Information Retrieval and Extraction from the Web: the CROSSMARC approach

Vangelis Karkaletsis, Constantine D. Spyropoulos

Software and Knowledge Engineering Laboratory, Institute of Informatics and Telecommunications, NCSR “Demokritos”, GR-15310, Athens, Greece, \{vangelis, costass\}@iit.demokritos.gr

Abstract

The paper presents the CROSSMARC approach for the complex task of identification of interesting web sites and web pages and the extraction of information from them. This task is hard because most of the information on the Web today is in the form of HTML documents, which are designed for presentation purposes and not for automatic extraction systems. This task becomes even harder in a multilingual context, where web pages in different languages need to be considered. CROSSMARC approach focuses on the easy customization of web information retrieval and extraction technology to new domains and languages. This is achieved by adopting and implementing an open, multi-lingual and multi-agent architecture that integrates the CROSSMARC components into a web-based prototype system, as well as by providing an infrastructure that facilitates customization of its components to new domains and languages.

1. Introduction

The retrieval and extraction of information from Web sites is a complex task. A number of systems have been developed to identify domain specific web sites (focused crawlers) and interesting web pages within these sites (site specific spiders) as well as to extract structured data from web pages. We present in this paper the CROSSMARC approach, which covers all the way from the identification of Web sites of interest (i.e. that contain Web pages relevant to a specific domain) in various languages, to the location of the domain-specific Web pages, to the extraction of specific information from the Web pages and its presentation to the end-user. CROSSMARC emphasizes on the easy customization to new domains and languages providing a customization infrastructure composed of various tools and methodologies.

The motivation for web focused crawling comes from the poor performance of general-purpose search engines, which depend on the results of generic Web crawlers. The aim is to adapt the behaviour of the search engine to the requirements of a user, much like Web browsing assistants do (Lieberman 2001). The term ‘focused crawling’ was introduced by Chakrabarti et al. (1999) where the system presented starts with a set of representative pages and a topic hierarchy and tries to find more instances of interesting topics in the hierarchy by following the links in the seed pages. Aggarwal et al. (2001) present a significant improvement of the focused crawling approach, which they call intelligent crawling. In contrast to the focused crawling method, it uses a combination of evidence, in order to rank the candidate hyperlinks by their level of interest and learns the relevant weight of these factors as it crawls. Making the assumption that the initial set of starting points can lead to all interesting pages, very central sites should be used as starting points for the crawl (e.g. Yahoo, Amazon, etc.). Another interesting approach to focused crawling is adopted by the InfoSpiders system (Menczer and Belew, 2000), a multi-agent focused crawler. The process is initialized by a set of keywords and a set of root pages. Each agent starts with a root page and performs focused crawling by evaluating the link value and following the most promising links. The crawler implemented according to CROSSMARC approach involves three different crawlers which exploit topic hierarchies, keywords from domain ontologies and lexica, and a set of representative pages.

1 CROSSMARC is an R&D project partially funded by the EC (IST-2000-25366). Project partners include: NCSR «Demokritos» (GR) (Coordinator), University of Edinburgh (UK), Universita di Roma Tor Vergata (IT), Velti (GR), Lingway (FR), Informatique CDC (FR) (see http://www.iit.demokritos.gr/skel/crossmarc/).
While in focused crawling, the aim is to adapt the behavior of the search engine to the requirements of a user, in site-specific spidering the spider navigates in a Web site, following best-scored-first links. Each Web page visited is evaluated, in order to decide whether it is really relevant to the topic, and its hyperlinks are scored in order to decide whether they are likely to lead to useful pages. Therefore, site-specific spidering involves two decision functions: one which classifies Web pages as being interesting (e.g. laptop offers) or not and one that scores hyperlinks, according to their potential usefulness, i.e., the likelihood that they lead to offer pages quickly. Thus, the input to the 1st decision function is a Web page visited by the spider and its output is a binary decision. This is a typical text classification task, which has been studied extensively in the information retrieval literature. Various machine learning methods have been used for constructing such text classifiers. Sebastiani et al. (2002) provides an up-to-date survey of such approaches. In CROSSMARC we examined a large number of classification approaches in order to find the most appropriate one for each domain and language.

