ORIGINAL RESEARCH

Exploiting web scraping in a collaborative filtering-based approach to web advertising

Eloisa Vargiu[1, 2], Mirko Urru[1]

1. Dipartimento di Matematica e Informatica, Università di Cagliari, Italy. 2. Barcelona Digital Technology Centre, Spain

Correspondence: Eloisa Vargiu. Address: Barcelona Digital Technology Center, Italy. Email: evargiu@bdigital.org.

Received: June 30, 2012 **Accepted:** August 6, 2012 **Online Published:** December 5, 2012
DOI: 10.5430/air.v2n1p44 **URL:** http://dx.doi.org/10.5430/air.v2n1p44

Abstract

Web scraping is the set of techniques used to automatically get some information from a website instead of manually copying it. The goal of a Web scraper is to look for certain kinds of information, extract, and aggregate it into new Web pages. In particular, scrapers are focused on transforming unstructured data and save them in structured databases. In this paper, among others kind of scraping, we focus on those techniques that extract the content of a Web page. In particular, we adopt scraping techniques in the Web advertising field. To this end, we propose a collaborative filtering-based Web advertising system aimed at finding the most relevant ads for a generic Web page by exploiting Web scraping. To illustrate how the system works in practice, a case study is presented.

Key words

Web advertising, Collaborative filtering, Web scraping

1 Introduction

Web Advertising is an emerging research field, at the intersection of information retrieval, machine learning, optimization, and microeconomics. It is one of the major sources of income for a large number of websites. Its main goal is to suggest products and services to the ever growing population of Internet users.

There are two primary channels for distributing ads: sponsored search (or paid search advertising) and contextual advertising (or content match). Sponsored search advertising displays ads on the page returned from a Web search engine following a query; whereas contextual advertising displays ads within the content of a generic, third part, Web page. A commercial intermediary, namely ad network, is usually in charge of optimizing the selection of ads with the twofold goal of increasing revenue and improving user experience. The ads are selected and served by automated systems based on the content displayed to the user.

Web scraping (also called Web harvesting or Web data extraction) is a software technique aimed at extracting information from websites [1]. Usually, Web scrapers simulate human exploration of the World Wide Web by either implementing low-level hypertext transfer protocol or embedding suitable Web browsers. Web scraping is closely related to Web indexing, which is an information retrieval technique adopted by several search engines to index information on the Web through a bot. In contrast, Web scraping focuses on the transformation of unstructured data on the Web, typically in HTML format, into structured data that can be stored and analyzed in a central local database or spreadsheet. Web scraping

is also related to Web automation [2], which simulates human Web browsing using computer software. Web scraping is currently used to online price comparison, weather data monitoring, website change detection, Web research, Web mashup, and Web data integration. Several (commercial) software tools, aimed at personalizing websites by adopting scraping techniques, are currently available.

In this paper, we present a collaborative filtering-based Web advertising system that exploits Web scraping techniques to suggest suitable ads to a given Web page. In particular, we address Web advertising as an information filtering task devising our proposed Web advertising system by exploiting collaborative filtering [3]. The proposed system, first, exploits collaborative filtering and, subsequently, relies on Web scraping to extract ads to be suggested. The idea to exploit collaborative filtering in a Web advertising has been proposed by Armano & Vargiu [4] and adopted also in Armano et al. [5]. To our best knowledge this is the first attempt to adopt Web scraping techniques to perform Web advertising. The underlying motivation in adopting Web scraping is that, in case of no available ad dataset (available only for companies that operate advertising systems, e.g., Yahoo!, Google, or Microsoft, not for academic purposes), instead of building an ad-hoc dataset by hand, this unsupervised approach could be adopted.

The rest of the paper is organized as follows. Section 2 summarizes the main work on Web advertising, collaborative filtering and Web scraping. In Section 3, we illustrate our proposed Web advertising system, focusing on its architecture and the exploitation of collaborative filtering and Web scraping. In Section 4, we illustrate a case study aimed at highlighting the effectiveness of the proposed approach. Section 5 ends the paper with conclusions and future research directions.

