Model-based Integration of Unstructured Web Data Sources using Graph Representation of Document Contents

Radek Burget[a]
Faculty of Information Technology, Brno University of Technology, Bozetechova 2, Brno, Czech Republic

Keywords: Information Integration, Domain Modelling, Document Processing, Structured Record Extraction.

Abstract: Unstructured or semi-structured documents on the web are often used as a media for publishing structured, domain-specific data which is not available from other sources. Integration of such documents as a data source to a standard information system is still a challenging problem because of the very loose structure of the input documents and usually missing semantic annotation of the published data. In this paper, we propose an approach to data integration that exploits the domain model of the target information system. First, we propose a graph-based model of the input document that allows to interpret the contained data in different alternative ways. Further, we propose a method of aligning the document model with the target domain model by evaluating all possible mappings between the two models. Finally, we demonstrate the applicability of the proposed approach on a sample domain of public transportation timetables and we present the preliminary results achieved with real-world documents available on the web.

1 INTRODUCTION

Despite much effort dedicated to the development of different technical means for annotating the semantics of the presented data such as Microformats[1], RDFa[2] and others, the World Wide Web is still an extremely large source of mostly unannotated documents. These documents often contain structured and potentially useful data presented in a way that is convenient for human readers but it is completely unsuitable for automated processing. Therefore, using the documents as a data source for traditional information systems that are based on structured data models presents a challenging task.

A typical domain-oriented information system uses a structured data representation and storage (for example a relational database), which has been designed based on the analysis of the target domain, identification of the individual entities, their properties and the relationships among them. However, on the web, many potential sources of domain-specific data have the form of documents designed primarily for human readers. Although the data contained in these documents follow basically the same structure that comes from the target domain, their integration to an existing information system is difficult because of the very loose way of their presentation without any formal annotation.

In (Burget, 2017), we have mentioned several domains, where this situation is quite typical such as scholarly data (conference proceedings contents), sports results or public transport time tables. In all these (and many other) domains, the data has a fixed and predictable structure that potentially allows its integration to existing applications in the respective domains. However, the corresponding data sources often have the form of periodically published documents (mostly web pages; PDF documents are typical for some domains such as timetables) whose human interpretation is assumed for understanding the presented data.

Traditionally, the integration of such web sources is implemented using different kinds of *wrappers* that recognize data fields in the documents by analyzing the underlying document code – mostly the HTML code represented as a Document Object Model (DOM) (Schulz et al., 2016). For each data source (the source of the input documents), the corresponding code patterns are different and therefore, a specific wrapper must be prepared. Such approach is reliable and feasible when considering a limited number of previously known data sources that provide a larger number of documents but it is not practical at

[a] https://orcid.org/0000-0001-5233-0456
[1] https://microformats.io/
[2] https://rdfa.info/

Burget, R.
Model-based Integration of Unstructured Web Data Sources using Graph Representation of Document Contents.
DOI: 10.5220/0008350103260333
In *Proceedings of the 15th International Conference on Web Information Systems and Technologies (WEBIST 2019)*, pages 326-333
ISBN: 978-989-758-386-5

all, when the input documents come from previously unknown sources, each document has been prepared independently and uses a completely different way of data presentation.

In this paper, we propose a model-based approach aiming to overcome the specific details of the individual documents by an automatic discovery of a mapping between the previously defined domain data model and the presented data records. The main presented contributions are the following:

- We present a technology- and language-independent graph-based model of the document contents that allows to interpret the contained data in different alternative ways.
- We propose a method for evaluating the possible mappings between the created document model and the target domain model that describes the expected structure of the contained data and for choosing the best mapping based on a statistical analysis.
- We demonstrate the application of the described approach on a sample domain of public transport timetables.

We also include preliminary results of this work in progress that show the applicability of the proposed document model and mapping methods on real-world documents.

