Towards Finding Scholarly Articles in Internet Using Hadoop MapReduce with Oozie Workflow

Jakub Jurkiewicz, Aleksander Nowiński
Interdisciplinary Centre for Mathematical and Computational Modelling, University of Warsaw, Warsaw, Poland
{J.Jurkiewicz,A.Nowinski}@icm.edu.pl

An article focuses on the new methods for automatic processing and analysis of the scientific papers. It covers the very first part of this task – discovery and harvesting of scientific publications from the internet. Article is focused on discovery and analysis of the html documents to identify publication resources. Usage of data from Common Crawl project allows operating on large subset of the web pages without a need to perform an expensive crawl of the WWW. We present methods for automatic identification of pages describing scholarly documents in WWW network using html meta headers. Presented set of rules applied to the data achieves reasonable quality.
A system based on these tools is also presented. It allows easy operating and transferring output to the COntent ANalysis SYStem(CoAnSys) - a processing and analysis system developed in ICM. For achieving this goal set of MapReduce tasks running with Hadoop And Ozzie has been used. The quality and efficiency of described rules are discussed. Finally future challenges for our system are presented.

Key words: Hadoop, web mining, scientific content finding, web page classification

Introduction

Information is very precious in the world today. Due to the methods of research evaluation, scientists all around the world produce more and more articles every day. This stream of scholarly communication is a rich resource for research and text mining. Such research is based on vast amount of publications. Interdisciplinary Centre for Mathematical and Computational Modelling of University of Warsaw *(*ICM) does a wide range of such research, in scope of automatic document layout analysis, text mining and semantic network research. As this involves massive processing of all available information, a framework for massive parallel processing, COntent ANalysis SYStem(CoAnSys) has been developed (Dendek et al. (2013)) and is being used. To be able to perform such research in ICM we attempt to collect as many scholarly articles as it is possible. ICM hosts resources of Polish Virtual Library of Science , which is served through Yadda (Zamlynska, Bolikowski & Rosiek (2008)) system developed in ICM. It includes contains publications from following providers:

- Elsevier 7.8 millions fulltext articles
- Springer 1.2 million fulltext articles
- IEEE 2.5 million fulltext articles

Polish bibliographic databases (Agro, Bazhum, Baztech, CEJSH etc.) – roughly 1 million article metadata

As the research is always data hungry, we expand our collection using publicly available articles.

It is easy to find a single publication in internet using existing databases like Scopus, Web of Science or search engines like Google Scholar and Microsoft Academic Search. But aggregating a number of the articles for the purpose of text-mining research is a different task. Databases mentioned before have strict licensing rules and it is not legal to use it for batch processing. Academic search engines on the other hand do not offer API to process result and perform research based on their indexes. Therefore there is a need for scientists interested in analysing scientific publications to build informational resources on their own.

As the biggest collection of data in the world is WWW itself, we have decided to attempt to identify scholarly articles there. Due to limited resources and time necessary to perform crawling of the internet we have decided to use data from Common Crawl project[0]. Common Crawl is a project to perform a complete crawl of the web pages of whole internet, and make result available to the users on Amazon S3 cloud . Access to this data is free, yet general fee for download the data applies. As the Common Crawl data set size is tenths of terabytes, downloading complete data is very expensive. But there is an option to use Amazon cloud to perform initial data filtering and reduce the cost of the transfer. To optimize the cost, we have decided to filter data representing scientific publications, using meta part of html pages to identify scholar article. We have created MapReduce tasks combined by Oozie workflow to filter content of meta part of html pages to obtain only pages describing scientific articles.

Description of the Problem

Web mining is one of the biggest challenges in computer science (Kosala & Blockeel (2000)). While Internet contains a lot of information, people are trying to obtain this information. Different methods have been used in the past two decades. Initially the most popular methods for looking information internet was to use web page catalogue like early Yahoo. Later on web crawlers like Google have emerged. To ease their work, accuracy and precision, meta tags have been introduced. Since beginning their usefulness have been questioned (Turner & Lise (1998)). Apart of different understanding of keywords' meaning other problem problem emerged: unethical search engine optimization, so called web spam (Gyongyi & Garcia-Molina (2005)). Due to keywords spamming search engines tend to ignore keywords in meta tags (Beel & Gipp (2010)). Finally there become common understanding, that there is no value in keywords in meta tags (Ardö (2010)).