2 Background

In this Section, we give a brief overview on the main topics addressed in this paper: Web advertising, collaborative filtering, and Web scraping.

2.1 Web advertising

From the beginning of the Web era, companies put graphical banner ads on Web pages at popular websites [6]. Banner advertising is a form of Web advertising that entails embedding an ad into a Web page [7]. It is intended to attract traffic to a website by linking it to the website of the advertiser. The ad is constructed from an image (GIF, JPEG, PNG), JavaScript program or multimedia object. Moreover, they often employ animation, sound, or video to maximize presence. The primary purpose of these ads was branding, i.e., to convey to the viewer a positive feeling about the brand of the company placing the ad. These ads were, typically, priced on a cost per mil basis, i.e., the cost to the company of having its banner ad displayed 1000 times. Some websites made contracts with their advertisers in which an ad was priced not by the number of times it is displayed, but rather by the number of times it was clicked on by the user (cost per click model). In such cases, clicking on the ad leads the user to a Web page set up by the advertiser, where the user is induced to make a purchase. Here, the goal of the ad is not so much brand promotion as to induce a transaction.

To formulate the Web advertising problem, let P be the set of Web pages and let A be the set of ads that can be displayed. The revenue of the network, given a page p, can be estimated as:

$$R = \sum_i^k \Pr(click \mid p, a) \cdot price(a_i, i) \quad (1)$$

where k is the number of ads displayed on page p and $price(a_i, i)$ is the click-price of the current ad a_i at position i. The price in this model depends on the set of ads presented on the page. Several models have been proposed to determine the price, most based on generalizations of second price auctions. For the sake of simplicity, we ignore the pricing model and reformulate the Web advertising problem as follows: for each page $p \in P$ we want to select the ad $a' \in A$ hat maximizes the click probability. Formally:

$$\forall p \in P: a'_p = argmax_{a \in A} \Pr(\text{click}|\text{p}, \text{a}) \quad (2)$$

An alternative way to formulate the problem is the following: let *P* be the set of Web pages and let *A* be the set ads that can be displayed. Let *f* be a utility function that measures the matching of *a* to *p*, i.e. $f: PXA \rightarrow R$, where *R* is a totally ordered set (e.g., non-negative integers or real numbers within a certain range). Then, for each page $p \in P$ we want to select $a' \in A$ that maximizes the page utility function. More formally:

$$\forall p \in P: a'_p = argmax_{a \in A} f(p.a) \quad (3)$$

Note that in this case the utility function *f* can also be viewed as an estimation of the probability that the corresponding ad be clicked.

A significant part of Web advertising consists of textual ads, the ubiquitous short text messages usually marked as sponsored links. There are two primary channels for distributing ads: sponsored search (or paid search advertising) and contextual advertising (or content match). Sponsored search advertising displays ads on the page returned from a Web search engine following a query. It can be thought as a document retrieval problem, where ads are documents to be retrieved in response to a query. Ads could be represented in part by their keywords. Carrasco et al. [8] approached the problem of keyword suggestion by clustering bipartite advertiser-keyword graphs. Joachims [9] proposed to use click-data to learn ranking functions for results of a search engine as an indicator of relevance. Ciaramita et al. [10] studied an approach to learn and evaluate sponsored search systems based solely on click-data, focusing on the relevance of textual content. Contextual advertising is a form of targeted advertising for ads appearing on Websites or other media, such as content displayed in mobile browsers. The ads themselves are selected and served by automated systems based on the content displayed to the user. Ribeiro-Neto et al. [11] examined a number of strategies to match pages and ads based on extracted keywords. In a subsequent work, Lacerda et al. [12] proposed a method to learn the impact of individual features using genetic programming. Broder et al. [13] classified both pages and ads into a given taxonomy and matched ads to the page falling into the same node of the taxonomy. Starting from that work, Armano et al. [14] proposed a semantic enrichment by adopting concepts. Furthermore, modern contextual advertising systems use text summarization techniques in conjunction with the model developed in Broder et al. [13], see, for instance Anagnostopoulos et al. [15], Armano et al. [16], Armano et al. [17]. Since bid phrases are basically search queries, another relevant approach is to view contextual advertising as a problem of query expansion and rewriting [18, 19].