2 RELATED WORK

The research in the topic of data record extraction from web documents has been running for over 20 years. Apart from historical HTML-based approaches (Schulz et al., 2016), due to the evolution of the web technology (mainly in the HTML and CSS languages and the dynamic web pages) and the increasing complexity of web documents, the recent approaches usually combine the analysis of the document code, with visual presentation properties (Potvin and Villemaire, 2019; Shi et al., 2015). However, most of the current methods use DOM[3] as the primary document representation (Figueiredo et al., 2017; Guo et al., 2019; Lockard et al., 2018; Shi et al., 2015; Yuliana and Chang, 2018). This limits the applicability of the methods to specific HTML documents where the DOM elements accurately delimit the desired data fields.

From the data integration point of view, the current methods infer the schema of the extracted records from the source documents themselves (Figueiredo et al., 2017; Shi et al., 2015; Yuliana and Chang, 2018). In all cases, the approach is to find a specific region or multiple regions (Figueiredo et al., 2017) that contain the records and then, a flat internal structure of the records is determined based on finding the regular patterns in the document code and by comparing the similarity and other characteristics of the repeating sequences. This approach allows easy application of the methods to any document independently on its domain; however, the integration of the extracted data to a domain information system requires further interpretation and transformation of the extracted records.

In contrast to the above mentioned data-driven approaches, there has been significantly less attention given to the research of the model-driven approaches. (Embley et al., 1999) uses a conceptual domain model that is directly mapped to HTML code based on different heuristics. In (Potvin and Villemaire, 2019) a flat list of extracted data field is used and (Lockard et al., 2018) integrates the extracted data with an existing knowledge base.

In our previous research (Burget, 2017), we have proposed a basic approach for matching individual binary relationships in a domain model to visual presentation patterns in the documents. In this paper, we generalize the matching to the whole domain models and above all, we introduce a formal graph-based document model of the input documents that makes the matching possible.

3 THE DATA INTEGRATION TASK

The data integration task we consider in this paper is the following: We have a (potentially unlimited) collection of unstructured input documents on the source side and a structured domain-specific information system on the target side.

The target information system is typically designed based on the analysis of the particular domain, which results in a domain data model such as a entity-relationship diagram (ERD) or its equivalent depending on the used design methodology. The model captures the basic entity sets, their properties (attributes) and the relationships among them. Independently on whether an ERD or another formalism is used, we may define a domain model for our purpose as follows:

Definition 1. *A domain model is a tuple $D = (E, P, R)$, where E is a set of entity sets, P is a set of properties (attributes in ERD) and $R \subset (E \times (E \cup P))$ is the set of relationships.*

[3] https://www.w3.org/DOM/

Figure 1 shows a simple ERD for the public transport timetables domain. Note that we consider just a part of the ERD that is relevant to the considered data sources; the complete ERD for a real-world information system would be obviously significantly larger.

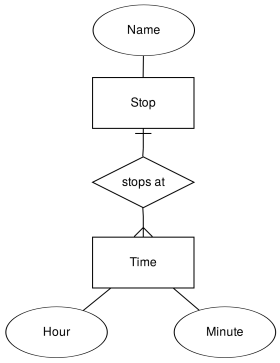

Figure 1: An entity-relationship model for the domain of public transportation timetables with two entity sets (*Time*, *Stop*), three properties (*Hour*, *Minute*, *Name*) and one relationship (*stops at*).

At the input, we assume a collection of *documents* that contain visually presented structured *data records* consisting of several *data fields*. The documents come generally from different sources and therefore, the way of data presentation, formatting or the implementation may be different for every single document. However, we put the following assumptions on the input documents:

- We assume formatted text documents where the document creator may specify the *visual properties* (fonts, colors, etc.) for every part of the document text as well as the *visual organization* of the contents (alignment, spacing, etc.) by any means. For the web sources, the HTML web pages and PDF documents are the most typical but our content model presented below in section 4 is independent on the actual technology.
- Every document contains multiple *data records* consisting of the *data fields* that may be directly mapped to the properties in the target ERD (i.e. without any additional transformations) and the records are *consistent* regarding their structure and visual presentation (their visual properties and organization as mentioned above).

