

# Flint: Google-Basing the Web

Lorenzo Blanco, Valter Crescenzi, Paolo Merialdo, Paolo Papotti  
Roma Tre University - Italy

[blanco,crescenz,merialdo,papotti]@dia.uniroma3.it

## ABSTRACT

Several Web sites deliver a large number of pages, each publishing data about one instance of some real world entity, such as an athlete, a stock quote, a book. Even though it is easy for a human reader to recognize these instances, current search engines are unaware of them. Technologies for the Semantic Web aim at achieving this goal; however, so far they have been of little help in this respect, as semantic publishing is very limited.

We have developed a system, called FLINT, for automatically searching, collecting and indexing Web pages that publish data representing an instance of a certain conceptual entity. FLINT takes as input a small set of labeled sample pages: it automatically infers a description of the underlying conceptual entity and then searches the Web for other pages containing data representing the same entity. FLINT automatically extracts data from the collected pages and stores them into a semi-structured self-describing database, such as Google Base. Also, the collected pages can be used to populate a custom search engine; to this end we rely on the facilities provided by Google Co-op.

## 1. INTRODUCTION

The proliferation of tools and facilities for publishing data on the Web is rapidly transforming a large portion of the Web in a huge collection of data-rich pages. Usually, the data delivered by these pages are exposed according to an implicit schema, and represent instances of some real world entity. While the richness of data contained in this increasing portion of the Web represents a promising opportunity, there is a lack of tools and techniques that support the consumption of the information it offers.

Wrapping techniques have been recently developed for extracting and annotating data from semi-structured Web pages (e.g. [2, 13, 17, 22], see [10] for a recent survey on the topic). These techniques take as input a set of pages that share a common template, infer the implicit schema according to which data are organized in the pages, and generate a wrapper that can be used to extract the data from any page that share the same template of the input set. However, the adoption of these techniques at a large scale is not feasible mainly because so far the wrapping approach is not supported

by effective techniques for discovering, crawling and indexing the collections of pages containing information of interest.

This paper presents FLINT, a system to support users in the tasks of discovering, annotating and indexing data-rich pages publishing information of interest. The system is domain independent, and the only required input is a small set of sample pages, each one containing data about one instance of an entity of interest. The system automatically infers an intensional summary of the target entity from the input pages, and searches the Web for pages publishing data representing instances of such an entity. The retrieved pages are then indexed and semantically annotated. Also, based on manual annotations that can be performed with a minimal manual effort by the user, the values of relevant attributes of the underlying conceptual entity are extracted and stored in a suitable database. The information contained in the indexed pages can thus be searched with the traditional IR approach, or by database style queries against the extracted data.

To give an example consider the three Web pages in Figure 1: the data published in each of them describe one instance of the HOCKEYPLAYER conceptual entity. FLINT automatically searches the Web for pages publishing data that represent instances of the same HOCKEYPLAYER entity. The retrieved pages are then indexed and annotated as instances of the HOCKEYPLAYER conceptual entity.

Also, the user can mark (by means of a suitable GUI) on the sample pages values of attributes of the target entity. Based on this information, the system infers a set of rules for extracting the values corresponding to the same attributes from all the retrieved pages. Again from our example, suppose the user labels on the sample pages the values of attributes such as `Position`, `Birth date`, `Height` and `Weight`. The system infers rules for extracting the values of these attribute from the retrieved pages (clearly from the pages that contain them).

As a proof of concept, we have implemented the system leveraging on some facilities that Google have recently launched: Google Co-op<sup>1</sup> and Google Base.<sup>2</sup>

- Google Co-op allows users to build a custom search engine over a domain of interest. With Google Co-op a user first specifies a set of labels (*Facet* in the Google terminology) each one representing a concept from the domain of interest, and then associates each label with a set of pages that fit the corresponding concept. Labels can then be explicitly used in the search process to restrict the results over a specific concept. In our context, we populate a Google Co-op search engine by associating the set of pages retrieved by FLINT with a suitable label (typically the name of the conceptual entity).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

EDBT'08, March 25–30, 2008, Nantes, France.

Copyright 2008 ACM 978-1-59593-926-5/08/0003 ...\$5.00.