2.2 Collaborative filtering

Collaborative filtering consists of automatically making predictions (filtering) about the interests of a user by collecting preferences or tastes from similar users (collaboration); the underlying idea is that those who agreed in the past tend to agree again in the future.

Several collaborative filtering systems have been developed to suggest items and goods, including news, photos, people, and books [20]. Collaborative filtering systems try to predict the utility of items for a particular user based on the items previously rated by other users. There have been many collaborative recommender systems developed in the academia and in the industry. Among others, let us recall: the Grundy system [21], GroupLens [22], Video Recommender [23], and Ringo [24] which have been the first systems that used collaborative filtering algorithms to automate recommendation. Other examples of collaborative recommender systems are: the book recommender system from Amazon.com [25] and the PHOAKS system that helps people to find relevant information on the Web [26]. In particular, collaborative recommender systems are the most suitable if the items to be recommended are multimedia with scarce descriptions, but rated by a community of users [27].

Let us note that analogously to the Web advertising problem, the recommendation problem can be formulated as follows: let *U* be the set of all users and let *I* be the set of all possible items that can be recommended, e.g., books, movies, and

restaurants (it can be very large, ranging in hundreds of thousands or even millions of items in some applications). Let f be a utility function that measures the usefulness of item i for user u, i.e.,$f: UxI \rightarrow R$, where R is a totally ordered set (e.g., non-negative integers or real numbers within a certain range). Then, for each user $u \in U$ we want to choose the item $i' \in I$ that maximizes the user's utility function. More formally:

$$\forall u \in U: i'_u = argmax_{i \in I} f(u, i) \quad (4)$$

In which the utility function is typically represented by ratings and is initially defined only on items previously rated by the users. For example, in a book recommendation application (e.g., Amazon.com), users initially rate some subsets of books they have read.

According to the analogy in those definitions, a few works that use collaborative filtering to perform Web advertising have been proposed. As for sponsored search, Anastasakos et al. [28] proposed a technique to determine the relevance of an ad document for a search query using click-through data. In their work, collaborative filtering is exploited to discover new ads related to a query using a click graph. As for contextual advertising, Armano & Vargiu [4] proposed to study the problem of contextual advertising as a recommendation problem, and vice versa.

2.3 Web scraping

Nowadays, quite a lot of researchers are working on extracting information about types of events, entities or relationships from textual data. Information extraction is used for search engines, news libraries, manuals, domain-specific text or dictionaries. A form of information extraction is text mining, an information retrieval task aimed at discovering new, previously unknown information, by automatically extracting it from different text resources [29]. In information extraction, text mining is used to scrap relevant information out of text files by relying on linguistic and statistic algorithms.

Web search and information extraction is typically performed by Web crawlers. A Web crawler is a program or automated script that browses the WWW in a methodical, automated manner [30]. A more recent variant of Web crawlers are Web scrapers, which are aimed at looking for certain kinds of information–such as prices of particular goods from various online stores—extracting, and aggregating it into new Web pages [31].

Scrapers are basically adopted to transform unstructured data and save them in structured databases. In screen scraping, a special form of scraping, a program extracts information from the display output of another program [32]. So that, the output which is scraped is created for the end user and not for other programs that is the difference to a normal scraper. In this paper, we focus on Web scrapers that extract textual information from Web pages. There are many methods to scrap information from the Web [33]. Since barriers to prevent machine automation are not effective against humans, the most effective method is human copy-paste. Although sometimes this is the only way to export information from a Web page, this is not feasible in practice, especially for big company projects, being too expensive. Another method is text grepping in which regular expressions are used to find information that matches some patterns. Further Web scraping techniques are HTTP programming, DOM parsing, and HTML parsers. Finally, a Web scraping method consists of making scraper sites that are automatically generated from other Web pages by scraping their content [34].