Figure 2 gives the overview of the document processing process. First, the visual properties and positions of all parts of the document text are computed. This is the only task that depends on the document type. For some document types such as HTML, this requires rendering the document by a web browser. In PDF documents, the necessary information is available directly. In the next steps, we identify the *text chunks* that represent the candidate substrings of the document text that potentially could represent a data field. Based on the extracted text chunks, we build a *page contents model*, which is basically a graph that describes the visual properties of the individual chunks and the visually presented relationships among them. We describe the model and its construction below in section 4.

The key part of the information integration process consists of finding the most appropriate mapping between the created document contents graph and the domain model. For this purpose, we also represent the domain model as a graph of the entity properties and the relationships among them and we search for a best mapping between the two graphs. The details of this process are described in section 5.

In the following sections, we will use the already mentioned public transport timetables as a sample domain. Our goal is to integrate the data about the stops and the corresponding times from the timetable documents as shown in Figure 1. We believe, this domain is suitable for illustrating the individual steps for the following reasons:

- It is challenging. The timetables are a good example of source documents that present data in a very ambiguous way and even the human readers need some experience to interpret the data properly in some more complex cases.
- It is practically useful. Although there exist different portals and aggregators in this domain, they are usually limited to certain countries, regions or groups of companies and they typically do not provide their structured data to third parties.
- There are many highly diverse documents from different transportation companies available on the web.

However, the presented integration approach is not limited to a single domain as long as the above mentioned assumptions on the input documents are met.

4 DOCUMENT CONTENTS MODEL

The goal of the proposed document contents model is to capture the possibly relevant parts of the document contents and their mutual relationships based on their visual presentation. We define the model as a graph:

Definition 2. *The document contents model is defined as a graph* $G = (C, E)$*, where* C *is a set of* text chunks

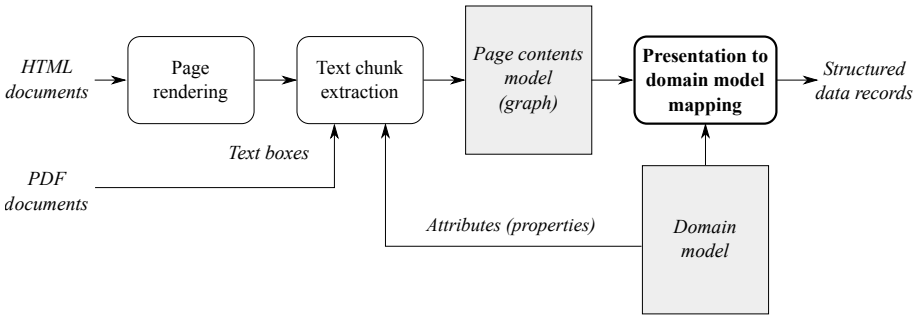

Figure 2: An overview of the data integration process.

Lincoln | County Hospital

Monday to Saturday except Bank Holidays

Lincoln Bus Station	0700	0720	0745	then every 30 mins	**15**	**45**	until
Monks Road 12	0710	0730	0755		**25**	**55**	
Tower Estate	0712	0732	0757		**27**	**57**	
County Hospital	0719	0739	0805		**35**	**05**	
Tower Estate	0722	0742	0808		**38**	**08**	
Monks Road 12	0725	0745	0811		**41**	**11**	
Lincoln Bus Station	0740	0800	0825		**55**	**25**	

Lincoln Bus Station	1545	1620	1650	1720	1750
Monks Road 12	1555	1630	1700	1730	1800
Tower Estate	1557	1632	1702	1732	1802
County Hospital	1605	1640	1710	1740	1810
Tower Estate	1608	1643	1713	1743	1813
Monks Road 12	1611	1646	1716	1746	1816
Lincoln Bus Station	1625	1700	1730	1800	1830

for Sunday journeys see line **17** & **18** timetables

Figure 3: An example time table.

that represent the relevant parts of the contents together with their visual formatting and form the vertices of the graph; $E \subset C \times C$ is a set of graph edges, that represent the relationships *among the chunks as expressed by the document layout.*

With a *text chunk*, we understand any piece of content (a substring of the document text), that possibly represents a value of a domain property. In the moment of the chunk extraction, we do not decide, whether the given substring really represents a part of a data record; the goal is to identify all substrings that "look like" a value of a given property when considered separately.

