Rainbow – Multiway Semantic Analysis of Websites

Vojtěch Svátek, Jirka Kosek, Martin Labský,
Jiří Bráza, Martin Kavalec,
Miroslav Vacura, Vladimír Vávra
Department of Information and Knowledge Engineering, University of Economics, Prague, Czech Republic
svatek@vse.cz

Václav Snášel
Department of Computer Science, Technical University, Ostrava, Czech Republic
vaclav.snasel@vsb.cz

Abstract

The Rainbow project aims at the development of a reusable, modular architecture for web (particularly, website) analysis. Individual knowledge–based modules separately analyse different types of web data and communicate the results via web–service interface. The output of analysis has the form of classes (of web resources) predefined in an ontology, extracted text, and/or addresses of retrieved web resources. Within the project, several original methods of analysis as well as (analytic) knowledge acquisition have been developed. The current domains of investigation are sites of small organisations offering products or services, and pornography sites. The paper is the first systematic overview of diverse methods developed or envisaged in Rainbow.

1. Introduction

While the 'pan–WWW' retrieval of documents is dominated by *computation–centred* methods relying on optimised keyword indexes, web analysis at the level of *website* seems to offer itself to *inference–centred*, knowledge–intensive methods, which would respect the peculiarities of different domains and data structures. Website analysis can often be performed offline (indexing scenario), or exploit the time slot available while the user reads a page (browsing scenario): the overhead of using comparably slow knowledge–based methods thus becomes acceptable. While knowledge–based methods are declared as the heart of the future *Semantic Web* (relying on explicit knowledge annotations), their use for analysis of the current web so far received limited attention. Yet, gradual semantic 'upgrade' of the current web is probably a more appropriate way of obtaining the Semantic Web than building it from scratch.

We present examples of problems that can be tackled by knowledge–based website analysis (section 2), principles of our *Rainbow* system developed for this purpose (section 3), and concrete examples of its web analysis services (section 4). Finally, we review related work (section 5) and set up future directions (section 6).

2. Selected problems in website analysis

Two application problems motivated the development of the first version of *Rainbow*: pornography recognition and extraction of key facts from small business sites.

2.1 Pornography recognition

Sites containing nudity are pervasively offered to WWW users, filtering out their content thus becomes an urgent task. The nature of pornography is reflected in different types of data the sites consist of: topological structure, keywords in text and URL, and composition of images that represent the 'ultimate target'. Vacura [16] showed that synergistic semantic analysis of different types of data (combined via weighted average) within such sites yields better results than methods used in isolation.

2.2 Extraction of key facts from business sites

A wide category of sites is that of *organisations* (mostly companies) *offering products or services*; we nickname it as 'OOPS', for brevity. The sites of large companies are, in their majority, dynamically linked to databases, and sometimes (though rarely) even offer standardised APIs to the information stored. In contrast, most websites of small companies are created and maintained as collections of plain HTML documents. Information useful for the potential customer (such as company profiles, price–lists or contact information) is buried inside the static HTML code of particular pages. This code is perfectly accessible but stunningly complex and messy for conventional, knowledge–free methods.

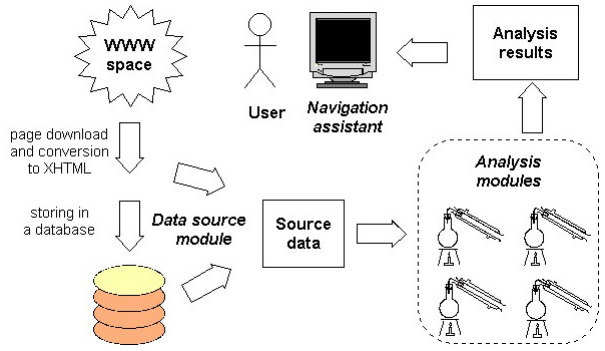

Figure 1. Scheme of current implementation

3. Principles of Rainbow

The development of the *Rainbow* architecture started in spring 2001. Although the individual (mostly student) sub–projects are relatively independent from each other, a few principles are collectively adhered to: separation of the *ways of analysis*, support by formal *ontologies*, and distinction of *generic* services types. The distributed modules of *Rainbow* are developed in different environments (Java, Python, CLIPS) and equipped with hand–written or inductively learned knowledge bases.