<sup>1</sup><http://www.google.com/coop>

<sup>2</sup><http://base.google.com>

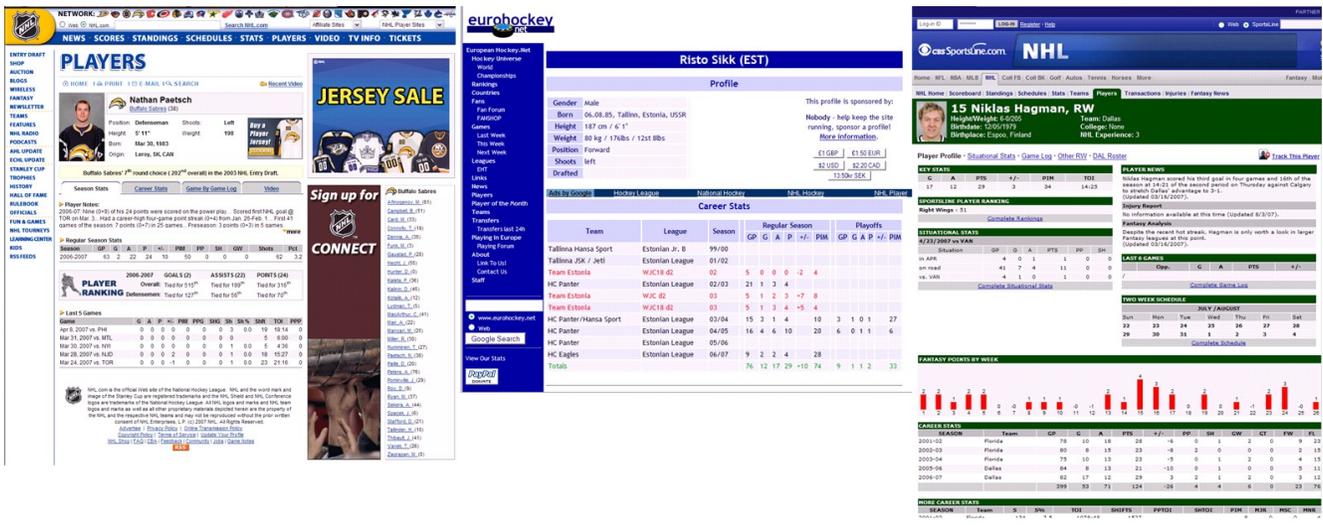


Figure 1: Three Web pages representing instances of the HOCKEYPLAYER entity

- Google Base can be seen as a self-describing, semi-structured database where users can upload their structured data. Google Base adopts a very simple data model: users describe their data using an item type and attribute/value pairs, without any restriction on names and values. In our context, Google Base is used to store the values extracted from the retrieved pages by FLINT. Each page gives rise to an item, of the same type of the corresponding entity, and the set of attribute/value pairs associated with each item corresponds to the data extracted from that page.

The search and querying facilities offered by Google Co-op and Google Base can thus be used to search and query the results of FLINT.

The rest of the paper is organized as follows. Section 2 provides a brief overview of our method for searching entities by sample; Section 3 discusses related work and Section 4 illustrates the demonstration scenario.

## 2. SYSTEM OVERVIEW

Figure 2 shows the main architectural components of FLINT. We now briefly comment the role of each component in the task of retrieving and annotating pages representing instances of the same conceptual entity as the input sample page. FLINT targets its effort against that vast portion of the Web which is composed by large, fairly structured Web sites, exploiting the regularities they exhibit.

FLINT works in four stages, as follows.

- First, it scans the Web sites of the sample pages in order to discover other pages representing instances of the same entity; this task is achieved by INDESIT [6].
- From the pages obtained in this initial stage, the ENTITYANALYZER module automatically extracts a description of the entity exemplified by the sample pages.
- Then, based on the entity description and on the pages collected in the first stage, the OUTDESIT module launches searches on the Web to discover new pages containing data that represent instances of the target entity. The pages found in this step are iteratively passed as input seeds to INDESIT, and this

interaction between INDESIT and OUTDESIT is recursively run, as long as new pages are found.

- Finally, the DATAEXTRACTOR module infers wrappers to extract annotated values from the collected pages.

