 0	0.841 ± 0.212	1 ± 0	0.889 ± 0.145
Hao et al. [28] IE	Name	0.97 ± 0.05	0.95 ± 0.06	0.96 ± 0.06	–	–	–
	Phone	0.79 ± 0.12	0.78 ± 0.12	0.79 ± 0.12	–	–	–
	Type	0.7 ± 0.29	0.68 ± 0.27	0.69 ± 0.28	–	–	–
	Website	0.96 ± 0.09	0.83 ± 0.08	0.89 ± 0.08	–	–	–

The analysis is based on the attribute level, i.e. separate statistics for each attribute

The author constructed four different rankings of IE methods based on the Hellinger-TOPSIS multi-criteria evaluation method, which utilized mean μ and standard deviation s of each F-measure, i.e. $\mu_{F_1}, s_{F_1}, \mu_{F_2}, s_{F_2}, \mu_{F_3}, s_{F_3}$ and μ_{F_4}, s_{F_4} (Tables 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22). Table 23 presents the created rankings.

Table 23 presents the position of each tested IE methods in the ranking. We may see that the ranking based on F_1 is different than rankings based on F_2, F_3, and it is similar to the ranking based on F_4 (in terms of a reverse order). Furthermore, the rankings based on the F_2 and F_3 are almost the same, and they are different from the ranking that is based on F_4. This observation is confirmed by Spearman's rank correlation coefficient. Table 24 presents the values of Spearman's rank (r_s) and their p-values.

Based on the r_s values from Table 24 we cannot reject the null hypothesis H_{S0} for the cases such as (1) F_1 ranking versus F_2, F_3, (2) F_2 and F_3 rankings versus F_4. In these cases, there is no monotonic association between pairs of rankings mentioned above. On the other hand, we can reject the null hypothesis H_{S0} for the cases such as (1) F_1 ranking versus F_4, (2) F_2 versus F_3. Furthermore, in these cases, there is a high value of r_s near the -1 and 1. This indicates a strong negative and positive relationship between the pairs of rankings. In the first case, that is, the higher we ranked in F_1, the lower we ranked in F_4 ranking and vice versa. In the second case, the higher we ranked in F_2, the higher we ranked in F_3 ranking and vice versa.

In the next step of the analysis, the author computed the p-value of Wilcoxon–Mann–Whitney statistic test and created the boxplots of the F-measure means and the F-measure standard deviations in terms of mean for considered IE methods, separately for each established ranking from Table 23.

Table 25 contains the p-values of Wilcoxon–Mann–Whitney tests for the pairs of methods from ranking that is based on F_1 (Table 23). Figure 15 presents the boxplots of the μ_{F_1} and s_{F_1} in terms of the mean for seven IE methods.

Table 25 and Fig. 15 show that the best results were achieved when two strategies were used, i.e. the strategy that was based on the cleaning value of A HREF HTML tags and utilized all created IE patterns, and the strategy that was based on HTML tags without the attribute value and utilized all created IE patterns. Furthermore, these strategies depended on the term of conducted Wilcoxon–Mann–Whitney tests. In this case, we cannot reject the null hypothesis H_{W0}. Moreover, we can notice that four BigGrams configurations gave better results than Hao et al. [28] IE approach. Three configurations gave independent results from Hao et al. [28] IE approach. The configuration that was based on the HTML tags with attribute and values, and utilized all created IE patterns gave the same results in the term of the Wilcoxon–Mann–Whitney test. In this case, we can not reject the null hypothesis H_{W0}.

Table 26 contains the p-values of Wilcoxon–Mann–Whitney tests for the pairs of methods from the ranking that is based on F_2 (Table 23). Figure 16 presents the boxplots of the μ_{F_2} and s_{F_2} in terms of the mean for six IE methods.

Table 26 and Fig. 16 show that the best result was achieved when the strategy based on the cleaning value of A HREF HTML tags was used, and when it utilized all created IE patterns, and used more than one HTML tags to create patterns. Furthermore, this strategy gave similar results to four out of the five remaining configurations in the term of conducted Wilcoxon–Mann–Whitney tests. For these cases, we cannot reject the null hypothesis H_{W0}. Only the configuration which is based on HTML tags with attribute and values and utilized the unique IE patterns gave different results, i.e. we can reject the null hypothesis H_{W0}.

Table 27 contains the p-values of Wilcoxon–Mann–Whitney tests for the pairs of methods from ranking that is based on F_3 (Table 23). Figure 17 presents the boxplots of the μ_{F_3} and s_{F_3} in terms of the mean for six IE methods.

