Tomasz Kaczmarek
Poznan University of Economics, Faculty of Informatics and Electronic Economy, Department of Information Systems

Dominik Zyskowski
Poznan University of Economics, Faculty of Informatics and Electronic Economy, Department of Information Systems

Adam Walczak
Poznan University of Economics, Faculty of Informatics and Electronic Economy, Department of Information Systems

Witold Abramowicz
Poznan University of Economics, Faculty of Informatics and Electronic Economy, Department of Information Systems

INFORMATION EXTRACTION FROM WEB PAGES FOR THE NEEDS OF EXPERT FINDING

Abstract: This paper describes a mechanism for the extraction of relevant information about people from Polish portals for professionals. The method of information extraction is based on hierarchical execution of XPath commands and regular expressions depending on the structure of processed documents. The extraction component EXT is a part of the eXtraSpec system, which task is to support Human Resources departments of Polish companies during recruitment and team building. EXT is able to deal with several sources of information and with user profiles that are acquired from professionals' portals. In this article we also discuss the advantages of the chosen extraction method in the context of the goals of the whole eXtraSpec system and we show the directions of future research.

Key words: web information extraction, eXtraSpec, hierarchical algorithm, XPath, Polish language

1. Introduction

Several information sources may be useful when searching for candidates in the process of staff recruitment. These sources can vary, from peers' opinions expressed during informal conversation to formalized CV documents prepared by candidates. In-between there is a whole universe of data that one can benefit from in the recruitment process. It is worth noting that information relevant to the expert finding may have traditional form, but

ISBN 978–83–7431–273–8 ISSN 0860-150X

also an electronic one. Moreover, not only textual sources can be useful, but also information that appears as the result of social networking between professionals from the particular business domain.

Currently, more and more information is available in an electronic form. The requirement for most recruitment processes is to provide a resume in a form of a computer file, which is later processed by internal workflow systems. One should also consider the development of social activities on the Web that result in the creation of professional profiles on dedicated portals like LinkedIn. It is easily imaginable that the amount of relevant information is enormous. This information may be utilized in the recruitment process, through combination of data from various sources, its cross-validation and complementing. Thanks to using standardized vocabulary it is possible to match a profile built in such a way against concrete requirements defined for a precisely described job offer.

The structure of this paper is as follows. First, we present the general picture of the eXtraSpec system along with a motivating example that justifies a practical usability of such a system. Then, we show the state of the art solutions in this field and comment on the related work already done. In the fourth section we outline our method of web information extraction. We describe the hierarchical extraction algorithm, then present the structure of profile extracted and the structure of extraction rules. Finally, we conclude our work and show the directions of future research.

2. Overview of eXtraSpec project

The goal of the eXtraSpec[1] project is to create tools that can help automate the recruitment process and expert finding. Our special interest is the analysis and extraction of information from Web sources, especially available in the Polish language.

The architecture of the eXtraSpec system consists of several components that play their roles in subsequent stages of the supported process [11]. As an input for the system serve documents retrieved from Polish portals for

[1] The work published in this article was supported by the project titled: "Advanced data extraction methods for the needs of expert search" (http://extraspec.kie.ue.poznan.pl) financed under the Operational Program Innovative Economy and partially supported by European Union's European Regional Development Fund programme (contract no. UDA-POIG.01.03.01-30-150/08-01).

professionals, like Profeo[2] or GoldenLine[3] (see Figure 1 and Figure 2). An important fact is that the system is dedicated to process documents written in the Polish language, which is a hard task due to the complexity of the Polish grammar. To our best knowledge, such research in the domain of the Polish language is quite a novelty. These documents are being processed by the extraction component (EXT), which role is to validate, preprocess them and later to create extracted profiles that store all relevant information from the point of view of the recruiter.