Definition 3. *A* text chunk *is a tuple $c = (t_c, s_c, p_c)$, where t_c is the text of the chunk (the actual substring of the document text), s_c represents the visual style of the text and p_c represents the position of the chunk as displayed in the resulting page.*

Definition 4. *The* chunk style *is further defined as $s_c = (fs, w, st, c, bc)$ where fs is the average font size, $w \in [0,1]$ is the average font weight from 0 (normal font) to 1 (bold font), $st \in [0,1]$ is the average font style (1 for italic font, 0 for regular font) and c and bc are the computed foreground and background colors of the displayed chunk.*

Definition 5. *The* position *$p_c = (x, y, w, h)$ describes the x and y coordinates of the chunk in the page and its width w and height h.*

The edges E of the graph represent the mutual *relationships* among the chunk pairs. Based on their mutual positions, we identify specific relationships that are interesting for further analysis of the whole data record organization. For example, two chunks may be in a *onRight*, *below*, *sameLine* or another relation as described in section 4.2.

Both the chunks and the relationships are extracted from rendered documents as shown in Figure 2. In the next sections, we provide the details of the chunk and relationship extraction.

4.1 Chunk Extraction

For the chunk extraction, we use a *connected line* as a smallest unit of the rendered document.

Definition 6. *A* connected line *represents a part of the document text that is positioned on a single line (considering the y coordinates of the individual characters) and it does not contain an empty space wider than a certain threshold Δx.*

In our experimental setup, we have used $\Delta x = 2.5f$ where f is the average font size used at the considered line. The goal of this setting is to ensure that the normal text formed by space-separated words forms single lines and the parts separated with a larger space create separate connected lines.

Example 1. *In the example timetable in Figure 3, the header and footer text forms continuous text lines; however, the stop names and the time data are separated by a larger space. Thus, we obtain the connected lines "Lincoln Bus Station", "0700 0720 0745", "15" and "45" for the first line of the schedule, etc. Note that the connected lines are not necessarily consistent regarding their style; they may contain text with different font weights, colors, etc. as we may notice for the station names.*

For each domain model property $p \in P$ (see Definition 1), we extract a set C_p of chunks from all the connected lines. The algorithm for the chunk discovery within the connected lines depends greatly on the type of the property p. For our sample domain, we have used a simple algorithm that finds all one- or two-digit numbers in the appropriate range for hours and minutes; the chunks for stop names are discovered using a simple regular expression allowing a sequence of alphanumeric characters and some commonly used punctuation. In our previous experiments on other domains (Burget, 2017), we have also mentioned the usage of named entity classifiers (Finkel et al., 2005) for recoginizing personal names and locations or even using the DBPedia Spotlight tool (Daiber et al., 2013) for recoginizing entities from the DBPedia dataset. Advanced algorithms for numeric value discovery have been proposed as well (Neumaier et al., 2016).

As the result of the chunk extraction, we obtain a complete set of chunks for all the properties $C = C_{p_1} \cup C_{p_2} \cup \ldots \cup C_{p_n}$ where $n = |P|$. Note that the chunk detection itself may be quite inaccurate as we use very approximate methods for the chunk extraction. The extracted chunks may even overlap; e.g. in the "Monks Road 12" string, we discover three chunks: the whole string forms the *name* chunk, the "12" substring forms the *hour* and *minute* chunks because both interpretations are possible. It is the task of the mapping phase (described in section 5) to complete the data records and exclude the incorrectly discovered chunks.

4.2 Relationship Modelling

After the chunks have been detected, we analyze all the chunk pairs $(c_1, c_2) \in C \times C$ and we investigate whether there is an relationship between c_1 and c_2 given by their mutual positions (x_1, y_1) and (x_2, y_2). We have identified several relationships that are interesting for further analysis. Every relationship is defined by a relation $E_x \subset C \times C$ and we say that there is a spatial relationship x between c_1 and c_2 iff $(c_1, c_2) \in E_x$. Currently, we consider the following relationships:

- *onRight* – $(c_1, c_2) \in E_{onRight}$ when c_1 and c_2 are placed on the same line just next to each other and c_2 is on the right side of c_1.
- *after* – c_2 is on the same line anywhere to the right of c_1.
- *sameLine* – c_1 and c_2 are on the same line regardless their mutual positions.
- *below* – c_2 is placed just below c_1.
- *lineBelow* – c_2 is placed on a line that is just below c_1.