Table 22 The comparison of the mean (μ) and standard deviation (s) values of precision (P_3, P_4), recall (R_3, R_4), and F-measure (F_3, F_4) obtained from the BigGrams system which worked on the HTML tags level with the different configurations ($Experiment_1$–$Experiment_6$) for the *IS* about universities (verticle: universities)

Method name	$a \in A$	$\mu_{P_3} \pm s_{P_3}$	$\mu_{R_3} \pm s_{R_3}$	$\mu_{F_3} \pm s_{F_3}$	$\mu_{P_4} \pm s_{P_4}$	$\mu_{R_4} \pm s_{R_4}$	$\mu_{F_4} \pm s_{F_4}$
$Experiments_1$	Name	0,628 ± 0,174	0,773 ± 0,127	0,659 ± 0,165	0,656 ± 0,327	0,701 ± 0,17	0,651 ± 0,235
	Phone						
	Type						
	Website						
$Experiments_2$	Name	0,478 ± 0,252	0,515 ± 0,273	0,489 ± 0,257	0,738 ± 0,375	0,462 ± 0,299	0,53 ± 0,289
	Phone						
	Type						
	Website						
$Experiments_3$	Name	0,746 ± 0,199	0,987 ± 0,029	0,784 ± 0,175	0,638 ± 0,266	0,968 ± 0,068	0,741 ± 0,216
	Phone						
	Type						
	Website						
$Experiments_4$	Name	0,614 ± 0,193	0,679 ± 0,208	0,625 ± 0,186	0,783 ± 0,23	0,688 ± 0,222	0,703 ± 0,181
	Phone						
	Type						
	Website						
$Experiments_5$	Name	0,775 ± 0,165	0,987 ± 0,029	0,817 ± 0,136	0,655 ± 0,242	0,968 ± 0,068	0,761 ± 0,185
	Phone						
	Type						
	Website						
$Experiments_6$	Name	0,78 ± 0,163	0,934 ± 0,143	0,817 ± 0,145	0,74 ± 0,226	0,877 ± 0,262	0,775 ± 0,203
	Phone						
	Type						
	Website						

The analysis is based on the group of attributes, i.e. statistics for the group of attributes

Table 23 The ranking of different IE methods, i.e. the BigGrams system, which worked on the HTML tags level with the different configurations ($Experiment_1$–$Experiment_6$) and Hao et al. [28] IE

Method	Rank based on F_1	Rank based on F_2	Rank based on F_3	Rank based on F_4
$Experiment_1$	3	2	3	3
$Experiment_2$	7	6	6	2
$Experiment_3$	2	4	4	6
$Experiment_4$	6	5	5	1
$Experiment_5$	1	3	2	5
$Experiment_6$	4	1	1	4
Hao et al. [28] IE	5	–	–	–

The ranking is based on the Hellinger-TOPSIS method that utilized mean (μ) and standard deviation (s) of different F-measures (F_1, F_2, F_3, and F_4)

Table 24 The statistical comparison of the rankings from Table 23, i.e. the values of Spearman's rank (r_s) and their p-values

Compared ranks	r_s	p-value
Rank based on F_1 versus Rank based on F_2	0.49	0.3287
Rank based on F_1 versus Rank based on F_3	0.60	0.2080
Rank based on F_1 versus Rank based on F_4	−0.83	0.0416
Rank based on F_2 versus Rank based on F_3	0.94	0.0048
Rank based on F_2 versus Rank based on F_4	−0.43	0.3965
Rank based on F_3 versus Rank based on F_4	−0.54	0.2657

Table 25 The p-values of the Wilcoxon–Mann–Whitney test of the IE methods, i.e. the BigGrams system which worked on the HTML tags level with the different configurations ($Experiment_1$–$Experiment_6$), and Hao et al. [28] IE ordered by the rank position, which is based on F_1 from Table 23

Compared experiments based on F_1	p-value
$Experiment_5$ versus $Experiment_3$	0.629
$Experiment_5$ versus $Experiment_1$	0.001254
$Experiment_5$ versus $Experiment_6$	5.309e−05
$Experiment_5$ versus Hao et al. [28] IE	4.29e−10
$Experiment_5$ versus $Experiment_4$	3.871e−08
$Experiment_5$ versus $Experiment_2$	1.27e−09
$Experiment_3$ versus $Experiment_1$	0.00453
$Experiment_3$ versus $Experiment_6$	0.0003088
$Experiment_3$ versus Hao et al. [28] IE	6.714e−10
$Experiment_3$ versus $Experiment_4$	4.796e−08
$Experiment_3$ versus $Experiment_2$	1.145e−09
$Experiment_1$ versus $Experiment_6$	0.8298
$Experiment_1$ versus Hao et al. [28] IE	0.006402
$Experiment_1$ versus $Experiment_4$	0.002801
$Experiment_1$ versus $Experiment_2$	6.451e−05
$Experiment_6$ versus Hao et al. [28] IE	8.783e−06
$Experiment_6$ versus $Experiment_4$	7.059e−06
$Experiment_6$ versus $Experiment_2$	6.997e−08

Table 25 continued

Compared experiments based on F_1	p-value
Hao et al. [28] IE versus *Experiment*$_4$	0.02283
Hao et al. [28] IE versus *Experiment*$_2$	0.0004232
Experiment$_4$ versus *Experiment*$_2$	0.1973