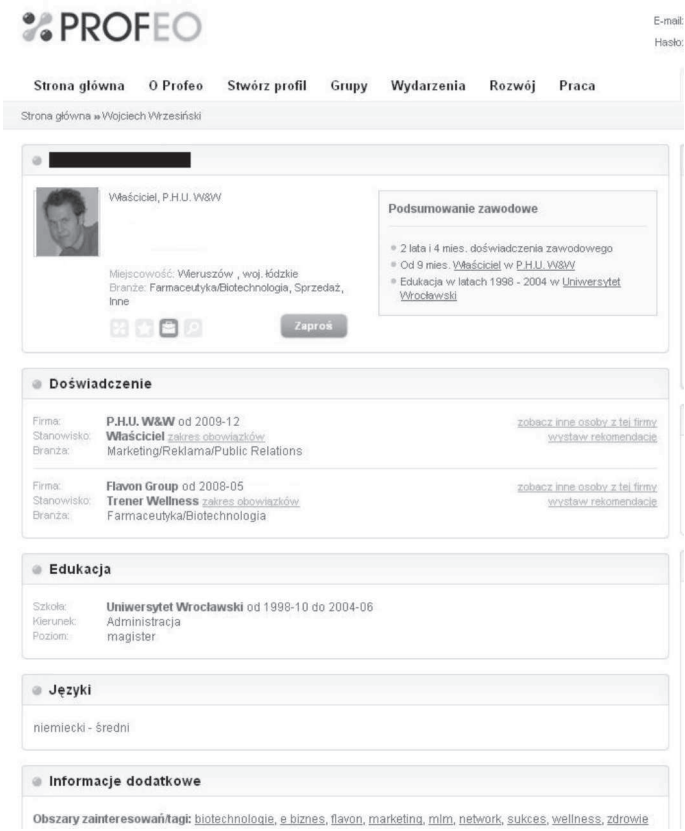

Figure 1. Profeo.pl – exemplary profile

The next step is to normalize profiles using internal domain ontologies in order to perform disambiguation and allow for reasoning over experts' profiles. Subsequent component – Fusion (FUS) – is responsible for detecting

[2] http://profeo.pl/
[3] http://www.goldenline.pl/

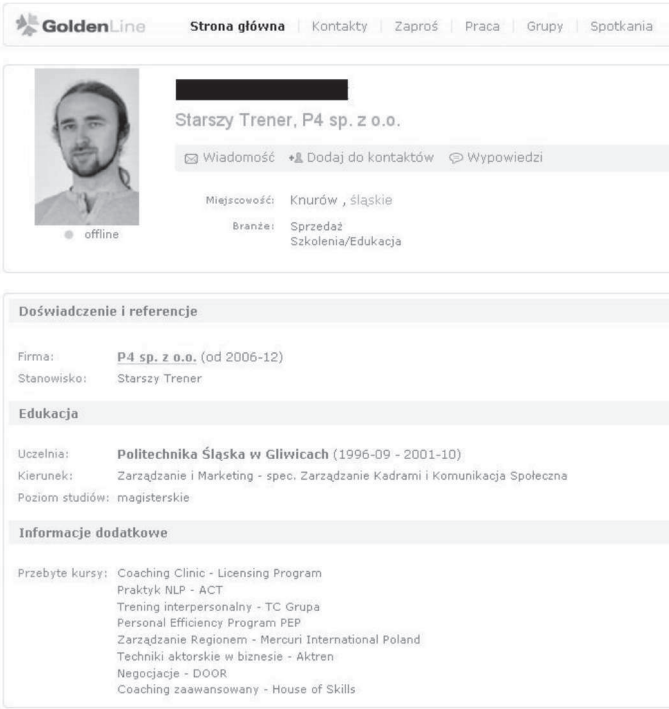

Figure 2. GoldenLine.pl – exemplary profile

whether profiles coming from diverse sources describe the same person. If so, an aggregated profile is created to keep an up-to-date information along with a revision history. These profiles are stored internally in the system. End users communicate with the system through a graphic user interface that allows for composing queries and visualizing the resulting profiles. In this paper we concentrate on the first step of the information flow in the eXtraSpec system, namely on the extraction component – EXT.

2.1. Motivating example

The example we describe here relates to the problem of recruitment of new employees to be involved in a new project launched by a Polish company XYZ S.A. Suppose HR executives of XYZ didn't find appropriate employees already working for the company who can take new tasks related to another project. XYZ has to start looking for new people by posting job advertisements or hiring "head-hunting" companies. These two methods are costly, so HR department workers start to browse social portals for professionals to find prospective employees. However, this task is very time consuming, as each profile has to be checked manually for validity and against a profile of

the ideal candidate. Moreover, people who have accounts on several portals often do not update all of them, so HR department can not be sure that the information is up-to-date.