As discussed in the previous Section, the extracted data and the retrieved pages are loaded into Google Base and Google Co-op, whose querying facilities can be used for searching the results computed by FLINT.

In the remainder of this section we provide some more details about the main components of the FLINT architecture.

### 2.1 INDESIT

INDESIT is responsible of seeking the Web site of a given seed page with the goal of collecting pages offering the same intensional information as the sample.

INDESIT relies on the observation that, within a large Web site, pages offering the same intensional information usually share a common template and common access paths. For example, consider the Web sites of the three sample pages shown in Figure 1: it is likely that in each of these Web sites, pages describing a hockey player have the same template as the corresponding sample page, and that the access paths to these pages are organized according to a common pattern.

Based on these ideas INDESIT implements a crawling algorithm that scans a given Web site toward pages sharing the same structure of an input seed page [6]. The crawler navigates the Web site searching for pages that contain lists of links leading to pages which are structurally similar to the seed page. These lists of links work like indexes to the searched pages. Therefore, INDESIT follows the links offered by these lists to collect the target set.

With respect to our running example, the output of INDESIT is the set of hockey player pages published in the Web sites of each sample page. In our perspective, each of these pages embeds data representing an instance of the HOCKEYPLAYER entity.

### 2.2 ENTITYANALYZER

Once INDESIT has collected a number of pages, the ENTITYANALYZER module computes a description of the conceptual entity for which the sample pages represent instances.

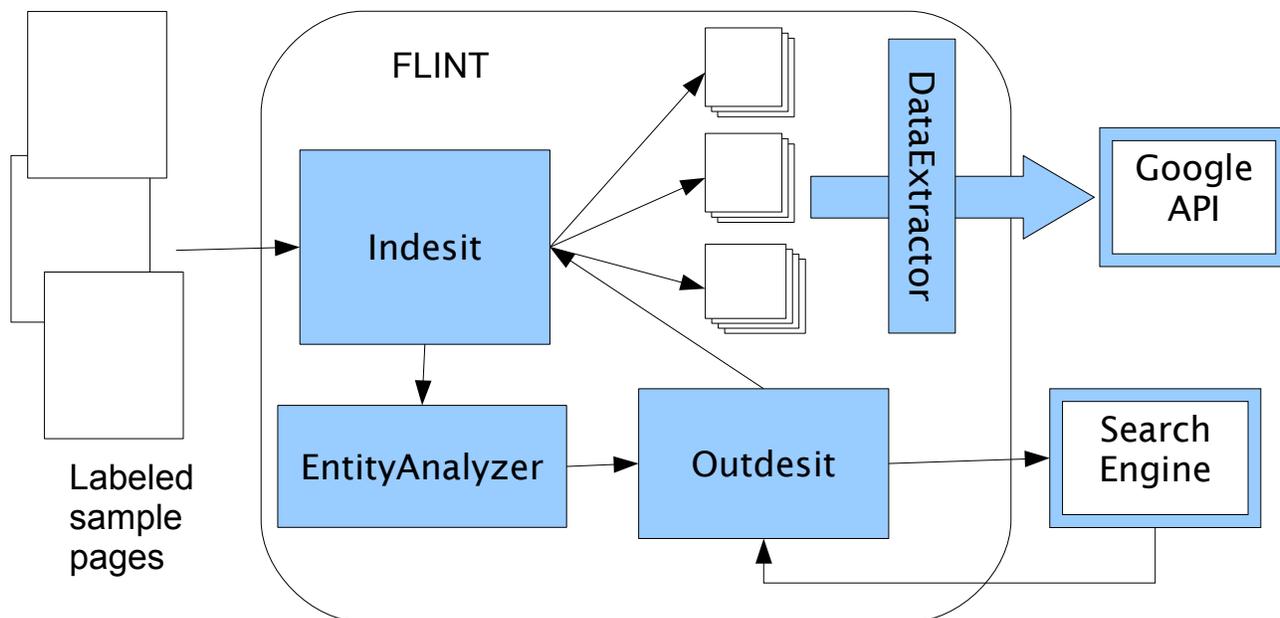


Figure 2: FLINT architecture

The description of a conceptual entity is composed by (i) an *intensional description*, and (ii) a *keyword*. The intensional description is expressed by means of a set of terms, each representing the name of an elementary feature, an attribute, of the entity (e.g., position, age, best score). The keyword is a term, extracted from the Web sites of the sample pages, that characterizes the overall conceptual domain of the entity (e.g., hockey).