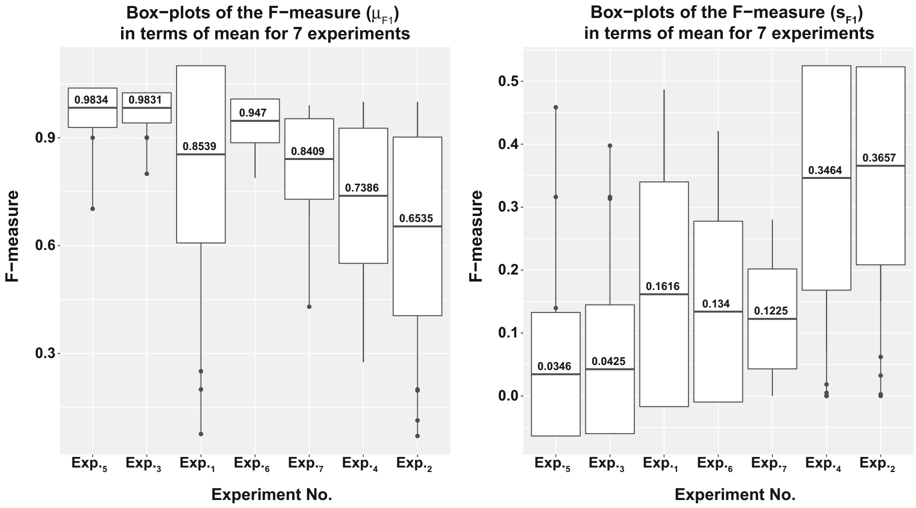

Fig. 15 The boxplots of the μ_{F_1} (the *left plot*) and s_{F_1} (the *right plot*) in terms of the mean for seven IE methods, i.e. the BigGrams system which worked on the HTML tags level with the different configurations (*Experiment*$_1$–*Experiment*$_6$), and Hao et al. [28] IE

Table 26 The p-values of the Wilcoxon–Mann–Whitney test of the IE method, i.e. the BigGrams system, which worked on the HTML tags level with the different configurations (*Experiment*$_1$–*Experiment*$_6$) ordered by the rank position, which is based on F_2 from Table 23

Compared experiments based on F_2	p-value
Experiment$_6$ versus *Experiment*$_1$	0.7677
Experiment$_6$ versus *Experiment*$_5$	0.4767
Experiment$_6$ versus *Experiment*$_3$	0.1657
Experiment$_6$ versus *Experiment*$_4$	0.006159
Experiment$_6$ versus *Experiment*$_2$	0.004419
Experiment$_1$ versus *Experiment*$_5$	0.7069
Experiment$_1$ versus *Experiment*$_3$	0.3204
Experiment$_1$ versus *Experiment*$_4$	0.08205
Experiment$_1$ versus *Experiment*$_2$	0.03618
Experiment$_5$ versus *Experiment*$_3$	0.4481
Experiment$_5$ versus *Experiment*$_4$	0.1616
Experiment$_5$ versus *Experiment*$_2$	0.07511
Experiment$_3$ versus *Experiment*$_4$	0.5707
Experiment$_3$ versus *Experiment*$_2$	0.2879
Experiment$_4$ versus *Experiment*$_2$	0.4127

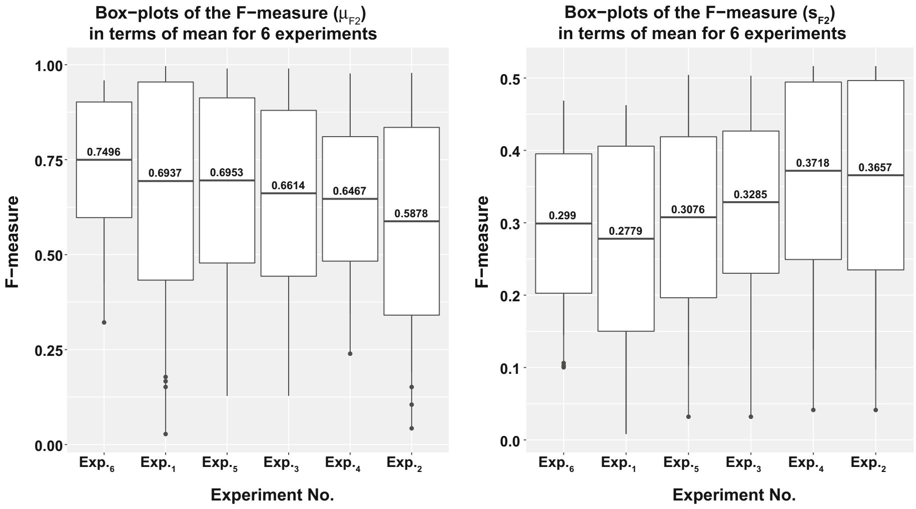

Fig. 16 The boxplots of the μ_{F_2} (the *left plot*) and s_{F_2} (the *right plot*) in terms of the mean for six IE methods, i.e. the BigGrams system which worked on the HTML tags level with the different configurations ($Experiment_1$–$Experiment_6$)

Table 27 The p-values of the Wilcoxon–Mann–Whitney test of the IE method, i.e. the BigGrams system, which worked on the HTML tags level with the different configurations ($Experiment_1$–$Experiment_6$) ordered by the rank position, which is based on F_3 from Table 23