All these issues reveal the need for a tool that is able:

- to process automatically user profiles from several popular Polish portals for professionals,
- to extract from web pages the most relevant information about candidates,
- to retrieve profiles of candidates that possess concrete characteristics,
- to track changes in candidates' profiles manifested on various portals,
- to cope with different vocabulary used to describe the same concepts.

Having a system that performs these tasks automatically, the process of recruitment can by shortened. Even if the proper candidate is not found, the system should suggest to compose a team of people that together would be able to accomplish all required tasks in the new project of XYZ.

The place of extraction component within such system is to retrieve profiles from websites, extract from them only relevant information and transform this information to the form processable by other components of the system, responsible for query answering and teams composition.

2.2. Information lifecycle in eXtraSpec

The complete architecture of the system is presented in [11]. The following components are responsible for the creation and processing of profiles in eXtraSpec:

1. Extraction (EXT) – responsible solely for retrieval of documents from dedicated sources, extraction of relevant content and creation of extracted profiles (see section 4).
2. Normalization (NOR) – its task is to disambiguate the names of organizations, jobs, and capabilities. In order to fulfill this task heuristic methods and dictionaries are used. Additionally, NOR takes care of time references. In effect, normalized profiles are created.
3. Fusion (FUS) – aggregates information from multiple profiles. One of the most important tasks of FUS is to determine whether several profiles belong to the same person or not.
4. Reasoning (REA) – processes user query for finding an expert having specific skills or composes a team that together can possess requested skills.
5. Graphical user interface (GUI) – a tool for users to formulate queries to the system and to visualize the results.

The modular architecture of the system allows for clear distinction of

the stages of documents processing and verification of efficiency of each component. The flow of documents within eXtraSpec architecture and dependencies between components are presented on the Figure 3.

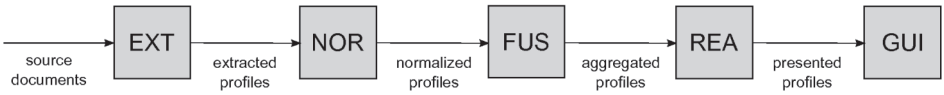

Figure 3. Dependencies between components of eXtraSpec system (based on [11])

3. Related work

The task of automatic information extraction from Web pages is rather well recognized in literature [1–4], but to our best knowledge, there were no attempts to develop a method of Web information extraction from Polish portals for professionals.

The ultimate goal of information extraction in the context of WWW is the translation of semi-structured information being in the form of HTML document, to a structured information (most frequently relational) [13, 14]. In contrast to text extraction, information extraction from Web pages uses less formal grammars, but mostly relies on the structure of processed documents (HTML, XML, PDF) and links between them. When used together with a Web source navigation techniques, extraction can compose the data retrieved from different sources and create a coherent data sets [15].

Manual extraction is a very time-consuming task, especially for a big amount of web pages. It is possible to develop a wrapper that extracts and navigates in the source document. The systems that were already presented can be categorized according to their capabilities and degree of realization of extraction tasks [16–18]. Most authors consider such distinctive criteria like: dependence on the source structure, ability to extract from multiple pages, need to manual creation of wrappers, used formalism (grammars, rules, statistics, logics)

The only approach to processing CVs in Polish was signalized only once [5]. A few exceptions are the standardized CV documents like Europass CV[4] or HR-XML Consortium[5] CV template that may be easily processed for further team building [6].

[4] http://europass.cedefop.europa.eu

[5] http://www.hr-xml.org

Therefore, in further phases of extraction the technique of Named Entity Recognition with statistical classification of entities is taken advantage of. Apart from this, eXtraSpec system uses semantics in the form of domain ontologies to prepare meaningful profiles. In literature, several semantically enhanced information extraction systems like Vulcain [7], SOBA [8], OBIE [9] or KIM [10] were presented, but again no method was dedicated for the Polish language.