As we may see, a chunk pair may belong to multiple relations as the spatial relationships (e.g. *after* and *sameLine*) are not mutually exclusive.

Finally, the complete set of relationships is then $E = \bigcup E_x$ for all the relations x listed above. Together with the set C of chunks, it creates the document content graph as defined in Definition 2.

5 MAPPING TO THE DOMAIN MODEL

Our information integration approach is based on the assumption that some of the extracted text chunks may be mapped to the individual properties of the domain model as defined in Definition 1 and similarly, some discovered spatial relationships among them may be mapped to the domain model relationships. During the mapping phase, we find all possible *mappings* from the constructed document contents graph to the domain model, we evaluate them and finally, we use the best mapping found.

Below, we describe the representation of the domain model used for the final mapping. Further in section 5.2, we define a mapping formally and finally in section 5.3, we discuss the way of evaluating the individual mappings and finding the most suitable one.

5.1 Domain Model Transformation

As the first step, we transform the domain model to a simplified graph model that describes only the properties and relationships as the entity sets have no direct representation in the documents. An example model for the timetables domain is shown in Figure 4. The properties are divided into *groups* (the dashed boxes) where each group corresponds to a set of properties that are always presented together in a 1:1 relationship.

Definition 7. *The* domain graph model *is a graph* $D_g = (G, R_g)$ *where* $G = \{G_1, G_2, \ldots, G_n\}$ *is a set of property groups,* $G_i \subset P$ *and* $G_i \cap G_j = \varnothing$ *for any* $1 \leq i, j \leq n$. *P is the set of domain properties.* $R_g \subset G \times G$ *is a set of relationships between groups.*

The domain graph is constructed from the domain model defined in Definition 1 as follows:

- All the properties of a single entity set belong to the same group.
- If two entity sets are in a 1:1 relationship, all their properties belong to a single group.

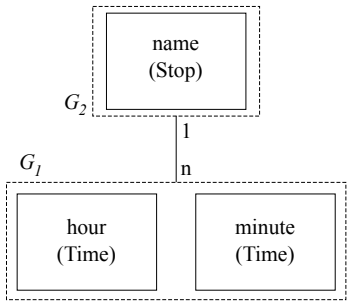

Figure 4: A domain graph model corresponding to the domain model shown in Figure 1 that represents the property groups and the relationships among them.

- The 1:M relationships are transformed to the relationships between the respective groups.

Currently, we don't consider M:N relationships in our method because they are difficult to represent in the documents in an understandable way and therefore, they are very rarely used in source documents.

Considering the example from Figure 1, we obtain two groups of properties: $G_1 = \{hour, minute\}$ and $G_2 = \{name\}$ as shown in Figure 4. Subsequently, we analyze the possible mappings between the document contents graph and the domain graph model.

5.2 Mapping Representation

When considering a particular document represented by the document contents graph, there exist many possible mappings between the chunks and the properties in the domain graph and similarly, between the relationships in the two graphs. In our approach, a mapping presents a hypothesis about the visual presentation of data records, which is subsequently evaluated and compared with other hypotheses.

Based on the above mentioned assumption that there exist multiple visually consistent data records in the source documents, a mapping basically describes two aspects of the records:

1. The visual style of the text chunks used for presenting each property $p \in P$ in the input document.
2. The actual spatial relationships (as mentioned in section 4.2) among the property values.