Compared experiments based on F_3	p-value
$Experiment_6$ versus $Experiment_5$	0.6742
$Experiment_6$ versus $Experiment_1$	0.3823
$Experiment_6$ versus $Experiment_3$	0.2786
$Experiment_6$ versus $Experiment_4$	0.03792
$Experiment_6$ versus $Experiment_2$	0.002953
$Experiment_5$ versus $Experiment_1$	0.6454
$Experiment_5$ versus $Experiment_3$	0.5737
$Experiment_5$ versus $Experiment_4$	0.2345
$Experiment_5$ versus $Experiment_2$	0.06496
$Experiment_1$ versus $Experiment_3$	0.7984
$Experiment_1$ versus $Experiment_4$	0.2345
$Experiment_1$ versus $Experiment_2$	0.01476
$Experiment_3$ versus $Experiment_4$	0.5737
$Experiment_3$ versus $Experiment_2$	0.3282
$Experiment_4$ versus $Experiment_2$	0.1049

Table 27 and Fig. 17 show that the best result was achieved when the strategy that was based on the cleaning value of A HREF HTML tags was used, and when it utilized all created IE patterns, and used more than one HTML tags to create patterns. Furthermore, this strategy gave similar results to three out of the five remaining configurations in the term of conducted Wilcoxon–Mann–Whitney tests. For these cases, we cannot reject the null hypothesis H_{W0}. The configurations based on the unique IE patterns gave different results, i.e. we can reject the null hypothesis H_{W0}.

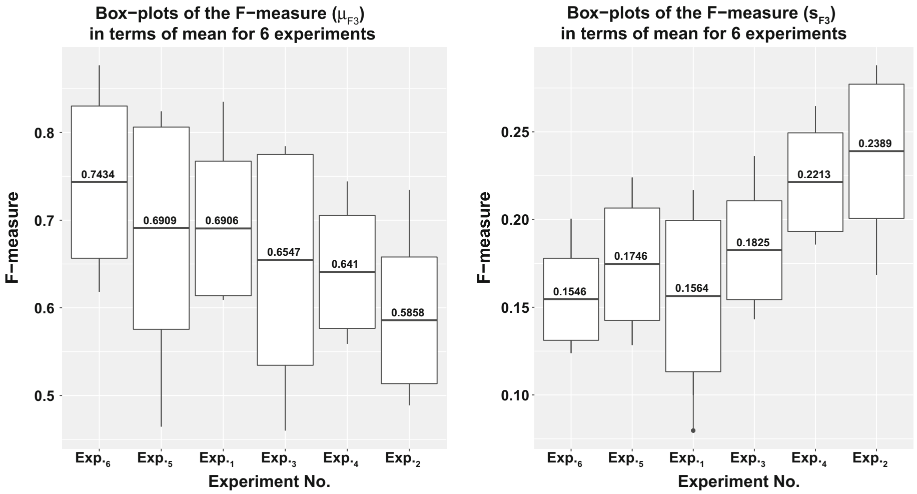

Fig. 17 The boxplots of the μ_{F_3} (the *left plot*) and s_{F_3} (the *right plot*) in terms of the mean for six IE methods, i.e. the BigGrams system which worked on the HTML tags level with the different configurations ($Experiment_1$–$Experiment_6$)

Table 28 The p-values of the Wilcoxon–Mann–Whitney test of the IE method, i.e. the BigGrams system, which worked on the HTML tags level with the different configurations ($Experiment_1$–$Experiment_6$) ordered by the rank position, which is based on F_4 from Table 23

Compared experiments based on F_4	p-value
$Experiment_4$ versus $Experiment_2$	0.7984
$Experiment_4$ versus $Experiment_1$	0.4418
$Experiment_4$ versus $Experiment_6$	0.5737
$Experiment_4$ versus $Experiment_5$	0.1949
$Experiment_4$ versus $Experiment_3$	0.1049
$Experiment_2$ versus $Experiment_1$	0.5054
$Experiment_2$ versus $Experiment_6$	0.8785
$Experiment_2$ versus $Experiment_5$	0.2345
$Experiment_2$ versus $Experiment_3$	0.08298
$Experiment_1$ versus $Experiment_6$	0.8785
$Experiment_1$ versus $Experiment_5$	0.3823
$Experiment_1$ versus $Experiment_3$	0.1949
$Experiment_6$ versus $Experiment_5$	0.5737
$Experiment_6$ versus $Experiment_3$	0.2786
$Experiment_5$ versus $Experiment_3$	0.5737

Table 28 contains the p-values of Wilcoxon–Mann–Whitney tests for the pairs of methods from ranking that is based on F_4 (Table 23). Figure 18 presents the boxplots of the μ_{F_4} and s_{F_4} in terms of the mean for six IE methods.

Table 28 and Fig. 18 show that the best results were achieved when the strategies that were based on the unique IE patterns were used. Furthermore, this strategy gave similar results to five out of the five remaining configurations in terms of conducted Wilcoxon–Mann–Whitney tests. For these cases, we cannot reject the null hypothesis H_{W0}.