Concerning the second decision function in site-specific spidering, this is a regression function, i.e., the input to the function is the hyperlink, together with its anchor and possibly surrounding text, and the output is a score, corresponding to the probability of reaching a product page quickly through this link. Like classification, there is a variety of machine learning methods that are available for learning regression functions. However, in contrast to text classification, the task of hyperlink scoring has not been studied extensively in the literature. Most of the work on scoring and ordering of links refers to Web-wide crawlers, rather than site-specific spiders, and is based on the popularity of the pages pointed by the links that are being examined, i.e., the number of other pages that have already been visited by the crawler and point to the same page (e.g. Cho et al., 1998). This approach is inappropriate for the spider implemented in CROSSMARC. The only really relevant work that has been identified in the literature is by Rennie and McCallum (1999), who use a type of simplified reinforcement learning in order to score the hyperlinks met by a Web spider. A reinforcement learning link scoring methodology was also examined in CROSSMARC and was compared against a rule-based methodology.

Apart from the process of identification and retrieval of Web pages that are relevant to a particular domain or task, the information management over the web requires also techniques for extracting information from the retrieved web pages. (Kushmerick 1997) first introduced the technique of wrapper induction for Information Extraction (IE) from HTML pages. The technique works extremely well for highly structured document collections so long as the structure is similar across all documents, however it is less successful for more heterogeneous collections where structure based clues do not hold for all documents (Soderland 1997). CROSSMARC approach combines language-based IE and wrapper induction based IE in order to develop site-independent IE techniques that can operate well on different types of web pages (from rigidly structured documents to free texts).

Concerning the presentation of the extracted information to the end-user, CROSSMARC approach exploits user modeling techniques. User profiles and user stereotypes are maintained in order to deliver the information to the end-user according to his/her preferences as well as taking into account the preferences of other users belonging in the same stereotype with him/her. CROSSMARC approach takes also into account the requirements of system availability all the time, bridging operation failures of either external entities or system components. That’s why a multi-agent architecture was adopted and implemented.

Finally, CROSSMARC approach involves a customization infrastructure that provides methodologies and tools for:

- the creation and maintenance of domain specific resources (e.g. ontologies, lexica, important entities and fact types for the domain),
- the collection of corpus necessary for the training and testing of the crawling, spidering and information extraction components,
- the customization of each component exploiting the domain specific resources and the collected corpora.
Sections 2 and 3 of the paper present the functionalities of the integrated CROSSMARC prototype and the customization infrastructure respectively.

2. CROSSMARC Prototype

CROSSMARC approach involves a number of processing steps that concern:

- the collection of interesting and domain-specific web pages (focused crawling to identify web sites of interest, spidering of the identified web sites to locate domain-specific web pages),
- the extraction of information about product/offer descriptions from the collected web pages,
- the storage and presentation of the extracted information to the end-user according to his/her preferences,

and which have been implemented by the open, multi-lingual and multi-agent architecture of the integrated prototype that is depicted in Fig. 1. CROSSMARC prototype involves agents for scheduling and supporting its components, which communicate through a blackboard.

The agent-based integrated prototype operates constantly according to the agent strategies and the initial settings of the system administrator. Facilities are provided for the system administrator for the management of agents, and the customization of the prototype components. Uniform web access is provided, not only to the integrated system but also to its components, for two types of users, the advanced user and the end-user. The prototype components and the process of configuring them to new domains or languages are briefly presented in the following sub-sections.

![Figure 1: CROSSMARC’s agent based architecture](image)

2.1 Crawling

CROSSMARC crawler implementation involves three different crawler versions (Stamatakis et al. 2003). The first one exploits the topic-based website hierarchies used by various search engines to return web sites under given points in these hierarchies. The second one takes a given set of queries, exploiting

---

2 Visit CROSSMARC site to access CROSSMARC system and its components, either as advanced user or as an end-user.
CROSSMARC domain ontology and lexicons, submits them to a search engine, and then returns those sites that correspond to the pages returned. The third one takes a set of ‘seed’ pages and then conducts a ‘similar pages’ search (available from advanced search engines such as Google) to find pages deemed similar to the seed pages. It then returns the sites corresponding to these pages. The list of web sites output from the crawler (output of the best combination of the 3 versions) is filtered using a light version of the site-specific spidering tool (NEAC). The light version of NEAC navigates the site until it finds an interesting web page. If it finds one, it considers the site as fit and stops navigating. Otherwise, the whole site is navigated (this is accelerated through a link scoring mechanism) and if no fit page is found, the site is characterized as unfit.