Let's consider the chunk style defined in Definition 4 and let S be a set of all distinct chunk styles used in the input document. Further, let R_s be the set of all spatial relationships between chunks discovered in the input document. Then, we may define the mapping between a property group G_i in the domain graph and the document contents graph as follows:

Definition 8 (Group Mapping.)**.** *For each group $G_i \in G$, the mapping is defined as $m_i = (f_{si}, f_{ri})$ where $f_{si} : G_i \mapsto S$ is a morphism that assigns a chunk style to each property in G_i and $f_{ri} : G_i \times G_i \mapsto R_s$ assigns spatial relationships to the property pairs.*

The f_{rg} morphism does not necessarily assign a relationship to all possible property pairs. For a unique description of the mapping, it is sufficient that the property pairs form a connected graph. For example, considering three properties a, b and c, the mapping may contain $(a,b) \mapsto onRight$, $(a,c) \mapsto below$ (which can be read as b is on the right side of a and c is below a). We obtain a connected graph of properties and therefore, it is not necessary to find any relationship for the remaining combinations such as (b,c). Considering the group G_1 in Figure 4, it is sufficient to find one of the morphisms $(hour, minute) \mapsto r$ or $(minute, hour) \mapsto r$, where $r \in R_s$.

Similarly, we define an *inter-group mapping* that corresponds to the way how the connection of two groups is visually presented in the document:

Definition 9 (Inter-group Mapping.)**.** *For a pair of groups $(G_i, G_j) \in G \times G$, the inter-group mapping is $m_{ij} = (p_i, p_j, r)$ where $p_i \in G_i$, $p_j \in G_j$ and $r \in R_s$.*

In other words, we define a spatial relationship r between two properties where the first property belongs to the first group and the second property belongs to the second group. Again, we have to find enough mappings between the group pairs so that we obtain a connected graph of groups. Then, the complete mapping is $m = (M_G, M_I)$ where M_G is a set containing a group mapping for each group in G and M_I is a set of the inter-group mappings.

Example 2. When considering our example timetable in Figure 3 and the domain graph in Figure 4, we find many different styles used for the presentation of the *hour* values in the document (when considering the style of all the chunks in C_{hour}) and similarly for the *minute* and *name* properties (the style morphisms f_{s1} and f_{s2}). Moreover, we find different ways how the $(hour, minute)$ pair is possibly presented, e.g. *minute* is on the right side of *hour* or *minute* is below *hour* or *hour* is below *minute*, etc. (the f_{ri} morphism). And finally, we find the possible presentation of the inter-group relation, e.g. *hour* is on the same line as *name*. Since *hour* and *name* are in separate groups in the domain graphs, we know that *hour* actually represents a complete $(hour, minute)$ group and there may exist multiple such pairs related to a single name because of the 1:N relationship between the groups.

By considering all combinations of chunk styles, and the applicable intra-group and inter-group rela-

tionship representations, we obtain a set M of all possible mappings from the contents graph to the domain model graph.

5.3 Evaluation of the Mappings

The last step is the evaluation all the mappings and choosing the most suitable one. For each mapping $m \in M$, we apply the style and spatial relationship mappings on the document text chunks and as a result, we obtain a set of candidate data records, where each data field of the record is represented by a text chunk.

For the evaluation, the following aspects of the discovered candidate records are important:

- The number of chunks actually covered by the records. Although we admit that some of the chunks may have been incorrectly identified, we assume that the correctly identified ones prevail and thus, more chunks contained in the discovered records indicate a better result.
- Visual consistency of the records. The given mapping defines the actual visual style of the chunks mapped to the individual properties as well as the spatial relationships among them (e.g. hours and minutes being at the same line). However, the records may differ in the distance and alignment of the particular chunks. When evaluating consistency, we compare the individual records and we observe the variance of the corresponding distances among the chunks. The lower the overall variance is, the more consistent (and thus better) are the records.

Additionally, we allow using certain number of wildcards in style specifications. It is quite common in visual presentation that some of the records or data fields are distinguished from the others by a different background color, font style, etc. In our experiments, we allow one wildcard in the chunk style, i.e. based on the style defined in Definition 4, one the (fs, w, st, c, bc) attributes may be disregarded.

As we may notice, the two evaluation criteria mentioned above are contradictory to some extent. It is easy to cover a large number of chunks and discover many data records when we allow low visual consistency of records and vice versa. For our experiments we have empirically set the total mapping score to $s = 0.6p + 0.4c$ where p is the percentage of chunks contained in the records and c is the visual consistency, $p, c \in [0..1]$.