It is worth noting that Web scraping may be against the terms of use of some websites. Being interested in the scientific issues concerned with the adoption of Web scraping to perform Web advertising, in this paper we do not take into account legal issues on adopting and implementing Web scraping techniques.

3 The proposed approach to web advertising

Our proposal is to exploit Web scraping to suggest suitable ads to a given Web page. To this end, we address Web advertising as a collaborative filtering task. In Web advertising, given a Web page *p*, relevant information are ads related to

p (i.e., to its content) and the third part is *p* itself, since it will display the filtered ads. Thus, we propose a Web advertising system that relies on collaborative filtering and exploits scraping techniques to analyze the page content. In particular, we decided to apply collaborative filtering for first and to subsequently rely on Web scraping to perform a content-based analysis. In so doing, given a Web page *p*, the collaborative filtering module exploits the collaboration of *p* retrieving a subset of its peer pages. The content of the retrieved peer pages is then analyzed by adopting Web scraping techniques.

For the sake of rapid prototyping, the system has been implemented in Python, the implemented modules and their connections are depicted in Figure 1. As shown, the system is composed of three modules: (i) inlink extractor, (ii) ad extractor, and (iii) ad selector.

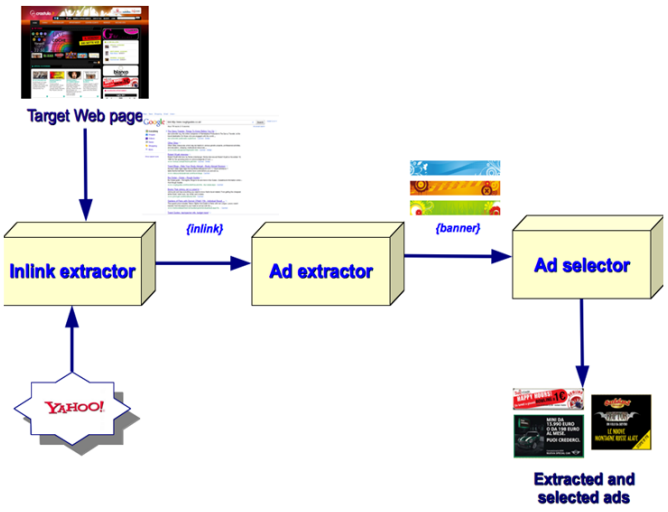

Figure 1. The architecture of the proposed system.

Inlink extractor. This module is devoted to find, given a page *p*, the peer pages. Suitable peer pages appear to be all the inlinks of *p* (also called backlinks), i.e., all pages that link to *p*. The inlink extractor collects the first 10 inlinks of a given page by relying on the Yahoo! Site Explorer. The adopted collaborative filtering approach is illustrated in Section 3.1.

The inlink extractor gives as output the list of the 10 extracted peer pages that will be scrapped by the Ad extractor in order to identify the most related ads.

Ad extractor. This module is aimed at extracting banner ads from the peer pages. To this end, we rely on Web scraping, i.e., a set of techniques used to automatically get some information from a website instead of manually copying it. In particular, we adopt scraping techniques to: (i) access tags as object members; (ii) find out tags whose name, contents or attributes match some selection criteria; and (iii) access tag attributes by using a dictionary-like syntax. Scraping is performed by using Python libraries provided by HTMLParser and Beautiful Soup. The adopted Web scraping techniques will be explained in details in Section 3.2.

The ad extractor module gives as output an ordered set of the extracted banners, together with the corresponding url and their descriptions. This set will be analyzed by the ad selector that selects the three banners to be inserted in the original webpage.

Ad selector. This module is aimed at selecting suitable banner ads from the set extracted by the ad extractor. To this end, several policies might be applied. In the current version of the system, we decide to adopt a random mechanism that selects and provides three ads.