4. Extraction method

In the described extraction component of the eXtraSpec system we are concentrated on the processing of semi-structured documents. The main task of this component is to extract concrete attributes of the profile from source documents. To do this, we create a tree of extraction rules, in which each rule is responsible for extracting elements of the profile. EXT component performs a transformation of HTML tree to another tree structure – in this case a profile tree, which is represented as XML document (see Figure 4). After one run of EXT over retrieved HTML documents there is one corresponding profile for each document. Such transformation could be performed with XSLT, but his technique is not optimal for the needs of eXtraSpec due to cumbersome maintenance and complexity. In order to allow for using the same extraction method for different Web sources, we proposed a hierarchical extraction algorithm that operates on extraction rules represented as XPath expressions and regular expression for advanced string transformations. Extraction component processes mainly HTML documents that come from business-related Polish social portals like GoldenLine or Profeo. Because of their internal form, they can be processed using analysis of tree structure created by HTML tags.

In the case of already mentioned business portals, the HTML structure enforces only the data schema. This means that apart from the structured data, there is information stored also in an unstructured way, like the description filled in by hand in natural language.

4.1. Structure of extracted profile

Like it was presented on Figure 4, the extracted profile has a tree-like structure. A single profile is created for every document that is processed in the system Leafs in the tree contain text passages which were extracted from the document. For the purpose of representation of a profile we use XML, for which adequate schema was prepared.

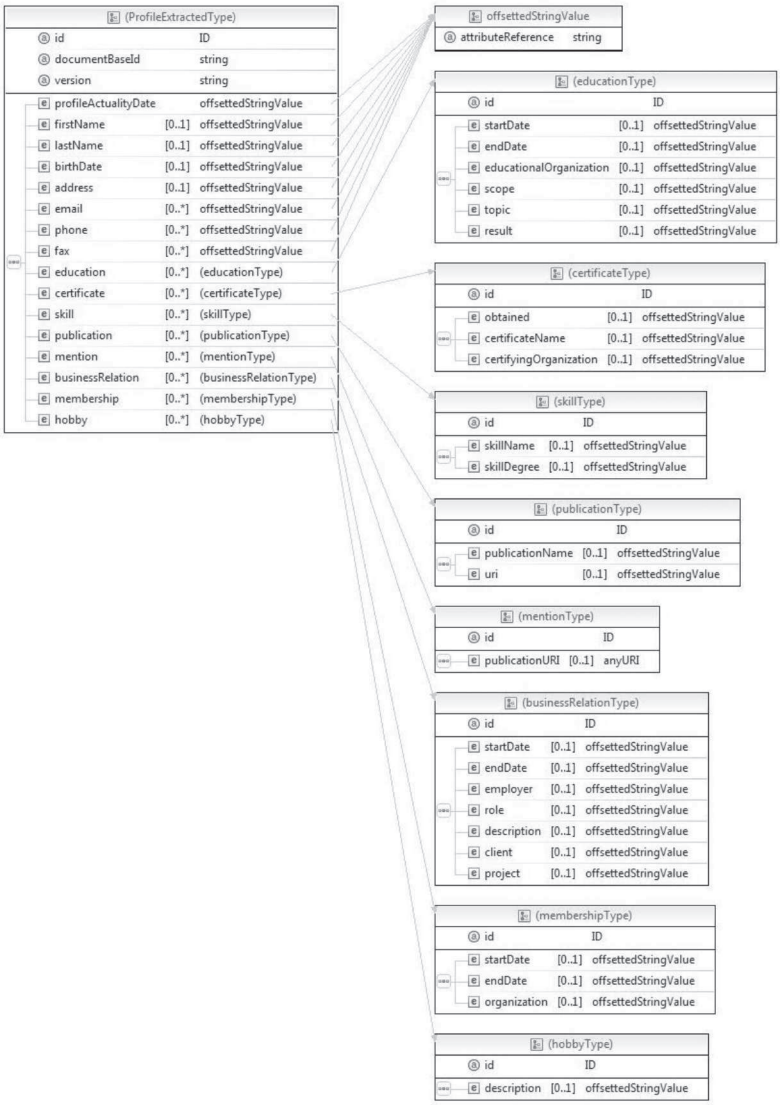

Figure 4. The schema of a profile produced by EXT component

We took into account all the interesting features of the expert at a fine level of granularity, which results in a document similar in structure to a complex CV. The following classes of information were taken into account:

- basic personal data (first and last name, birth date, address, email, phone and fax numbers)
- education – for each entry concerning education we record start and end date of a given education phase, organization that was responsible for

this stage, topic and scope of education and its level (primary school, secondary school, college, university etc.)
- certificates – which are characterized by date, its name and the name of certifying organization
- skills with skill name and declared level of proficiency
- publications – described with the name (or reference to the publication) and link to the publication on the Web
- mentions – this category contains a list of references to locations in different documents, where the person was mentioned
- work experience – each entry contains start and end dates for a work period, name of the employer, occupied post and the description, and, for projects – project name and name of the customer
- membership – start and end dates of the membership, the name of the organization
- hobby with the description field

Except the above mentioned categories, certain metadata about the profile are stored: its identification number, the date that it was created and reference to the original document.

4.2. Hierarchical extraction algorithm

The extraction algorithm works by processing HTML web pages discovered in the selected sources and producing for each document an extracted profile. As the sources of data for the EXT component selected social portals were chosen, where people publish their CVs or similar information, using predefined forms with varying (but usually limited) degree of flexibility.

The extraction approach that was chosen is based on two assumptions. First, that the target and source data is a tree, and second, that the source data has a fixed structure that follows tree-based approach. A fixed structure means that a) all categories of nodes in a tree are known and fixed, b) their position in relation to each other does not change (although a node may be missing) and c) recursion is not allowed (nodes can not be nested within each other infinitely). The first assumption is true for the target data (profile extracted) because it was defined to follow it, but it might not be true for certain web pages, which do not follow tree-like organization of data. That would include for example web pages which are built in tabular orientation of data. An important distinction has to be drawn between using tables to position the data – which does not preclude using tree structures, and using tabular data orientation. For example, a web page where people would be listed in columns of a table, with their personal data placed in rows is not a tree-like structure and consequently can not be handled in the approach

chosen for extraction of information in our case. However, by observing a number of portals we found that a non-tree structuring does not occur in practice, therefore this limitation is not significant.

The approach to information extraction that we have chosen is based on hierarchy of extraction rules. The hierarchy of rules resembles hierarchy of data in the target profile to be extracted – each rule extracts part of the HTML document that relates to certain element in the extracted profile. That part can be further processed by child rules to extract more detailed pieces of information.

In each rule hierarchy there is a single root rule, that would extract the part of HTML document tree which includes the whole information required to build a profile.

Each extraction rule may point to a target element of the extracted profile. There can be rules which do not point to a profile element – they are used as grouping "facilities" for child rules.

Each extraction rule indicates, whether the element that it points to is atomic (an expected result is a simple leaf element – for example first name), compound (target element may be a complex element like education entry), or is a collection of elements of the same type (expected extraction result is a set of education entries).

Each rule fulfills a condition, that its extraction result has to encompass all the information required to fill the target of the rule (even though it can be done by child rules). This limits the child rules to extract only part of the subtree extracted by the parent rule.

We transform each obtained HTML document into a valid XML document by cleaning it with standard HTML Tidy[6] utility and the extraction rules are applied in a descending order by processing the extraction hierarchy.

Such approach can be formalized as transformation of a source tree into a target tree, and consequently described for example in XSLT. However, in order to avoid the complexity of XSLT (or other formalisms similar in power) we chose to limit ourselves to a subset of transformations that are possible, which is motivated by the assumptions described above about the structure of source and target data, and our desire to simplify the creation of the extraction rule hierarchies.

Since the extraction rules hierarchy is dependent on the target data structure, which is fixed, the hierarchies do not differ in general structure.

[6] http://tidy.sourceforge.net

The most important differences concern the rules, which in turn are mainly dependent on the source structure. As a consequence, each document source has to have it's own hierarchy of extraction rules prepared. At a current stage we don't use any mechanisms to automatically discover the source structure and build the rules automatically.