3.1. Multiple ways of analysis

In *Rainbow*, we concentrate on six basic data types:

- free–text sentences
- HTML mark–up (and fragments of text inside it)
- URL addresses
- metadata in META and RDF tags
- link topology
- images.

The currently implemented version of *Rainbow* only provides simple forms of analysis of the first four data types. Link topology analysis is soon to be added, while image analysis has only been implemented within a dedicated (pornography–recognition) setting. The services are described by means of a *web–service* (WSDL [4]) interface, and communicate via synchronous requests and answers wrapped in SOAP [3] messages.

The analysis modules access the web by means of a *data–source* module, which downloads the pages, canonises HTML to well–formed XHTML, and stores the data in a relational (MS SQL) database. Another auxiliary module is a *navigation assistant*, mainly serving for testing purposes. It can be installed as an additional pane into the Netscape/Mozilla client and displays the results of multiway analysis (with the currently available services run in parallel) for the page currently viewed by the user. The scheme of the current implementation is at Figure 1.

3.2. Ontology support

As in any distributed knowledge–based application, the role of shared *ontology* is to unify the semantics. The ontology developed for *Rainbow*, being relatively light–weighted in terms of language (it was developed in DAML+OIL [1] but RDF/S [11] constructs prevail), is structurally rather complex. It consists of four layers: two *domain–neutral* and two *domain–dependent* ones.

1. The *upper web ontology* (UWO) defines the most abstract concepts and relations, such as *Document*, *Document Fragment*, *Hyperlink*, *part-of* or *adjacent-to*.
2. For each type of data, such as free text or link topology, there is a single *partial generic model* (PGM). Examples of PGM concepts are *Downward Link* (in the topology PGM) or *Image Gallery* (in the HTML PGM). They are descendants of the UWO concepts *Hyperlink* and *Document Fragment*.
3. For each type of data and each problem domain, there is a *partial domain model* (PDM). Examples of PDM concepts are *Link from Company Intro Page to Menu Page* (in the topology PDM for the OOPS domain) or *Porno Fingerprint Gallery* (in the HTML PDM for the pornography domain). They are descendants of the mentioned PGM concepts.
4. Finally, the collections of PDMs are merged into *domain web ontologies* (DWO), such as for pornography or for OOPS. Identities, subsumptions or just correlations among concepts from different PDMs are established.

The current versions of DWOs have been constructed by human modelling effort. We however started to experiment with an empirical approach based on a *common dataset*. From the descriptions of the same *web resource*, e.g. a (physical) *Document*, by different analysis methods, a vector of attribute values can be generated. The attributes are derived from the UWO relations, such as (document) 'has-class HUB' or 'contains FORM'. The table of vectors (one per resource–object in the dataset) serves as input to the adapted *FCAMerge* [12] method, which generates a *concept lattice*. In the lattice, concepts and relations relevant for the merged ontology can be discovered with limited human effort.

In the present state of the project, the ontologies merely serve as a (semantically unambiguous) basis for *documentation* of the services. They are exploited by human developers and reflected in WSDL descriptions. The logical next step will be automated *verification* of the consistency of services, in connection with formal models of *generic service types* (see next section).

3.3. Generic types of services

As a conceptual framework for structuring the variety of potential *Rainbow* services, three generic types of services have been identified. They are denoted, in turn, as *classification*, *extraction* and *retrieval*, and appear in different forms within different ways of analysis:

- *Classification* takes a *resource* (identified by URL and/or XPointer address) as its input, and returns its *class* (defined in an ontology) as output.
- *Extraction* takes as input a *mereological context* (resource from which the target will be extracted) and *constraints* (class of the target resource), and returns the *content* of the resource as output.
- Similarly, *retrieval* takes as input a *topo–mereological context* (resource within which and/or in the neighbourhood of which the target will be sought) and *constraints* (class of the target resource), and returns *resource/s* as output.

The generic services are formally defined in a special *inference ontology*, which will serve, in connection with ontologies described in section 3.2, for checking the consistency of services committed to the particular type. A trivial check can match the input and output of the same task. For example, the task of retrieving the *Hub Page* in a topology inherits from the generic *retrieval* task the following feature: object/s on *output* must be instance/s of the 'toClass' concept of property *identified-by* for the concept corresponding to the type of resource on *input*. As *Hub Page* is a sub–concept of the UWO concept *Document*, the output should be of type *URL* as only allowed identifier of *Document*s. More complex checks might span over several services and prevent e.g. deadlock or invocation of tasks deemed to fail.