 ISSN 1927-6974 E-ISSN 1927-6982

Let us note that how to insert the banner in the given page is out of the scope of this paper, being it dependent on the server that provides that page.

3.1 Collaborative filtering to extract similar pages

We exploit collaborative filtering to select the most relevant pages related to a given page *p*. The underlying idea is that, a page *p1* links *p* (i.e., it is an inlink of *p*) if the topics of *p1* are related to the topics of *p* [35]. Vice versa, *p* links a page *p2* (i.e., it is an outlink of *p*) if their topics are in some relationships.

Figure 2 gives a view on the all possible kinds of related links. Two kinds of inlinks and outlinks may exist: those that link to an external domain (i.e., from A and to B in the Figure) and those that link to the same domain of the target webpage (i.e., from T1 and to T2 in the Figure). In this work, we consider only inlinks belonging to different domains. In other words we disregard inlinks that come from the same Web domain and outlinks, since –statistically speaking– inlinks are more informative than outlinks [36].

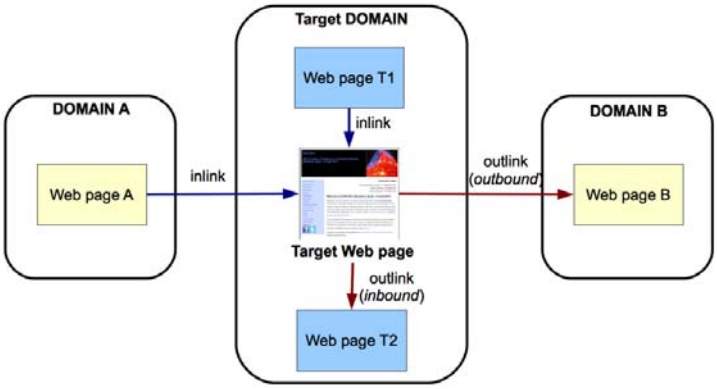

Figure 2. A graphical view of related links.

3.2 Web scraping to extract banner ADS

The ad extractor module takes as input all the extracted inlinks and analyzes them to extract the information related to all the embedded ads. In particular, in this work we are interested in extracting banner ads. Thus, the module looks for HTML anchor tag <a> and selects those that refer to an image:

where brand_url is the url of a company or a service to be advertised and banner_ad_url is the url of the image containing the ad (i.e., the banner).

To extract the HTML code, the ad extractor relies on Web scraping through two specialized libraries: HTMLParser and Beautiful Soup. HTMLParser defines a class HTMLParser that serves as the basis for parsing text files formatted in HTML and XHTML. The class is instantiated without arguments and its instance is fed by HTML data and calls handler functions when tags begin and end. The class is meant to be overridden by the user to provide a desired behavior. BeautifulSoup is a Python library that parses broken HTML. BeautifulSoup is not a real HTML parser but uses regular expressions to dive through tag soup. The main features of BeautifulSoup are: it yields a parse tree that makes approximately as much sense as the original document, in case of the programmer gives it bad markup; it provides a few simple methods and Pythonic idioms for navigating, searching, and modifying a parse tree; and it automatically converts incoming documents to Unicode and outgoing documents to UTF-8.

First, the ad extractor retrieves all the links embedded in the Web page:

```
bs = BeautifulSoup(page)
divs = bs.findAll('span', attrs={ 'herf', 'class' : 'result'})
results = []
i = 0
while i < 10:
   soup = BeautifulSoup(repr(divs[i]))
   results.append(soup.findAll('a', attrs = {'href' : True})[0]['href'])
   i = i + 1
```

After its creation, bs, an instance of class BeautifulSoup, contains a well-formed HTML code of the selected page. From *bs*, the HTML code that contains the results of the query is extracted and saved in divs. Finally, the first 10 ininks have been extracted and appended in the results array.

Then, the module scraps each inlinks:

```
i = 0
while i < 10:
   p = urlopen(results[i]).read()
   b = BeautifulSoup(p)
   j = 0
   link = b.findAll('a', attrs = {'href': True})
   links = []
   for x in link:
      soup = BeautifulSoup(repr(x))
      links.append(soup.findAll(['a','img']))
```

From each inlink, the source code is extracted, saved in *p*, and then well-formed (*b*). All the links are then extracted and those that are images are saved in the links array.