For improving existing domains or adding new ones, the system administrator will just have to change the settings within the crawler configuration files for the three separate versions. The administrator will then have to perform an evaluation in order to find the optimal start points for each version as well as their optimal combination. Customization into a new domain requires also the customization of the site-specific spidering tool, a light version of which is used to filter the results of the crawler.

In our crawling experiments (for the two domains and the four languages of the project), we tried to achieve acceptable performance for all the languages by putting increased effort into the initial stage of forming hypotheses about what would be good directory and query start points. As the evaluations show in Table 1, we do not seem to have been entirely successful. Our hope was that each domain and language would require just one cycle of hypothesis and evaluation but the poor results for some languages seem to indicate that it might sometimes be necessary to perform further cycles.

<table>
<thead>
<tr>
<th>Language</th>
<th>1st Domain (Laptop offers) F-measure (%)</th>
<th>2nd Domain (Job offers) F-measure (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>60,6</td>
<td>89,7</td>
</tr>
<tr>
<td>Italian</td>
<td>40,0</td>
<td>57,3</td>
</tr>
<tr>
<td>Greek</td>
<td>40,5</td>
<td>67,4</td>
</tr>
<tr>
<td>French</td>
<td>70,5</td>
<td>44,7</td>
</tr>
</tbody>
</table>

Table 1: Crawler Evaluation

2.2 Spidering
CROSSMARC spidering component consists of three modules:

- Site navigation. It traverses a Web site, collecting information from each page visited, and forwarding part of the collected information to the “Page Filtering” module and another part to the “Link Scoring” module.
- Page filtering. It is responsible for deciding whether a page is an interesting one (e.g. contains laptops offers) and therefore should be stored or not.
- Link scoring. It validates the links to be followed, in order to accelerate site navigation (only links with score above a certain threshold are followed).

Before forwarding a page in the ‘Page Filtering’ module, this must be processed by a “Language Identification” module which decides on the language of the page returning a language_id. The “Language Identification” module is based on some simple rules that check for the occurrence of frequent words (Souter et. al. 1994). According to its language_id, the page is forwarded to the corresponding filter in the “Page Filtering” module which decides on whether the page contains interesting information or not. Before saving an interesting page in order to be processed by the information extraction agent, the spider passes its content through a tool that transforms the HTML code into XHTML. The spider used in the final version of the prototype integrates a machine learning based version of the page filtering module and a rule-based version of the link scoring module.

For improving existing domains or adding new ones, the administrator will have to:
• Train the machine learning based page filtering module to the new domain or re-train it to an existing one. Training requires the use of: (a) the domain ontology and lexicons, (b) the corpus formation tool for the creation of a representative training corpus of positive and negative web pages (Table 2a), (c) the use of the WebPageClassifier tool to construct the domain-specific classifier.

• Train the rule-based link scorer to the new domain or re-train it to an existing one. Training requires: (a) the specification of new settings in the configuration file of the link scoring module, and (b) experimenting with each specification until the optimal setting is found.

A large number of classification approaches were examined for implementing the page filtering module (Naïve Bayes classifier, Nearest-neighbour classifier, Decision tree induction, Support Vector Machines, AdaBoost boosting method, LogitBoost boosting method). This was done in order to find the most appropriate method for each domain and language. The method selected finally was Support Vector Machines. Based on the evaluation experiments for page filtering in the spidering component, in both domains (see the evaluation datasets in Table 2a and the evaluation results in Table 2b), the main conclusion is that we are able to identify with a fairly high degree of confidence, when a Web page retrieved by the spidering component is interesting according to the domain that is being examined.