Our method for extracting the entity description is based on the assumptions that different instances of the same conceptual entity are likely to share a common set of attributes, and that—as Web pages are produced for human consumption—several attribute names are explicitly published in the templates of the sample pages. It is worth saying that this phenomenon has been observed also by Madhavan *et al.* in their studies on Web scale data integration [19].

For each sample page, the ENTITYANALYZER computes the set of terms that belong to the corresponding template. This task is performed by analyzing the set of terms that occur in a bunch of structurally similar pages returned by INDESIT, and removing those elements that belong also to the “site template”, i.e. to that portion of the template that is shared by every page in the site. In this way, from each sample page a set of terms is extracted: their intersection is used as intensional description of the entity.

The entity description is completed by a keyword which is generated by analyzing, with standard term weighting techniques, the words that appear in a number of pages belonging to the Web sites of the input samples.

### 2.3 OUTDESIT

The results produced by the initial INDESIT execution, together with the inferred entity description, are used to propagate the search on the Web. This step is performed by OUTDESIT, which issues a set of queries against a search engine and elaborates the results in order to select only those pages that can be considered as instances of the target entity. Finally, the selected pages are used as seeds to trigger again an INDESIT scan, and the whole process is repeated

until new pages are found.

To correctly expand the search on the Web, FLINT needs to address several issues. First, it has to feed the search engine with keywords that are likely to produce new pages representing instances of the input entity. Second, as these pages will be used to run a new instance of INDESIT, it has to filter them in order to choose those that really correspond to instances of the input entity.

The keywords to be submitted to the search engine are generated by means of a simple yet effective technique. As we are searching for instances of a given entity, we need values that work as identifiers for the instances of the entity. We observe that, since pages are designed for human consumption, the anchors associated with the links to our instance pages usually satisfy this properties: they are expressive, and they univocally identify the instance described in the target page. In our running example, the anchor to a player page usually corresponds to the name of the athlete.

Therefore, FLINT issues a number of queries against a search engine, where each query is composed by the anchor of a link to one of the pages retrieved by the previous INDESIT execution. To focus the search engine toward the right domain, each query is completed with the keyword associated to the entity description: by narrowing the search we avoid false results due to homonyms.

Each search produces a number of results; however, only a fraction of the returned pages are suitable for our purposes. A crucial issue is how to drop out pages that do not represent instances of the target entity. The inclusion of false positives in this step would compromise the whole process, as any error would be propagated in the successive steps. This problem can be avoided by checking its template: only pages whose template contains terms that match with the intensional description of the entity are taken into account.

Each selected page is then given as input to INDESIT, which collects again structurally similar pages from each site. The new anchors found by INDESIT are then used by OUTDESIT to perform new searches on the Web.

The pages retrieved by OUTDESIT are used to populate a custom

search engine. As discussed above, our prototype relies on Google Co-op: the user creates a *facet* with the name of the conceptual entity and associates to it all the pages retrieved by OUTDESIT. According to the search facilities offered by Google Co-op the label can be used to restrict or to refine searches over the indexed pages.

## 2.4 DATAEXTRACTOR

The DATAEXTRACTOR module performs the task of extracting data from the pages retrieved by OUTDESIT. To achieve this goal the system requires that the user labels the values of single-valued attributes of interest on a sample page. Based on a light version of ROADRUNNER [13, 3, 12], a system to automatically infer Web wrappers for data rich pages, the DATAEXTRACTOR generates a wrapper program to extract the labeled values. The inferred wrapper can extract data from every page sharing the same template as the labeled sample page. Since INDESIT retrieves pages from one site according to their structural similarities with a seed page, the inferred wrapper is executed over all the pages obtained by the same INDESIT scan that collected the sample page.

The DATAEXTRACTOR exploits the redundancy of data published in the retrieved pages to generate wrappers for all the pages retrieved by OUTDESIT.