Figure 3. The two ways of representing an ad.

Finally, the extracted ads are collected in an ads repository and, once selected by the ad selector, they can be put in the original Web page in two ways: (i) as banners, by simply presenting the retrieved images, or (ii) as textual ads, by composing the corresponding url, its title and its snippet retrieved by asking to the Yahoo! search engine. Figure 3 shows the two ways for representing the same ad.

4 A case study

Let us note that due to the unsupervised nature of the proposed approach, a fair comparison with a classical Web advertising system is not feasible. In fact, Web advertising systems use a pre-indexed or pre-classified set of ads, whereas

 ISSN 1927-6974 E-ISSN 1927-6982

in the current approach the system gathers ads directly from the webpages without any further information. Thus, to show how the proposed system works, we propose a suitable case study.

Let us consider the task of suggesting ads to a given Web page, i.e., the home page of the portal Crastulo (http://www.crastulo.it/). Crastulo is a local portal that collects information on events in Cagliari, such as concerts, shows, and openings. A fragment of the home page is depicted in Figure 4.

Figure 4. A fragment of the webpage adopted in the case study.

First, the proposed system queries Yahoo! asking for the first 10 inlinks belonging to a different domain of the given page (see Figure 5). Subsequently, the ad extractor performs scraping to capture all the banner ads contained in each inlink. Finally, three ads are randomly selected to be displayed in the original Web page. Figure 6 shows the three selected ads represented by their banners.

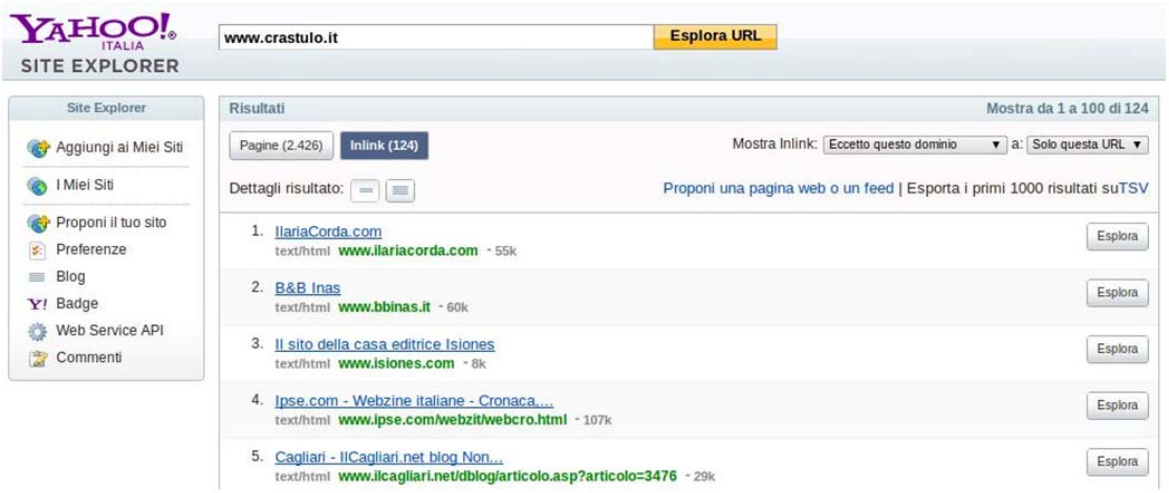

Figure 5. The inlinks of the webpage in hand proposed by Yahoo! site explorer.

Figure 6. The suggested retrieved ads.