4.3. Structure of extraction rules

Hierarchical extraction performed by EXT takes advantage of the extraction rules to extract specific data from source documents. For the sake of easier management of extraction rules, we decided to keep the rules for different sources separately. This is also due to the uniqueness of each source, because each portal sticks to its own presentational structure of the business profile. Therefore, it is impossible to apply the same extraction rule in a variety of sources, since there are differences in the DOM models of various business portals.

Extraction rules, similarly to the extraction method, are hierarchical rules. This means that each rule possesses a given parent rule which must be executed before. Otherwise, the execution of child rule will not be successful and the requested fragment of source document will not be extracted. In the current version of EXT component, extraction rules are being formed twofold:

- as XPath expressions
- as regular expressions (Regex)

Because of the fact that source documents are Web pages, the majority of information is being extracted by XPath expressions. As a result, content blocks or just text strings are being obtained. If such result can not be directly stored in some field of profile extracted, due to data type mismatch, there is an additional need to process it. Sometimes, more detailed XPath rules are used, and sometimes we use Regex commands to process text strings. Regex allows for a sophisticated transformations of strings, which is useful is many cases. As a simple example may give the phrase "John Smith" which we can obtain in a result of XPath expression. Then we can use regular expression to split this string into two: "John" and "Smith", and later put them directly in the PE.

An exemplary extraction rule is presented in the Listing 1. This rule has id=5, and its result is being stored in the PE field named "address". As one can see, this rule is formulated in XPath. It has a parent extraction rule (id=1), that must be executed before. Arity value means that the result of this rule is a single, simple object, that can be directly put to the profile extracted.

```
<extractionRule id="5" relatedElement="address"
parentExtractionRuleId="1" language="XPath" arity="single">
/html/body/div[12]/div/div/table/tbody/tr/td[2]/div/dl[1]/dd
</extractionRule>
```

Listing 1. Example extraction rule with its target element (address in the extracted profile), indication of parent rule and arity