### Table 2a: The evaluation datasets for the two domains

<table>
<thead>
<tr>
<th>Language</th>
<th>1st Domain Positive Pages</th>
<th>1st Domain Negative Pages</th>
<th>2nd Domain Positive Pages</th>
<th>2nd Domain Negative Pages</th>
<th>Total Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>447</td>
<td>213</td>
<td>1340</td>
<td>771</td>
<td>1384</td>
</tr>
<tr>
<td>Italian</td>
<td>361</td>
<td>283</td>
<td>749</td>
<td>635</td>
<td>1050</td>
</tr>
<tr>
<td>Greek</td>
<td>281</td>
<td>216</td>
<td>669</td>
<td>835</td>
<td>904</td>
</tr>
<tr>
<td>French</td>
<td>330</td>
<td>191</td>
<td>903</td>
<td>723</td>
<td>1756</td>
</tr>
</tbody>
</table>

### Table 2b: Page filtering evaluation results

<table>
<thead>
<tr>
<th>Language</th>
<th>1st Domain F-measure (%)</th>
<th>2nd Domain F-measure (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>96.9</td>
<td>83.2</td>
</tr>
<tr>
<td>Italian</td>
<td>93.7</td>
<td>73.7</td>
</tr>
<tr>
<td>Greek</td>
<td>92.7</td>
<td>87.9</td>
</tr>
<tr>
<td>French</td>
<td>96.9</td>
<td>82.3</td>
</tr>
</tbody>
</table>

Concerning link scoring, we examined a reinforcement learning methodology as well as a rule-based one. The main conclusion of the study in the 1st domain of CROSSMARC was that link scoring, either using rules or machine learning, did not seem to add much value to the page collection process. However, the results in the 2nd domain were more encouraging. If the system works under limited computational resources, it seems to be better off scoring the links with either of the two methods proposed in the project than using blind search. In this manner, it will be able to retrieve a large proportion of the interesting pages per site.

### 2.3 Information Extraction

Information Extraction (IE) from the domain-specific web pages collected by the crawling and spidering agents, relies on a pipeline of three components: a named entity recognition (NERC) component, a demarcator and a fact extraction (FE) component. NERC identifies domain-specific named entities in pages from different sites (Grover et al. 2002), whereas the demarcator classifies the identified entities in the products/offers appearing inside the page. FE identifies domain-specific facts, i.e. assigns domain-specific roles to entities. In order to store the extracted facts in the products’ database, these are first normalized according to the ontology. This is necessary in order to be able to present them to the
end-user according to the user’s preferred locale. After normalisation is performed, the extracted information is converted into a common XML representation, which is used by the Data Storage agent in order to feed the products database.

The multi-lingual IE system of CROSSMARC integrates four mono-lingual IE sub-systems which operate as autonomous processors (one of the monolingual IE systems is presented in (Petasis et al. 2003)). For interfacing with the IE sub-systems a proxy mechanism was developed. More specifically, we developed a module named Information Extraction Remote Invocation (IERI) which takes the XHTML pages produced by the spider and routes them to the corresponding monolingual IE sub-system according to their language (see Fig. 2). The customization of a mono-lingual IE sub-system to a new domain requires the construction of a representative training and testing corpus for the new domain and for all the languages supported. For this purpose, we specified a corpus collection and a corpus annotation methodology that was used in the two domains and the four languages of the project.

![Figure 2: Multilingual IE in CROSSMARC](image)

NERC customization involves the following steps: (a) creation of the NERC DTD, which specifies the important entities for the domain, using the NERC editor of the ontology management system; (b) collection and annotation of the training and testing corpus per language using the corpus collection and NERC annotation methodology; (c) adapting certain resources to the domain (fill the gazetteers with instances from the corresponding lexicon) – this depends on the specific NERC module design; (d) writing of the rules in a rule-based approach, or training of the machine learning technique over the annotated training corpus – this depends again on the specific NERC module design; (e) experimentation and evaluation in order to improve performance.

Demarcator customization involves the following: (a) study of the domain using the collected corpora for the NERC stage; (b) annotation of the product information in the same corpora with the NERC stage; (c) writing of the heuristics for the specific domain; (d) finding the optimal set of heuristics for the new domain, through a series of experiments.

Finally, concerning the customization of FE, this involves: (a) creation of the FE XML, which specifies the important fact types for the domain, their relations to the NERC entities and their possible values according to the ontology schema, using the Template editor of the ontology management system; (b) fact annotation in the same corpora with the NERC and demarcator stages; (c) writing of the rules in a rule-based approach, or training of the machine learning technique over the annotated training corpus –
Four different IE approaches were implemented in the context of CROSSMARC, sharing the same basic idea, that of exploiting the results of named entity recognition in order to produce IE systems that could be applied to pages with different layouts, coming from different web sites, thus handling the main problem of the existing wrapper induction techniques (Sigletos et al. 2003b). Apart from the evaluation of the NERC, demarcator and FE techniques, we also experimented with the complete IE system (NERC, demarcator, FE) for each of the four languages. The evaluation was performed in both domains of the project (see the evaluation datasets in Table 3a and the evaluation results in Table 3b).