5 Conclusions and future work

In this paper, we presented a novel collaborative filtering-based Web advertising system that exploits Web scraping techniques to suggest suitable ads to a given Web page. Considering a Web advertising task as information filtering one, we devised a Web advertising system that adopts collaborative filtering features. The proposed system, first, relies on collaborative filtering by exploiting peer pages and, subsequently, it resorts to Web scraping to perform the page content analysis. To show how the system works in practice, we presented a suitable case study, i.e., how to suggest banner ads to the home page of the Italian portal Crastulo. To our best knowledge this is the first attempt to exploit Web scraping techniques to perform Web advertising.

As for the future work, we are setting up experiments aimed at calculating the performances of the proposed system in term of precision at k, i.e., the ability of the system in suggesting k relevant ads, varying k in [1-5]. In particular, we are interested in selecting a set of users, asking them to give a degree of relevance to each retrieved ads, e.g., relevant, somewhat relevant, or irrelevant. Moreover, further research directions could be concerned with adapting the proposed system to social networks (such as Facebook, Google+, or Twitter) to suggest ads according to users' preferences and tastes.

References

[1] Schrenk, M. Webbots, spiders, and screen scrapers: a guide to developing Internet agents with PHP/CURL. No Starch Press, 2007.

[2] Bolin, M., Webber, M., Rha, P., Wilson, T. & Miller, R.C. Automation and customization of rendered web pages. Proceedings of the 18th annual ACM symposium on User interface software and technology, UIST '05, pp. 163-172. ACM, New York, NY, USA, 2005.

[3] Goldberg, D., Nichols, D., Oki, B.M. & Terry, D. Using collaborative filtering to weave an information tapestry. Commun. ACM. 1992; 35(12): 61-70. http://dx.doi.org/10.1145/138859.138867

[4] Armano, G. & Vargiu, E. A unifying view of contextual advertising and recommender systems. Proceedings of International Conference on Knowledge Discovery and Information Retrieval (KDIR 2010). 2010: 463-466.

[5] Armano, G., Giuliani, A. & Vargiu, E. Intelligent Techniques in Recommendation Systems: Contextual Advancements and New Methods, chap. IntelligentTechniques in Recommender Systems and Contextual Advertising: Novel Approaches and Case Studies. S. Dehuri, M.R. Patra, B.B. Misra, A.K. Jagadev (eds.), IGI Global. 2012: 105-128.

[6] Manning, C.D., Raghavan, P. & Schtze, H. Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA, 2008. http://dx.doi.org/10.1017/CBO9780511809071

[7] Reid, R.H. Architects of the Web: 1,000 Days that Built the Future of Business. Wiley, 1997.

[8] Carrasco, J., Fain, D., Lang, K. & Zhukov, L. Clustering of bipartite advertiser-keyword graph. Proc. International Conference on Data Mining (ICDM'03). Melbourne, Florida, 2003.

[9] Joachims, T. Optimizing search engines using clickthrough data. ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD). 2002: 133-142.

[10] Ciaramita, M., Murdock, V. & Plachouras, V. Online learning from click data for sponsored search. Proceeding of the 17th international conference on World Wide Web, WWW '08. ACM, New York, NY, USA. 2008; 227-236.

[11] Ribeiro-Neto, B., Cristo, M., Golgher, P.B. & Silva de Moura, E. Impedance coupling in content-targeted advertising. SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, New York, NY, USA. 2005; 496-503.

[12] Lacerda, A., Cristo, M., Gonçalves, M.A., Fan, W., Ziviani, N. & Ribeiro-Neto, B. Learning to advertise. SIGIR '06: Proceedings of the 29th annual international ACM SIGIRconference on Research and development in information retrieval, ACM, New York, NY, USA. 2006; 549-556.

[13] Broder, A., Fontoura, M., Josifovski, V. & Riedel, L. A semantic approach to contextual advertising. SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, New York, NY, USA. 2007; 559-566. http://dx.doi.org/10.1145/1277741.1277837

[14] Armano, G., Giuliani, A. & Vargiu, E. Semantic enrichment of contextual advertising by using concepts. International Conference on Knowledge Discovery and Information Retrieval, 2011.