By means of standard record linkage techniques [16], the DATAEXTRACTOR groups the retrieved pages by instance. Then, the values extracted in the initial phase are used to progressively (and automatically) label the remaining pages. The labeled pages are used by the wrapper generation system to create new wrappers. The process is repeated until all the retrieved pages have been processed. This approach is inspired by a machine learning technique introduced by Lerman *et al.* [18] for the issue of wrapper maintenance.

The extracted data are then stored into a semi-structured, self describing data base. Currently we rely on Google Base, but it would be easy to upload the extracted data in other on line services, such as, for example, Freebase<sup>3</sup> or Swivel<sup>4</sup>. In Google Base, the data extracted from each page give rise to a new item, with a suitable title and its set of labeled values.

## 3. RELATED WORK

Our method is inspired to the pioneering DIPRE technique developed by Brin [7]. With respect to DIPRE, which infers patterns that occur locally within single web pages to encode tuples, we infer global access patterns offered by large Web sites containing pages of interest.

Several Web information extraction (IE) techniques have been derived from DIPRE [1, 15, 5]. Compared to our approach these approaches are not able to exploit the information offered by data rich pages. In fact, they concentrate on the extraction of facts: large collections of named-entities (such as, for example, names of scientists, politicians, cities), or simple binary predicates, e.g. *born-in(politician, city)*. Moreover, they are mostly effective with facts that appear in well-phrased sentences, whereas they fail to elaborate data that are implied by Web page layout or mark-up practices, such as those typically published in Web sites containing data rich pages.

Our work is also related to researches on focused web crawling [9, 21], which face the issue of fetching web pages relevant to a given topic. However our goal is different as we attempt to retrieve pages that publish data representing an instance of the entity exemplified by means of an input set of sample pages.

The problem of retrieving documents that are relevant to a user's

<sup>3</sup><http://www.freebase.com>

<sup>4</sup><http://www.swivel.com>

information need is the main objective of the information retrieval field [4, 20]. Although our problem is different in nature, in our method we also exploit state-of-the-art keyword extraction and term weighting results from IR. We observe that the task performed by OUTDESIT might resemble the "similar pages" facility offered by several search engines. However, the semantics of the OUTDESIT searches is radically different, as our method aims at searching for pages similar in the *intensional* description, not in the extensional one.

There are several recent research projects that address issues related to ours. The goal of CIMPLE is to develop a platform to support the information needs of the members of a virtual community [14]. Compared to our method, Cimple requires an expert to provide a set of relevant sources and to design an entity relationship model describing the domain of interest. The MetaQuerier developed by Chang *et al.* has similar objectives to our proposal, as it aims at supporting exploration and integration of databases on the Web [11]. However it concentrates on the deep-web, while we search for pages on the surface-web. A new data integration architecture for Web data is the subject of the PayGo project [19]; the project focuses on the heterogeneity of structured data on the Web: it concentrates on explicit structured sources, such as Google Base and the schema annotations of Google Co-op, while our approach aims at finding data rich pages containing information of interest. Somehow, our approach can be seen as a service for populating the data sources over which PayGo works. Cafarella *et al.* are developing a system to populate a probabilistic database with data extracted from the Web [8]. Data extraction is performed by TextRunner [5], an information extraction system that suffers the same problems discussed above for IE systems, and therefore is not suitable for working on data rich Web pages, which are the target of our searches.

## 4. STATUS OF THE DEMONSTRATION

We have focused our experiments<sup>5</sup> on the sport domain. The motivation of our choice is that it is easy to interpret the published information, and then to evaluate the precision of the results produced by our method. The goal of our experiments was to search for a set of pages, each one containing data about one athlete (player) of a given sportive discipline. We have concentrated on several disciplines, such as basketball, hockey, golf, and so on.

For each experiment we report:

- the input pages;
- the entity description inferred by the system;
- the set of pages retrieved by the system (together with some detail to illustrate how the entity description is used by the system to filter pages returned by a search engine during the discovery process).

The data extracted by the system have been uploaded into a Google Base database, whose url is available from the demonstration Web site, together with the url of a Google Co-op custom, entity aware search engine populated with the retrieved pages.

## 5. ACKNOWLEDGMENTS

This work was supported by the PRIN-2006 Program of the Italian Ministry of Scientific Research. Paolo Papotti was partially supported by an IBM Faculty Award grant.