<table>
<thead>
<tr>
<th>Testing Corpus</th>
<th>1st Domain</th>
<th>2nd Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pages, Offers, NEs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>English</td>
<td>50,423,5111</td>
<td>50,110,900</td>
</tr>
<tr>
<td>Italian</td>
<td>50,267,3296</td>
<td>50,156,1219</td>
</tr>
<tr>
<td>Greek</td>
<td>50,136,1759</td>
<td>50,128,847</td>
</tr>
<tr>
<td>French</td>
<td>53,204,2400</td>
<td>50,166,1774</td>
</tr>
</tbody>
</table>

Table 3a: The evaluation datasets for the two domains

<table>
<thead>
<tr>
<th></th>
<th>1st Domain</th>
<th>2nd Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>English IE</td>
<td>48,8</td>
<td>50,7</td>
</tr>
<tr>
<td>French IE</td>
<td>-</td>
<td>66,0</td>
</tr>
<tr>
<td>Greek IE</td>
<td>58,5</td>
<td>53,5</td>
</tr>
<tr>
<td>Italian IE</td>
<td>79,9</td>
<td>67,8</td>
</tr>
</tbody>
</table>

Table 3b: Overall IE Evaluation Results for both domains

Full IE is a complex task which becomes even more complicated by the visual nature of web pages. Taking into account the complexity of the task, we believe that the level of performance achieved by all systems can provide quality IE results, especially for product/offer descriptions extracted from simpler web pages. In addition, the existing systems can be tuned further in order to achieve better performance and CROSSMARC provides various tools to support this task.

2.4 Data Storage

The data storage component stores the extracted facts, from the XML file produced by IE, into domain-specific databases according to the following principles: (a) all facts belonging in the same product/offer form a record in the database, (b) in each record the page URL and the storage date-time are also kept, (c) all records extracted from the same web page form a dataset. The main operations of the data storage component are: (a) insert a dataset (all records from a dataset are deleted and the new ones extracted from the same page are added), (b) delete a dataset, (c) delete a record, (d) query the database. The addition of a new domain is done for both the data storage and the data presentation components. The system administrator will have to register the ontology for the new domain and the corresponding FE schema in the appropriate configuration file.

2.5 Personalization

CROSSMARC User Interface (UI) presents the extracted information according to the user’s preferences. Personalization is performed by a general-purpose personalization server (PServer) which is integrated with the UI. More specifically:

- UI remembers user choices from previous sessions, through user profiles maintained by the Pserver,
users can be informed about the most popular choices of their group, through stereotype profiles maintained by the Pserver,

UI is adapted according to stereotype definitions

stereotype definitions are created and maintained using the Stereotypes editor of the Ontology Management system taking into account the stereotype profiles.

Concerning personalization, customization of the UI to a new domain involves the following tasks: (a) create the stereotype definitions for the new domain and languages using the stereotypes editor of the ontology management system, (b) update the UI configuration file with the new domain information (e.g. address of the new ontology and lexicons, new stereotype definitions).

3. Infrastructure

CROSSMARC infrastructure for configuring to new domains / languages involves:

- the Protégé-based ontology management system for the creation and maintenance of the ontology, the lexicons and other ontology-related resources,
- a methodology and a tool for the formation of corpus necessary for the training and testing of the page filtering and link scoring modules in the spidering component,
- a methodology for the collection and annotation of corpus necessary for the training and testing of the information extraction components.

3.1 Ontology Management

Ontology has a key role in CROSSMARC processing stages (Hachey et al. 2003):

- in crawling, it is used in specifying the user queries in the 2nd component of the crawler;
- in spidering, it provides the domain-specific features that are used to characterize the Web pages as positive and negative, thus it plays a crucial role in the training of the page filtering module;
- in information extraction, (a) it specifies the important entities for the domain and provides the corresponding language specific instances to fill the gazetteers of known entities, (b) it is used for the normalization of the entities belonging in a specific product/offer;
- in data storage and presentation, the normalized values are translated to the end-user’s language exploiting the link between the ontology and the corresponding lexicon; the user stereotypes are also defined according to the domain ontology.