4. Services in Rainbow

In this section, we report on concrete instances of services, grouped according to generic type. In the end of the section, the space of (existing or potential) types of services is mapped, taking into account the generic service type, data type and resource type. We confine ourselves to a brief overview; most methods are described in papers available at `http://rainbow.vse.cz`.

4.1. Resource classification

Classification (also 'categorisation') is ubiquitous in web access tasks. It amounts to assignment of semantic class (from a classification scheme) to a given resource.

In *Rainbow*, we repeatedly experimented with *page* classification based on *URL*. Surprisingly, a knowledge base with 50 *domain–neutral* empirical rules applicable on this simple form of data revealed the nature of the page in 30–50%, depending on the topic area [13]. A *domain–specific* URL analysis tool for pornography recognition achieved accuracy over 95% [16]. Finally, a URL knowledge base exists for the OOPS domain but has not been thoroughly tested yet.

The mentioned experiments were oriented on *context–free* classification of pages, which may take place e.g. when post–processing search results. In website analysis, however, we more often classify the *hyperlinks* starting at the current page, as common in 'navigation assistance'. Hyperlink classification can be translated to classification of pages referenced by them; the known *context* however constrains the classification. We examined the hyperlinks (URLs and anchors) at the main pages of companies when seeking the page with 'general company profile'. Here, a knowledge base with four heuristic rules sufficed for recognition of 90% of desired links [14].

More sophisticated classification methods have been employed in the *pornography recognition* sub–project [16], focusing on narrow (mostly binary) tasks. Examples of such tasks were recognition of 'fingerprint gallery' in HTML code and topology, recognition of porno page by RSAC 'nudity–rating' in META tags, or recognition of 'pornographic image' according to the amount of body colour, position of object, and boundaries of object.

We have not so far used traditional *text categorisation* techniques based on the 'bag–of–word' representation. If useful, these could be implemented on the top of the co–operating *AmphorA* full–text tool (cf. section 4.3).

4.2. Information extraction

Information Extraction (IE) is a stream of research aiming at conversion of text into structured database records. State–of–the–art *web* IE techniques (see [10] for an overview) rely on *specific* patterns of text, HTML and punctuation, in which the target information is wrapped. For the OOPS domain of *Rainbow*, in accordance with our multiway paradigm, we pursue a slightly different approach. The cue to the detection of target information are relatively generic *lexical indicators*. Subsequently, depending on the target, either *free–text–centred* or *HTML–centred* extractors are applied.

Extraction from *free text* (via shallow parsing of key sentences) seems to be useful for certain type of company information such as 'general profile', which is. Indicative terms, in this case, are e.g. 'offer', 'specialize in' or 'manufacture'. In the first try, we arrived at a collection of 20 such domain–neutral terms via reuse of *headings* in the *Open Directory* catalogue. The terms from headings (themselves domain–dependent) were found in the text of pages referenced by the given directory page; this process amounts to fully–automated *labelling* of training data for subsequent *inductive learning* of indicators. The precision of most indicators in determining a 'general–profile' sentence reached over 60%, for some even 90% [7].

Extraction from *HTML–formatted text*, in contrast, is suitable for structured types of information such as contact info or catalogues. For *contact address* and *e–mail*, *lexical indicators* (such as 'Contact:' or 'E-mail:') were tested in connection with generic *outlook–oriented* HTML structures expressing the assignment of *value* (actual address etc.) to a *property* ('has-address' etc.). Within a sample of 60 pages, such method revealed applicable on 40% of *contact addresses*; the rest would require more complex, esp. statistical, extraction techniques. Further, the method was applicable on 70% of *e-mails*, of which one fourth would escape a wrapper relying on `<a href="mailto:XXX">` structures [15].

Third target for IE is the content of *metadata* (META, RDF). The current 'metadata' module of *Rainbow* merely crops the values of 'semantic' attributes of META tags, such as 'keywords', 'description' or 'author', and discards those related e.g. to HTML authoring.