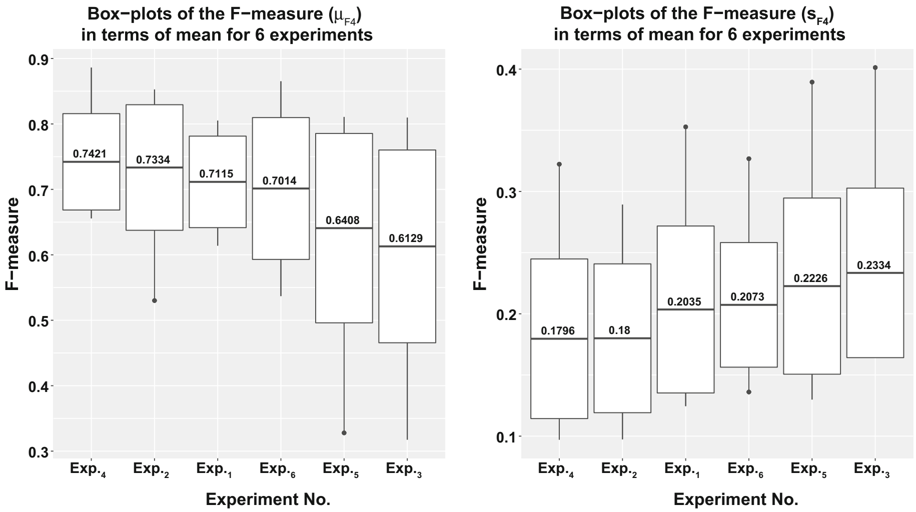

Fig. 18 The boxplots of the μ_{F_4} (the *left plot*) and s_{F_4} (the *right plot*) in terms of the mean for six IE methods, i.e. the BigGrams system which worked on the HTML tags level with different configurations (*Experiment*$_1$–*Experiment*$_6$)

A.5 Summarization

The newest significant findings of these additional experiments and comparisons are as follows:

- the indicators proposed by Hao et al. [28] IE are similar to the indicators such as precision, recall, and F-measure in the domain level (the case where the indicator values in each page are not computed),
- the indicators proposed by Hao et al. [28] IE are too optimistic when we want to measure the value the precision, recall, and F-measure in the domain level. It is better to use a value of the sets intersection ($V_{ref} \cap V_{rec}$) rather than assume 1 if it is matched to at least one of the extracted values to the reference set. Thanks to this, we may receive a better estimate of the indicators value for the BigGrams system,
- strategies that are based on the unique patterns should be applied if we want to achieve good results in the domain level point of view,
- there are no significant difference between the macro-average of indicators measured by the attribute level and the group of attributes,
- the strategy of the pre-processing has the influence on the final IE result,
- the best results for the page-level analysis were achieved when we used the approach that cleaned only the chosen HTML tags.

If we remove too much information from HTML tags, i.e. its attributes and values, the created patterns are too general and extract too much incorrect information. On the other hand, if we use all available information (the HTML tags attributed and their values), the created patterns are too specific and do not extract all available information. Also, the author observed the same behaviour when too short or too long patterns were used. For these reasons, the created system must have an adaptation component to achieve better performance in the future. This element in a dynamic way must (1) choose an appropriate strategy of HTML tags cleaning, (2) determine the best configuration in term of pattern lengths, and balance between these two properties to maximize the value of the indicators. These two properties are new and non-trivial tasks for future research. Furthermore, we must set a better semantic measure of extracted terms to receive better precision. The author observed that for some cases, we might obtain a low precision because the reference set does not include the extracted value, even if this value is semantically correct, and it is valuable for a given website or page. This approach may be accomplished in two ways: (1) using a better-labelled data set or (2) using a "fuzzy match". The process of improving and correction of the data set may be time-consuming; for this reason, the "fuzzy match" should be used. For example, we may utilize a Jaccard similarity measure between extracted and reference value rather than the perfect match strategy.