Thus ontology management in CROSSMARC should handle apart from the domain knowledge (domain concepts and instances) and the lexical knowledge (lexicalization of concepts and instances), the knowledge required for a specific processing stage (e.g. specify entities for NERC, facts to be extracted) and the knowledge for the user preferences (e.g. preferences of a group of users according to the ontology). For this purpose, our Protégé-based ontology management system involves the following editors and functionalities:

- ontology editor for the creation and maintenance of domain ontologies;
- lexicon editor for the creation and maintenance of lexicons under domain ontologies;
- NERC editor for the creation of the NERC DTD;
- Template editor for the creation of the FE XML schema;
- Stereotypes editor for the creation and maintenance of the user stereotype definition according to the ontology;
- functionalities for exporting the ontology in XML, the lexicon(s) in XML, the NERC DTD, the FE XML schema, the XML stereotype definition.

3.2 Corpus Formation for Spidering

CROSSMARC devised a methodology for the formation of the corpus of web pages in a specific domain and language that are used for the training and testing of the page filtering and link scoring modules of the spidering component. The formation of such a corpus is not always a trivial task, particularly so for

3 See (Fridman et al. 2000).
Web page classification. As a simple example of the difficulties that one faces, in the domain of identifying product description pages one needs to select pages that are representative of the positive class, i.e., good product description pages, and also pages that are representative of the opposite class. The latter task, i.e., the selection of good negative instances, is very important for the quality of the resulting classifier and at the same time very difficult, due to the variety of pages that do not belong to the positive class. Unless the person responsible for the construction of the dataset combines expertise in the domain and in machine learning methods, the construction process will require particular effort and may lead to sub-optimal results. For this purpose, we devised a simple approach that aims to facilitate the formation of training corpus for Web page classification. This approach is based on an interactive process between the user (person responsible for corpus formation) and a simple classifier, which is responsible for selecting Web pages and presenting them to the user for classification. This approach is realized by the Corpus Formation Tool, which helps users build a corpus of positive and negative pages, with respect to a given domain.

3.3 Corpus Collection and Annotation for Information Extraction
CROSSMARC devised a corpus collection methodology for the compilation of training and testing corpora that are up-to-date, and representative of interesting product descriptions appearing in active web sites. It determines how different web pages characteristics must be taken into account and how they are to be represented in the training and testing corpora by establishing a set of rules to be followed in the formation of the corpora. The domain dependent resources (ontology, NERC DTD and FE XML Schema) are also taken into account.

The creation of consistently annotated corpora is very important for the training and evaluation of IE systems. The corpus annotation methodology that has been developed and followed at CROSSMARC is comparable to standard annotation practice for the purposes of Information Extraction. The annotation task is based on Annotation Guidelines that are issued for each new domain. Two human annotators use the guidelines in order to annotate the same pages. Then a third person, preferably the one responsible for the annotation guidelines inspects the annotations produced and give further instructions on the creation of the final annotations. CROSSMARC provides annotators with the Web Annotator tool (Sigletos et al. 2003a). This tool was used to annotate CROSSMARC corpora in the two domains and four languages of the project.

4. Concluding Remarks
CROSSMARC produced a set of tools, resources, methodologies that can be further exploited for research or commercial purpose. Based on the experimental results in its main processing stages, several conclusions were deduced which can be used by other researchers in the area. The integrated prototype and its components can be accessed and tested through the CROSSMARC site. In addition, access will also be provided to the collected and annotated corpora for the four languages and the two domains of the project. These corpora will be accessible for research purposes from the CROSSMARC site, forming a benchmark for experiments in the areas of web crawling, spidering, information extraction. In this way several other researchers in the area could take advantage of CROSSMARC work in these areas.

Acknowledgements
We would like to thank all our colleagues from the CROSSMARC consortium for the project work we carried out successfully together.

References


Soderland S. Learning to extract text-based information from the world wide web. In Proceedings of 3rd International Conference in Knowledge Discovery and Data Mining (KDD-97), pages 251–254.