The benefits of our extraction methods (from free text and HTML) are *transparency* and *domain–independence*, which should enable instant reuse of the same tools for a new domain. Instead of tedious learning of complex extraction patterns, *indicators* would merely be substituted and a few parameters tuned. Clearly, we pay for it by lower *precision*. With respect to that, we aim at *interactive* support of 'semantic web' knowledge annotation rather than at fully *autonomous* IE.

4.3. Resource retrieval

Traditional document–oriented *Information Retrieval* (IR) is dominated by *computation–centred* methods. On the other hand, retrieval in the general sense is as ubiquitous as classification, and for some its forms *knowledge–based* methods may suit. In *Rainbow*, we treat separately *index–based* retrieval of resources in large quantities of data, and *direct retrieval*, applicable on complex but not excessively large structures.

Web documents can be *indexed* either as 'bags of words' or taking HTML mark–up into account. For the latter, understood as *XML data indexing*, an original method has been developed [8]. Data and tags are stored as points in a multi–dimensional data space, where each dimension corresponds to a level in the XML tree. Rapid access to the data is ensured by special index structure: the *UB–tree* [2]. The method has been implemented in a full–text IR tool named *AmphorA*; integration of this tool with the *Rainbow* architecture is under investigation. Index–based retrieval will help the knowledge–based tools of *Rainbow* focus on relevant sites and their portions. For the given type of information (say, company profile or address) to be harvested, *AmphorA* will return XPath addresses of occurrences of relevant *lexical indicators*, so that linguistic or HTML analysis tools will only explore their surroundings.

Table 1. Space of website analysis services

Data type: Resource	HTML code	URL	Free text	Topo-logy	Meta-data	Image
Document Collection		CLA		CLA **RET**		
Document	CLA	**CLA**	CLA	**CLA** RET	**CLA**	
Hyperlink		**CLA**		CLA RET		
HTML Fragment	**CLA** RET	CLA				
Image						**CLA**
Phrase			**CLA**			
	EXT		**EXT**		**EXT**	

Direct retrieval is an alternative in situations when the indexing overhead does not pay off due to low frequency of the task, when the nature of data disables their easy indexing, or when lexical indicators are not present. For example, in order to classify a document as *Hub Page*, outgoing links have to be *retrieved* (and counted) in the topology; in order to extract contact information not accompanied by lexical indicators, the 'contacts' page can be found by following a directly *retrieved* link from the main page rather than by keyword search.

In general, retrieval in website analysis is typically not directly targeted at the end user. The resources retrieved are passed further to either classification or extraction.

4.4. General overview

Table 1 lists the types of (existing and potential) services in terms of *source data type*, and *type of resource* to be *classified* (CLA) or *retrieved* (RET). *Extraction* (EXT) is only bound to data type: it is not inherently associated with a resource. Service types implemented (in some form) within *Rainbow* are in boldface.

The fill–up of the table is mostly intuitive. We can e.g. see that *URL* only serves for classification of resources: single documents, collections (e.g. clusters of language versions), hyperlinks, or HTML fragments (containing the 'href' attribute). *Topology* analysis can only be applied on elements/parts of the 'webgraph'. Some fill–ups however depend on the viewpoint and architectural constraints. For example, since the *image analysis* module is assumed to be called by the HTML analysis module, the fact that it classifies inline images does not entail that it classifies the whole page (e.g. as *Pornographic Document*) even if the page contains nothing but the image.

5. Related work

It is hard to align a multi–focal project such as *Rainbow* with existing research, as a whole. For brevity,

we mention only a few related projects; other could be found in papers devoted to particular aspects of *Rainbow*.

Extraction of *company profiles* has been addressed by Krötzch [9]. Their approach is similar to ours in the attention paid to multiple modes of information presentation (HTML structures, phrasal patterns), they however concentrated on a specific domain, casting technology. Compared to our domain–neutral approach, their technique is more precise and comprehensive but not directly reusable for other areas.

Ester [5] apply probabilistic techniques (Naive Bayes and Markov chains) to classify *whole* company websites based on classes of individual pages. There is however no extraction of information below the page level.

Multi–agent systems for website analysis, e.g. [6], typically decompose the tasks according to *semantic class* of target information. Our 'syntactical' separation has the virtue of keeping the collection of modules stable. When a new domain is addressed, only the knowledge bases (instead of whole agents) are changed, while low–level (data–type–dependent) analysis routines can be reused.