References

1. Agichtein E, Gravano L (2000) Snowball: extracting relations from large plain-text collections. In: Proceedings of the Fifth ACM conference on digital libraries, DL'00. ACM, New York, pp 85–94. doi:10.1145/336597.336644
2. Ali R, Lee S, Chung TC (2017) Accurate multi-criteria decision making methodology for recommending machine learning algorithm. Expert Syst Appl 71:257–278
3. Banko M, Cafarella MJ, Soderland S, Broadhead M, Etzioni O (2007) Open information extraction from the web. In: Proceedings of the 20th international joint conference on artifical intelligence, IJCAI'07. Morgan Kaufmann Publishers Inc., San Francisco, pp 2670–2676. http://dl.acm.org/citation.cfm?id=1625275.1625705
4. Blohm S (2014) Large-scale pattern-based information extraction from the world wide web. Karlsruher Institut für Technologie. http://www.ebook.de/de/product/18345051/sebastian_blohm_large_scale_pattern_based_information_extraction_from_the_world_wide_web.html, http://d-nb.info/1000088529
5. Brin S (1999) Extracting patterns and relations from the world wide web. In: Selected papers from the international workshop on the world wide web and databases, WebDB '98. Springer-Verlag, pp 172–183. http://dl.acm.org/citation.cfm?id=646543.696220
6. Brin S (November 1999) Extracting patterns and relations from the world wide web. Technical Report 1999-65, Stanford InfoLab. http://ilpubs.stanford.edu:8090/421/, previous number = SIDL-WP-1999-0119
7. Bronzi M, Crescenzi V, Merialdo P, Papotti P (2013) Extraction and integration of partially overlapping web sources. PVLDB 6(10):805–816. http://www.vldb.org/pvldb/vol6/p805-bronzi.pdf
8. Bunescu R, Pasca M (2006) Using encyclopedic knowledge for named entity disambiguation. In: Proceesings of the 11th conference of the European chapter of the association for computational linguistics (EACL-06). Trento, pp 9–16. http://www.cs.utexas.edu/users/ai-lab/?bunescu:eacl06
9. Carlson A, Betteridge J, Hruschka Jr, ER, Mitchell TM (2009) Coupling semi-supervised learning of categories and relations. In: Proceedings of the NAACL HLT 2009 workskop on semi-supervised learning for natural language processing
10. Carlson A, Betteridge J, Kisiel B, Settles B, Hruschka Jr ER, Mitchell TM (2010) Toward an architecture for never-ending language learning. In: Proceedings of the Twenty-Fourth conference on artificial intelligence (AAAI 2010)
11. Chang CH, Kayed M, Girgis M, Shaalan K (2006) A survey of web information extraction systems. IEEE Trans Knowl Data Eng 18(10):1411–1428
12. Chang C, Lui S (2001) IEPAD: information extraction based on pattern discovery. In: Shen VY, Saito N, Lyu MR, Zurko ME (eds) Proceedings of the Tenth International World Wide Web Conference, WWW 10. ACM, Hong Kong, pp 681–688, May 1–5
13. Chiticariu L, Li Y, Reiss FR (2013) Rule-based information extraction is dead! long live rule-based information extraction systems! In: Proceedings of the 2013 conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18–21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL. ACL, pp 827–832. http://aclweb.org/anthology/D/D13/D13-1079.pdf
14. Cimiano P (2006) Ontology learning and population from text: algorithms, evaluation and applications. Springer-Verlag, New York Inc, Secaucus
15. Crescenzi V, Mecca G (2004) Automatic information extraction from large websites. J ACM 51(5):731–779

16. Cucerzan S (2007) Large-scale named entity disambiguation based on wikipedia data. In: Proceedings of the 2007 joint conference on EMNLP and CNLL. pp 708–716
17. Czerski D, Ciesielski K, Dramiński M, Kłopotek M, Łoziński P, Wierzchoń S (2016) What NEKST?—semantic search engine for polish internet. Springer International Publishing, Cham, pp 335–347. doi:10.1007/978-3-319-30165-5_16
18. Dalvi BB, Callan J, Cohen WW (2010) Entity list completion using set expansion techniques. In: Voorhees EM, Buckland LP (eds.) TREC. National Institute of Standards and Technology (NIST). http://dblp.uni-trier.de/db/conf/trec/trec2010.html
19. Dalvi BB, Cohen WW, Callan J (2012) Websets: extracting sets of entities from the web using unsupervised information extraction. In: Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, WSDM '12, ACM, New York, pp 243–252. doi:10.1145/2124295.2124327
20. de Knijff J, Frasincar F, Hogenboom F (2013) Domain taxonomy learning from text: the subsumption method versus hierarchical clustering. Data Knowl Eng 83:54–69
21. Downey DC (2008) Redundancy in web-scale information extraction: probabilistic model and experimental results. University of Washington. http://books.google.pl/books?id=THnZtgAACAAJ
22. Etzioni O, Banko M, Soderland S, Weld DS (2008) Open information extraction from the web. Commun ACM 51(12):68–74. doi:10.1145/1409360.1409378
23. Etzioni O, Cafarella M, Downey D, Popescu AM, Shaked T, Soderland S, Weld DS, Yates A (2005) Unsupervised named-entity extraction from the web: an experimental study. Artif Intell 165(1):91–134. doi:10.1016/j.artint.2005.03.001
24. Ferrara E, Meo PD, Fiumara G, Baumgartner R (2014) Web data extraction, applications and techniques: a survey. Knowl Based Syst 70:301–323
25. Forman G, Scholz M (2010) Apples-to-apples in cross-validation studies: pitfalls in classifier performance measurement. ACM SIGKDD Explor Newsl 12(1):49–57
26. Furche T, Gottlob G, Grasso G, Guo X, Orsi G, Schallhart C, Wang C (2014) DIADEM: thousands of websites to a single database. PVLDB 7(14):1845–1856. http://www.vldb.org/pvldb/vol7/p1845-furche.pdf
27. Haav H (2004) A semi-automatic method to ontology design by using FCA. University of Ostrava, Department of Computer Science
28. Hao Q, Cai R, Pang Y, Zhang L (2011) From one tree to a forest: a unified solution for structured web data extraction. In: Proceedings of the 34th international ACM SIGIR conference on research and development in information retrieval (SIGIR 2011). Association for Computing Machinery, Inc., pp 775–784
29. Harrell Jr F, Dupont C (2013) Hmisc: Harrell miscellaneous. R Package
30. Harris Z (1954) Distributional structure. Word 10(23):146–162
31. He Y, Xin D (2011) SEISA: set expansion by iterative similarity aggregation. In: Srinivasan S, Ramamritham K, Kumar A, Ravindra MP, Bertino E, Kumar R (eds) Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28–April 1, 2011. ACM, pp 427–436
32. Hemnani A, Bressan S (2002) Extracting information from semi-structured web documents. In: Proceedings of the workshops on advances in Object-Oriented Information Systems OOIS'02. Springer-Verlag, London, pp 166–175. http://dl.acm.org/citation.cfm?id=645790.667826
33. Hollander M, Wolfe DA, Chicken E (2013) Nonparametric statistical methods. Wiley, Hoboken
34. Hsu C, Dung M (1998) Generating finite-state transducers for semi-structured data extraction from the web. Inf Syst 23(9):521–538. http://dl.acm.org/citation.cfm?id=306766.306775
35. Jiménez P, Corchuelo R (2016) On learning web information extraction rules with TANGO. Inf Syst 62:74–103
36. Jiménez P, Corchuelo R (2016) Roller: a novel approach to web information extraction. Knowl Inf Syst 49(1):197–241
37. Kang Y, Haghighi PD, Burstein F (2016) Taxofinder: a graph-based approach for taxonomy learning. IEEE Trans Knowl Data Eng 28(2):524–536
38. Karlgren J, Sahlgren M (2001) From words to understanding. In: Uesaka Y, Kanerva P, Asoh H (eds) Foundations of real-world understanding. CSLI Publications, Stanford, pp 294–308
39. Kayed M, Chang C (2010) Fivatech: page-level web data extraction from template pages. IEEE Trans Knowl Data Eng 22(2):249–263
40. Kou G, Lu Y, Peng Y, Shi Y (2012) Evaluation of classification algorithms using MCDM and rank correlation. Int J Inf Technol Decis Mak 11(1):197–225
41. Krohling RA, Lourenzutti R, Campos M (2015) Ranking and comparing evolutionary algorithms with hellinger-topsis. Appl Soft Comput 37:217–226
42. Krohling RA, Pacheco AG (2015) A-topsis-an approach based on topsis for ranking evolutionary algorithms. Procedia Comput Sci 55:308–317