Although we have developed separate sets of extraction rules for various data sources, all rule sets use the same schema. This allows the user to easily analyze the syntax of rules for the documents coming from different sources. The XML schema of extraction rule is presented below on the Listing 2.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified"
targetNamespace="http://extraspec.org/schema/ExtractionRulesSchema"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="extractionRules">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" maxOccurs="unbounded" name="extractionRule">
<xs:complexType>
<xs:simpleContent>
<xs:extension base="xs:string">
<xs:attribute name="language" type="xs:string" use="required" />
<xs:attribute name="parentExtractionRuleId" type="xs:string" use="required" />
<xs:attribute name="id" type="xs:unsignedByte" use="required" />
<xs:attribute name="relatedElement" type="xs:string" use="required" />
<xs:attribute name="arity" use="required">
<xs:simpleType>
     <xs:restriction base="xs:string">
     <xs:enumeration value="single">
     </xs:enumeration>
     <xs:enumeration value="compound">
     </xs:enumeration>
     <xs:enumeration value="collective">
     </xs:enumeration>
     </xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="sourceName" type="xs:string" use="optional" />
<xs:attribute name="representationMimeType" type="xs:string" use="optional" />
</xs:complexType>
</xs:element>
</xs:schema>
```

Listing 2. XML schema of the extraction rule

Firstly, we define the namespace used by extraction rules. Then, the following attributes are being defined:

- language – defines the language of the extraction rule. Currently two options are possible: XPath and Regex. This attribute is required for well-formedness of the rule.
- parentExtractionRuleId – the id of parent rule that must be executed before. This attribute is required for well-formedness of the rule.
- id – unique id number of the extraction rule. This attribute is required for well formedness of the rule.
- relatedElement – denotes the name of the field of PE, in which the result of this rule is stored. This attribute is required for well-formedness of the rule.
- arity – denotes the type of content block that is extracted by this rule. This attribute is required. We have three types of arity:
  - single – for extracted elements that in PE are of type Offsetted-StringValue. This is the simplest data type used in PE.
  - collective – used when one rule extracts several elements of the same type. An example may serve the extraction of block with employers, in which the element "professional experience" is described by the same attributes.
  - compound – complex type in which one extraction rule processes a content block consisting several elements of various type or a collection of elements.
- source name – the name of data source processed by a given set of extraction rules. This attribute is optional.
- representationMimeType – describes document type, on which the rules are executed. This attribute is optional.

Extraction rule sets are stored in XML format, so no special toolkit is required to manage them. Rule sets are kept as XML files and stored in a dedicated folder, accessible only by EXT component.

5. Discussion

There are two questions to discuss regarding the described approach to information extraction: efficiency of the approach and the limitations of the method.

To consider efficiency of the hierarchical extraction rules, one has to take into account, that the rules in our case (contrary to other information extraction scenarios) are not statistical in nature. After an analysis of popularity of Polish business portals, we decided to serve the two most popular in the first version of extraction component. For GoldenLine we created a set

of 32 extraction rules, whereas Profeo is being processed by a separate set of 34 extraction rules. We base on the assumption that the sources of documents have a fixed structure. It stems from the fact, that web portals do not allow for upload of arbitrary document as a user profile, but allow only to fill in the form, which has fixed, hierarchical structure. Therefore, if this structure is guessed correctly, and the extraction rules are crafted to match it, there is no possibility of a wrong result of the extraction, if only the form was filled correctly. The only limit of the method in that respect is the granularity of information it can extract – if certain fields in the form allow for arbitrary text (for example descriptions), it is hard to create extraction rules to handle it because this type of content does not adhere to the fixed tree-like structure assumption. For that we envision applying other techniques (originating in Natural Language Processing) which are more suitable for dealing with free text content.

The limitation of the method stems from assuming tree-like structure of the source. Although it is supported by our observation that other structures do not occur in our application domain, it is worth to discuss other motivations for such simplification. One of our goals was to simplify the creation of the whole extraction rules bundle. We achieved it by organizing it into a hierarchical, non-recursive structure which is the simplest and yet still powerful enough to extract into a required target structure. Second, such a simplification enables automated discovery of source structure. Although complex, it is feasible for a limited number of sources. Algorithms for discovery of web page structures were already researched [12] and they can be tuned to the tree-like character of the sources in our application domain.

6. Conclusions and future work