6 EXPERIMENTAL EVALUATION

For the evaluation on real-world documents, we have implemented the proposed method in Java. For input document processing, we have used the CSSBox[4] rendering engine for HTML documents and the PDFBox[5] library for reading PDF documents.

Our preliminary tests (being this a work in progress) were run on 30 timetables in PDF available online on the websites of various transportation companies that operate in different en countries (Czechia, Spain, Italy and the United States). As a second use case, we have extracted the publication data (authors, titles and sessions) from CEUR Workshop Proceedings[6] (HTML documents).

The tests have shown the practical usability of the proposed document contents model described in section 4 as well as the domain mapping method. However, in about 10% of input documents, the correct mapping was not evaluated as the best one and the evaluation function had to be adjusted for obtaining correct results. Therefore, we consider the mapping evaluation the main issue for our ongoing research.

7 CONCLUSIONS

In this paper, we have proposed an approach to the integration of the data contained in web documents to structured information systems. Unlike most of the existing approaches that derive the data structure from the input documents, our method is driven by a previously defined domain model of the information system.

In order to make the information integration possible, we have designed a graph-based model of the document contents and subsequently, we have proposed a method for finding the best mapping of the document contents model to the domain model. Our preliminary results show that the approach allows integration of real-world HTML and PDF documents and mapping of the published data to the fixed domain model. The evaluation of the possible mappings seems to be the most challenging topic for our next research.

[4]http://cssbox.sourceforge.net
[5]https://pdfbox.apache.org/
[6]http://ceur-ws.org/

ACKNOWLEDGEMENTS

This work was supported by the Ministry of the Interior of the Czech Republic as a part of the project Integrated platform for analysis of digital data from security incidents VI20172020062.

REFERENCES

Burget, R. (2017). Information extraction from the web by matching visual presentation patterns. In *Knowledge Graphs and Language Technology: ISWC 2016 International Workshops: KEKI and NLP&DBpedia*, Lecture Notes in Computer Science vol. 10579, pages 10–26. Springer International Publishing.

Daiber, J., Jakob, M., Hokamp, C., and Mendes, P. N. (2013). Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)*.

Embley, D. W., Campbell, D. M., Jiang, Y. S., Liddle, S. W., Lonsdale, D. W., Ng, Y.-K., and Smith, R. D. (1999). Conceptual-model-based data extraction from multiple-record web pages. *Data Knowl. Eng.*, 31(3):227–251.

Figueiredo, L. N. L., de Assis, G. T., and Ferreira, A. A. (2017). Derin: A data extraction method based on rendering information and n-gram. *Information Processing & Management*, 53(5):1120 – 1138.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370.

Guo, J., Crescenzi, V., Furche, T., Grasso, G., and Gottlob, G. (2019). Red: Redundancy-driven data extraction from result pages? In *The World Wide Web Conference*, WWW '19, pages 605–615, New York, NY, USA. ACM.

Lockard, C., Dong, X. L., Einolghozati, A., and Shiralkar, P. (2018). Ceres: Distantly supervised relation extraction from the semi-structured web. *Proc. VLDB Endow.*, 11(10):1084–1096.

Neumaier, S., Umbrich, J., Parreira, J. X., and Polleres, A. (2016). Multi-level semantic labelling of numerical values. In *The Semantic Web – ISWC 2016*, pages 428–445, Cham. Springer International Publishing.

Potvin, B. and Villemaire, R. (2019). Robust web data extraction based on unsupervised visual validation. In *Intelligent Information and Database Systems*, pages 77–89, Cham. Springer International Publishing.

Schulz, A., Lässig, J., and 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)*, pages 562–567.

Shi, S., Liu, C., Shen, Y., Yuan, C., and Huang, Y. (2015). Autorm: An effective approach for automatic web data record mining. *Knowledge-Based Systems*, 89:314 – 331.

Yuliana, O. Y. and Chang, C.-H. (2018). A novel alignment algorithm for effective web data extraction from singleton-item pages. *Applied Intelligence*, 48(11):4355–4370.