43. Liu B, Zhai Y (2005) NET—a system for extracting web data from flat and nested data records. In: Ngu AHH, Kitsuregawa M, Neuhold EJ, Chung J, Sheng QZ (eds.) Web Information Systems Engineering—WISE 2005, 6th International Conference on Web Information Systems Engineering, New York, November 20–22 2005, Proceedings of the Lecture Notes in Computer Science, vol 3806. Springer, pp 487–495
44. Maedche A, Staab S (2000) Mining ontologies from text. In: Procedia of Knowledge Engineering and Knowledge Management (EKAW 2000). LNAI 1937, Springer
45. Maimon O, Rokach L (2005) Introduction to knowledge discovery in databases. In: Maimon O, Rokach L (eds.) The data mining and knowledge discovery handbook. Springer, pp 1–17. http://dblp.uni-trier.de/db/books/collections/datamining2005.html
46. Manning CD, Raghavan P, Schütze H (2008) Introduction to information retrieval. Cambridge University Press, New York
47. Mironczuk M, Czerski D, Sydow M, Klopotek MA (2013) Language-independent information extraction based on formal concept analysis. In: Informatics and applications (ICIA), 2013 second international conference on, pp 323–329
48. Moens M (2006) Information extraction: algorithms and prospects in a retrieval context (the information retrieval series). Springer International Series on Information Retrieval, Springer, Secaucus. http://books.google.pl/books?id=t5oMg54hBxwC
49. Navigli R, Velardi P, Faralli S (2011) A graph-based algorithm for inducing lexical taxonomies from scratch. In: Walsh T (ed) IJCAI 2011, Proceedings of the 22nd international joint conference on artificial intelligence, Barcelona, Catalonia, July 16–22, 2011. IJCAI/AAAI, pp 1872–1877
50. Park BK, Han H, Song IY (2005) PIES: a web information extraction system using ontology and tag patterns. Springer, Berlin, pp 688–693. doi:10.1007/11563952_65
51. Pasupat P, Liang P (2014) Zero-shot entity extraction from web pages. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22–27, 2014, Baltimore, MD, USA, Volume 1: Long Papers. pp 391–401. http://aclweb.org/anthology/P/P14/P14-1037.pdf
52. Pawlak Z (1981) Information systems theoretical foundations. Inf Syst 6(3):205–218. http://www.sciencedirect.com/science/article/pii/0306437981900235
53. Piskorski J, Yangarber R (2013) Information extraction: Past, present and future. In: Poibeau T, Saggion H, Piskorski J, Yangarber R (eds) Multi-source, multilingual information extraction and summarization. Springer, Berlin, pp 23–49. Theory and Applications of Natural Language Processing
54. Priss U (1996) Formal concept analysis in information science. Annu Rev Inf Sci Technol 40:521–543
55. Qiu D, Barbosa L, Dong XL, Shen Y, Srivastava D (2015) DEXTER: large-scale discovery and extraction of product specifications on the web. PVLDB 8(13):2194–2205. http://www.vldb.org/pvldb/vol8/p2194-qiu.pdf
56. Riloff E, Jones R (1999) Learning dictionaries for information extraction by multi-level bootstrapping. In: Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence. AAAI'99/IAAI '99. American Association for Artificial Intelligence, Menlo Park, pp 474–479. http://dl.acm.org/citation.cfm?id=315149.315364
57. Santafe G, Inza I, Lozano JA (2015) Dealing with the evaluation of supervised classification algorithms. Artif Intell Rev 44(4):467–508
58. Sarawagi S (2008) Information extraction. Found Trends Databases 1(3):261–377. doi:10.1561/1900000003
59. Schoenmackers S (2011) Inference over the web. University of Washington, Seattle
60. Schulz A, Lässig J, Gaedke M (2016) Practical web data extraction: are we there yet?—a short survey. In: 2016 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2016, Omaha, NE, USA, October 13–16 2016. IEEE Computer Society, pp 562–567
61. Sleiman HA, Corchuelo R (2013) A survey on region extractors from web documents. IEEE Trans Knowl Data Eng 25(9):1960–1981
62. Sleiman HA, Corchuelo R (2012) An unsupervised technique to extract information from semi-structured web pages. In: Wang XS, Cruz IF, Delis A, Huang G (eds) Web Information Systems Engineering—WISE 2012—13th International Conference, Paphos, Cyprus, November 28–30, 2012. Proceedings of the Lecture Notes in Computer Science, vol 7651. Springer, pp 631–637
63. Sleiman HA, Corchuelo R (2013) Tex: an efficient and effective unsupervised web information extractor. Knowl Based Syst 39:109–123. http://dblp.uni-trier.de/db/journals/kbs/kbs39.html
64. Sleiman HA, Corchuelo R (2014) A class of neural-network-based transducers for web information extraction. Neurocomputing 135:61–68
65. Sleiman HA, Corchuelo R (2014) Trinity: on using trinary trees for unsupervised web data extraction. IEEE Trans Knowl Data Eng 26(6):1544–1556