In this paper we have presented an extraction component called EXT, being a part of the eXtraSpec project. EXT is capable of processing web pages written in the Polish language in order to extract the information relevant for the needs of expert finding and team building. To perform its tasks, EXT uses an original algorithm of content extraction, which is done in accordance to the hierarchy of a given HTML document. This means that the structure of the web page is reflected in a resulting profile extracted and the content of specific fields is transformed by XPath or Regex and settled in a corresponding field.

The directions of future work also include development of text proces-

sing techniques to cope with fields that are manually filled by humans, because automatic extraction from such fields using XPath or regular expressions is cumbersome and sometimes does not give satisfactory results. So far, the developed extraction rules cope very well with structured information presented on these portals, but currently we do not process natural language full text phrases that are also present in the business profiles on these portals. Natural language processing of Polish phrases is one of the possible future research tracks in the project. This technique would be especially interesting in case of processing the content blocks that can be freely filled by users. In such cases, it is not possible to use predefined extraction rules, as some heuristic methods are needed.

We also plan to prepare an extensive evaluation of extraction rules by comparing the results obtained automatically and manually.

BIBLIOGRAPHY

[1] Rahardjo, B. and Yap, R. H. 2001. *Automatic information extraction from web pages*. in Proceedings of the 24th Annual international ACM SIGIR Conference on Research and Development in information Retrieval (New Orleans, Louisiana, United States). SIGIR '01. ACM, New York, NY, 430–431.

[2] Chang, C., Hsu, C., and Lui, S. 2003. *Automatic information extraction from semi-structured Web pages by pattern discovery*. Decis. Support Syst. 35, 1 (Apr. 2003), 129–147.

[3] Vadrevu, S. 2008. *Automated Information Extraction from Web Pages Using Presentation and Domain Regularities*. Doctoral Thesis. UMI Order Number: AAI3304898., Arizona State University.

[4] Novotny, R., Vojtas, P., and Maruscak, D. 2009. *Information Extraction from Web Pages*. in Proceedings of the 2009 IEEE/WIC/ACM international Joint Conference on Web intelligence and intelligent Agent Technology – Volume 03 (September 15–18, 2009). Web Intelligence & Intelligent Agent. IEEE Computer Society, Washington, DC, 121–124.

[5] Kaczmarek, T., Kowalkiewicz, M., and Piskorski, J. (2005). *Information Extraction from CV*. in W. Abramowicz (Ed.), 8th International Conference on Business Information Systems (pp. 185–189). Poznań: Wydawnictwo Akademii Ekonomicznej w Poznaniu.

[6] De Meo, P., Plutino, D., Quattrone, G. and Ursino, D. (2010). *A team building and team update system in a projectised organisation scenario*. in Int. J. Data Mining, Modelling and Management, Vol. 2, No. 1, pp. 22–74.

[7] Todirascu, A., Romary, L., and Bekhouche, D. 2002. *Vulcain – An Ontology-Based Information Extraction System*. in Proceedings of the 6th international Conference on Applications of Natural Language To information Systems-Revised Papers (June 27–28, 2002). B. Andersson, M. Bergholtz, and P. Johan-

nesson, Eds. Lecture Notes In Computer Science, vol. 2553. Springer-Verlag, London, 64–75.

[8] Buitelaar, P., Cimiano, P., Frank, A., Hartung, M., and Racioppa, S. 2008. *Ontology-based information extraction and integration from heterogeneous data sources*. in Int. J. Hum.-Comput. Stud. 66, 11 (Nov. 2008), 759–788.

[9] K. Bontcheva. 2004. *Open-source tools for creation, maintenance, and storage of lexical resources for language generation from ontologies*. in 4th Conf. on Language Resources & Evaluation, Lisbon, Portugal.

[10] Popov, B., Kiryakov, A., Ognyanoff, D., Manov, D., and Kirilov, A. 2004. *KIM – a semantic platform for information extraction and retrieval*. Nat. Lang. Eng. 10, 3–4 (Sep. 2004), 375–392.

[11] Abramowicz W., Kaczmarek T., Stolarski P., Węcel K., Wieloch K., *Architektura Systemu Wyszukiwania Ekspertów eXtraSpec*, TWZP 2010, Prace Naukowe Akademii Ekonomicznej we Wrocławiu, Katowice 2010.

[12] Kowalkiewicz, M., Orlowska, M., Kaczmarek, T. i Abramowicz, W. (2006). *Towards more personalized Web: Extraction and integration of dynamic content from the Web*. in Proceedings of the 8th Asia Pacific Web Conference APWeb 2006. Harbin, China: Springer Verlag.

[13] Line Eikvil. *Information extraction from world wide web – a survey*. Technical report. 1999.

[14] Andrew McCallum, William W. Cohen. *Information extraction from the world wide web*. Tutorial, 2002.

[15] Ion Muslea. *Extraction patterns for information extraction tasks: A survey*. The AAAI-99 Workshop on Machine Learning for Information Extraction, 1999.

[16] Boris Chidlovskii, Uwe M. Borghoff, Pierre-Yves Chevalier. *Towards sophisticated wrapping of webbased information repositories*. 5th International RIAO Conference, p. 123–135, 1997.

[17] Jane Yung jen Hsu, Wen tau Yih. *Template-based information mining from html documents*. 14th National Conference on Artificial Intelligence, strony 256–262, 1997.

[18] Juliana S. Teixeira, Berthier A. Ribeiro-Neto, Alberto H. F Laender, Altigran S. da Silva. *A brief survey of web data extraction tools*. SIGMOD Record, 31(2): 84–93, 2002.

Tomasz Kaczmarek
Poznan University of Economics
Faculty of Informatics and Electronic Economy
Department of Information Systems
t.kaczmarek@kie.ue.poznan.pl

Information extraction from web pages for the needs of expert finding

Dominik Zyskowski
Poznan University of Economics
Faculty of Informatics and Electronic Economy
Department of Information Systems
d.zyskowski@kie.ue.poznan.pl

Adam Walczak
Poznan University of Economics
Faculty of Informatics and Electronic Economy
Department of Information Systems
a.walczak@kie.ue.poznan.pl

Witold Abramowicz
Poznan University of Economics
Faculty of Informatics and Electronic Economy
Department of Information Systems
W.Abramowicz@kie.ue.poznan.pl