6. Conclusions and future work

In the *Rainbow* project, we investigate systematic website analysis as a relatively novel task. The multiway approach should give better control over the process than methods operating on a unified (and thus complex) representation. Advantage can be taken of natural complementarity and supplementarity of information deduced from different types of data. Some of the partial techniques have been tested on real data, and a simple running prototype of the system has been developed.

The integrated functionality of multiple modules is, however, only rudimentary. Attention should be paid to the design of operational *control structures*, respecting the knowledge–level distinctions outlined in section 3.3, and governing the behaviour of *Rainbow* with respect to application tasks. There is also a need of integration schemes for *uncertain* results from different modules. The prospects of *shared ontology construction* and exploitation of *full–text retrieval* have been discussed in sections 3.1 and 4.3. Finally, an *RDF repository* for analysis results (which can be understood as 'semantic web metadata') is under design. The results could be used as additional, *conceptual information* for web search engines, as *facts* over which semantic agents could reason, or as starting point for generation of *natural–language summaries* of websites.

Acknowledgements

The authors would like to thank their *Rainbow* team-mates: Petr Berka, Vilém Sklenák, Petr Strossa and Filip Volavka. The project is partially supported by grant no. *201/03/1318* of the Grant Agency of the Czech Republic.

References

[1] DAML+OIL, online at `http://www.daml.org/`

[2] Bayer, R., "The Universal B–Tree for multidimensional indexing: General Concepts", in: Proc. World-Wide Computing and its Applications (WWCA), Tsukuba, Japan, 1997.

[3] Box, D. et al., *Simple Object Access Protocol (SOAP) 1.1*, W3C Note, online at `http://www.w3.org`.

[4] Christensen, E. et al., *Web Services Description Language (WSDL) 1.1*, W3C Note, online at `http://www.w3.org`.

[5] Ester, M., Kriegel, H.–P., Schubert, M., "Web Site Mining: a new way to spot Competitors, Customers and Suppliers in the World Wide Web", in: Proceedings KDD 2002.

[6] Freitas, F.L.G., Bittencourt, G.: "Cognitive Multi–agent Systems for Integrated Information Retrieval and Extraction over the Web", in: Proc. SBIA-IBERAMIA 2000, LNCS 1952, Springer Verlag 2000.

[7] Kavalec, M., Svátek, V., "Information Extraction and Ontology Learning Guided by Web Directory", in: ECAI W'shop on NLP and ML for ontology engineering. Lyon 2002.

[8] Krátký, M., Pokorný, J., Snášel, V., "Indexing XML Data with UB-trees", in: ADBIS 2002, Research Communications, Bratislava 2002.

[9] Krötzch, S., Rösner, D. "Ontology based Extraction of Company Profiles", in: Workshop DBFusion, Karlsruhe 2002.

[10] Kushmerick, N., Thomas, B.: "Adaptive information extraction: Core technologies for information agents", in: Intelligent Information Agents R&D in Europe: An AgentLink perspective. LNCS 2586, Springer 2003.

[11] Lassila, O., Swick, R.: *Resource Description Framework (RDF) Model and Syntax Specification*. Recommendation, World-Wide Web Consortium, Feb. 1999.

[12] Stumme, G., Maedche, A., "FCA-Merge: A Bottom--Up Approach for Merging Ontologies", in: Proceedings IJCAI 2001, Morgan Kaufmann 2001.

[13] Svátek, V., Berka, P.: "URL as starting point for WWW document categorisation", in: RIAO'2000 – Content-Based Multimedia Information Access, CID, Paris, 2000, 1693–1702.

[14] Svátek, V., Berka, P., Kavalec, M., Kosek, J., Vávra, V.: "Discovering Company Descriptions on the Web by Multiway Analysis", in: Intelligent Information Processing and Web Mining., Springer Verlag, to appear.

[15] Svátek V., Bráza J., Sklenák V., "Towards Triple–Based Information Extraction from Visually–Structured HTML Pages", in: Poster Track of WWW2003, Budapest 2003.

[16] Vacura, M., *Recognition of pornographic WWW documents on the Internet* (in Czech), PhD Thesis, University of Economics, Prague, 2003.