[15] Anagnostopoulos, A., Broder, A.Z., Gabrilovich, E., Josifovski, V. & Riedel, L. Just-in-time contextual advertising. CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management. ACM, New York, NY, USA, 2007: 331-340. http://dx.doi.org/10.1145/1321440.1321488

[16] Armano, G., Giuliani, A. & Vargiu, E. Studying the impact of text summarization on contextual advertising. 8th International Workshop on Text-based Information Retrieval, 2011.

[17] Armano, G., Giuliani, A. & Vargiu, E. Using snippets in text summarization: a comparative study and an application. Italian Workshop on Information Retrieval (IIR 2012), 2012.

[18] Murdock, V., Ciaramita, M., Plachouras, V. A noisy-channel approach to contextual advertising. Proceedings of the 1st international workshop on Data mining and audience intelligence for advertising, ADKDD '07. ACM, New York, NY, USA, 2007: 21-27.

[19] Ciaramita, M., Murdock & V., Plachouras, V. Semantic associations for contextual advertising. Journal of Electronic Commerce Research. 2008; 9(1): 1-15.

[20] Adomavicius, G. & Tuzhilin, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering. 2005; 17(6): 734-749. http://dx.doi.org/10.1109/TKDE.2005.99

[21] Rich, E. User modeling via stereotypes, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.

[22] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. & Riedl, J. Grouplens: an open architecture for collaborative filtering of netnews. CSCW '94: Proceedings of the 1994 ACM conference on Computer supported cooperative work. ACM, New York, NY, USA. 1994; 175-186. http://dx.doi.org/10.1145/192844.192905

[23] Hill, W., Stead, L., Rosenstein, M. & Furnas, G. Recommending and evaluating choices in a virtual community of use. CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA. 1995; 194-201.

[24] Shardanand, U. & Maes, P. Social information filtering: algorithms for automating “word of mouth”. CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1995; 210-217.

[25] Linden, G., Smith, B. & York, J. Amazon.com recommendations. IEEE Internet Computing. 2003; 07(1): 76-80. http://dx.doi.org/10.1109/MIC.2003.1167344

[26] Terveen, L., Hill, W., Amento, B., McDonald, D. & Creter, J. Phoaks: a system for sharing recommendations. Communication of ACM. 1997; 40(3): 59-62. http://dx.doi.org/10.1145/245108.245122

[27] Sarwar, B., Karypis, G., Konstan, J. & Reidl, J. Item-based collaborativefiltering recommendation algorithms. WWW '01: Proceedings of the 10th international conference on World Wide Web. ACM, New York, NY, USA. 2001; 285-295. http://dx.doi.org/10.1145/371920.372071

[28] Anastasakos, T., Hillard, D., Kshetramade, S. & Raghavan, H. Collaborative filtering approach to ad recommendation using the query ad click graph. Proceedings of CIKM 2009, 2009. http://dx.doi.org/10.1145/1645953.1646267

[29] Berry, M.W. Survey of Text Mining. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.

[30] Kobayashi, M. & Takeda, K. Information retrieval on the web. ACM Comput. Surv. 2000; 32: 144-173. http://dx.doi.org/10.1145/358923.358934

[31] Adams, A. & McCrindle, R. Pandora's box: social and professional issues of the information age. John Wiley & Sons, 2008.

[32] Lewerenz, E. An example of website screen scraping. Proceedings of MWSUG 2009, 2009.
[33] Mehlfuehrer, A. Web scraping - a tool evaluation. Master's thesis, Wien University, 2009.
[34] Penman, R.B. Web scraping made simple with sitescraper. Text, 2009.
[35] Koolen, M. & Kamps, J. Are semantically related links more effective for retrieval? Proceedings of the 33rd European Conference on Advances in information retrieval, ECIR'11. Springer-Verlag, Berlin, Heidelberg. 2011: 92-103.
[36] Armano, G., Giuliani, A. & Vargiu, E. Are related links effective for contextual advertising? a preliminary study. International Conference on Knowledge Discovery and Information Retrieval, 2012.

 ISSN 1927-6974 E-ISSN 1927-6982