66. Small S, Medsker L (2014) Review of information extraction technologies and applications. Neural Comput Appl 25(3–4):533–548
67. Sokolova M, Lapalme G (2009) A systematic analysis of performance measures for classification tasks. Inf Process Manag 45(4):427–437
68. Tandon N, de Melo G, Weikum G (2011) Deriving a web-scale common sense fact database. In: Proceedings of the twenty-fifth AAAI conference on artificial intelligence, AAAI'11. AAAI Press, San Francisco, CA, pp 152–157
69. Tao C, Embley DW (2009) Automatic hidden-web table interpretation, conceptualization, and semantic annotation. Data Knowl Eng 68(7):683–703
70. Team RC (2017) R: a language and environment for statistical computing. In: R foundation for statistical computing. Vienna, Austria
71. Umamageswari B, Kalpana R (2017) Web harvesting: web data extraction techniques for deep web pages. In: Kumar A (ed) Web usage mining techniques and applications across industries, pp 351–378
72. Varlamov MI, Turdakov DY (2016) A survey of methods for the extraction of information from web resources. Program Comput Softw 42(5):279–291
73. Wang RC, Cohen W (2007) Language-independent set expansion of named entities using the web. In: Proceedings of the 2007 Seventh IEEE International Conference on Data Mining, ICDM '07. IEEE Computer Society, Washington, pp 342–350. doi:10.1109/ICDM.2007.104
74. Wang RC, Cohen WW (2009) Character-level analysis of semi-structured documents for set expansion. In: Proceedings of the 2009 conference on Empirical Methods in Natural Language Processing: Volume 3, EMNLP'09, Association for Computational Linguistics, Stroudsburg, pp 1503–1512. http://dl.acm.org/citation.cfm?id=1699648.1699697
75. Weninger T, Fumarola F, Barber R, Han J, Malerba D (2011) Unexpected results in automatic list extraction on the web. SIGKDD Explor Newsl 12(2):26–30. doi:10.1145/1964897.1964904
76. Weninger T, Johnston TJ, Han J (2013) The parallel path framework for entity discovery on the web. TWEB 7(3):161–1629
77. Wolff KE (1994) A first course in formal concept analysis. In: Faulbaum F (ed) StatSoft '93. Gustav Fischer Verlag, Jena, pp 429–438
78. Yates A, Banko M, Broadhead M, Cafarella MJ, Etzioni O, Soderland S (2007) Textrunner: open information extraction on the web. In: HLT-NAACL (Demonstrations), pp 25–26. http://acl.ldc.upenn.edu/N/N07/N07-4013.pdf
79. Yuen SY, Chow CK, Zhang X, Lou Y (2016) Which algorithm should I choose: an evolutionary algorithm portfolio approach. Appl Soft Comput 40:654–673

Marcin Michał Mirończuk he is a graduate of Bialystok Technical University, Faculty of Electrical Engineering in Poland. He received MSc degree, in 2007. He received the Ph.D. degree from Bialystok Technical University, Faculty of Computer Science, in 2013. He is currently working in National Information Processing Institute - Laboratory of Intelligent Information Systems, Warsaw, Poland. Also, he has received the scholarship from Poland Institute of Computer Science of the Polish Academy of Sciences where he worked from 2012 to 2014. His research interests are in the areas of applied mathematics, data/text mining technique and information extraction.