<sup>5</sup>The experimental results are available at <http://flint.dia.uniroma3.it>.

## 6. REFERENCES

- [1] E. Agichtein and L. Gravano. Snowball: extracting relations from large plain-text collections. In *DL '00: Proceedings of the fifth ACM conference on Digital libraries*, pages 85–94, New York, NY, USA, 2000. ACM Press.
- [2] A. Arasu and H. Garcia-Molina. Extracting structured data from web pages. In *ACM SIGMOD International Conf. on Management of Data (SIGMOD'2003), San Diego, California*, pages 337–348, 2003.
- [3] L. Arlotta, V. Crescenzi, G. Mecca, and P. Merialdo. Automatic annotation of data extracted from large web sites. In *WebDB*, pages 7–12, 2003.
- [4] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.
- [5] M. Banko, M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *IJCAI*, 2007.
- [6] L. Blanco, V. Crescenzi, and P. Merialdo. Efficiently locating collections of web pages to wrap. In *WEBIST*, 2005.
- [7] S. Brin. Extracting patterns and relations from the World Wide Web. In *Proceedings of the First Workshop on the Web and Databases (WebDB'98) (in conjunction with EDBT'98)*, pages 102–108, 1998.
- [8] M. J. Cafarella, O. Etzioni, and D. Suciu. Structured queries over web text. *IEEE Data Eng. Bull.*, 29(4):45–51, 2006.
- [9] S. Chakrabarti, M. van den Berg, and B. Dom. Focused crawling: a new approach to topic-specific Web resource discovery. *Computer Networks (Amsterdam, Netherlands)*, 31(11–16):1623–1640, 1999.
- [10] C. Chang, M. Kayed, M. Girgis, and K. Shaalan. A survey of web information extraction systems. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1411–1428, October 2006.
- [11] K. C.-C. Chang, H. Bin, and Z. Zhen. Toward large scale integration: Building a metaquerier over databases on the web. In *CIDR 2005, Second Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, 2007*, pages 44–66, 2005.
- [12] V. Crescenzi and G. Mecca. Automatic information extraction from large web sites. *Journal of the ACM*, 51(5), September 2004.
- [13] V. Crescenzi, G. Mecca, and P. Merialdo. ROADRUNNER: Towards automatic data extraction from large Web sites. In *International Conf. on Very Large Data Bases (VLDB 2001), Roma, Italy, September 11-14*, pages 109–118, 2001.
- [14] A. Doan, R. Ramakrishnan, F. Chen, P. DeRose, Y. Lee, R. McCann, M. Sayyadian, and W. Shen. Community information management. *IEEE Data Eng. Bull.*, 29(1):64–72, 2006.
- [15] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S.Weld, and A. Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165:91–134, 2005.
- [16] N. Koudas, S. Sarawagi, and D. Srivastava. Record linkage: similarity measures and algorithms. In *SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 802–803, New York, NY, USA, 2006. ACM Press.
- [17] K. Lerman, L. Getoor, S. Minton, and C. Knoblock. Using the structure of web sites for automatic segmentation of tables. In *ACM SIGMOD International Conf. on Management of Data (SIGMOD'2004), Paris, France, 2004*.
- [18] K. Lerman, S. Minton, and C. A. Knoblock. Wrapper maintenance: A machine learning approach. *J. Artif. Intell. Res. (JAIR)*, 18:149–181, 2003.
- [19] J. Madhavan, S. Cohen, X. L. Dong, A. Y. Halevy, S. R. Jeffery, D. Ko, and C. Yu. Web-scale data integration: You can afford to pay as you go. In *CIDR 2007, Third Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 7-10, 2007*, pages 342–350, 2007.
- [20] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. <http://www.informationretrieval.org>.
- [21] S. Sergej, B. Michael, G. Jens, S. Stefan, T. Martin, W. Gerhard, and ZimmerPatrick. The bingo! system for information portal generation and expert web search. In *CIDR 2003, First Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, 2003*, 2003.
- [22] J. Wang and F. Lochovsky. Data-rich section extraction from html pages. In *Proceedings of the 3rd International Conference on Web Information Systems Engineering (WISE 2002), 12-14 December, Singapore*, pages 313–322. IEEE Computer Society, 2002.