The same problem applies to a number of other useful headers of the html web pages: there is significant amount of bias, and it is often hard to identify useful data among it.

But in case of the scholarly publication situation is different: the same section of web page contains usually rich metadata, which describes publication in detail, often including even bibliographical references. This is due the fact, that academic search engines (of which most important is Google Scholar) promote separate set of tags for scholarly publications. Usually there is no interest for the commercial pages to be classified as scholar, so usually only scholar pages should contain these tags. Basing on this assumption, we should be able to make a decision based solely on meta headers section, if a web page is scholar publication or not. In the article we show how we achieve this goal.

Our methodology

Every web page (or nearly every) contains header. This header was meant to describe web page and provide information for machines processing this page. It contains information about author of web page, it's title, some keywords or even short description. Most of the meta headers are composed of predicate (name) and object (content). Example predicate sets important for scholar pages are: dublin core, highwire press tags, and prism. Example of header of nice webpage describing scholar article is presented below:

```
<head>
   <title>Effects of the synergist S,S,S-tributyl phosphorotrithioate
on indoxacarb toxicity and metabolism in the European corn borer,
Ostrinia nubilalis (Hübner) – Pesticide Biochemistry and Physiol-
ogy – Tom 90, Numer 1 (2008) – Biblioteka Nauki – Yadda</title>
   <base href="http://yadda.icm.edu.pl/yadda/" />
   <meta http-equiv="Content-type" content="text/html; charset=
utf-8" />
   <meta http-equiv="Cache" content="no_cache" />
   <link rel="canonical" href="http://yadda.icm.edu.pl/yadda/element/
bwmeta1.element.elsevier-d3daa233-d284-3be1-
9307dd475838e925"/>
   <!-- OpenSearch description document -->
   <link rel="search"
type="application/opensearchdescription+xml"
   href="http://yadda.icm.edu.pl/yadda/api/search/description"
   title="Biblioteka Wirtualna Nauki" />
   <meta name="citation_journal_title" content="Pesticide Bio-
chemistry and Physiology"/>
   <meta name="citation_issn " content="00483575"/>
   <meta name="citation_title" content="Effects of the synergist
S,S,S-tributyl phosphorotrithioate on indoxacarb toxicity and
metabolism in the European corn borer, Ostrinia nubilalis
(H&#252;bner)"/>
   <meta name="citation_author" content=" Analiza  P. Alves"/>
   <meta name="citation_author" content="William J. Allgeier"/>
   <meta name="citation_author" content="Blair D. Siegfried"/>
   <meta name="citation_volume" content="90"/>
   <meta name="citation_issue" content="1"/>
   <meta name="citation_date" content="2008"/>
   <meta name="citation_firstpage" content="26"/>
   <meta name="citation_lastpage" content="30"/>
   <meta name="citation_publisher" content="Elsevier Science"/>
   <meta name="citation_keywords" content="Ostrinia nubilalis,
Indoxacarb, Oxadiazine, DEF, Synergist, Insecticide metabolism"/>
   <!-- Custom META tags -->
   <link rel="stylesheet" type="text/css" href="css/yaddaweb.
css?v=1.12.1-newlayout&rev=35312" />
</head>
```

This header contains a lot of useful information that could be used to create bibliographic record. It includes title of the article its abstract, keywords, authors etc.

Common crawl data is stored in Amazon s3 cloud, and both processing data and downloading includes some fee. To optimize the cost we have decided to filter pages using Amazon Elastic Cloud. As a preliminary filter we have decided to analyse only pages, that have at least one scholar tag set (other than standard meta fields) in the header section. Common crawl data from year 2009 contains information about web pages from 201,641,703 web pages. After filtration described above, done on Amazon cloud, we obtain 2,014,937 web pages headers. This headers were transferred to our own local system.

When we checked this header set we found out, that subset of 40 randomly chosen urls contains 35 pages describing scholar articles. This shows that preselection on amazon cloud was extremely effective. Set of 2,014,937 web pages headers was our working set. To check classification method described below, we randomly choose 58 scholar web pages headers and 19 non-scholar headers. We decided to increase amount of non-scholar webpages in test set, to better check how classifier deals with them. All chosen web pages were from different internet domains.

While selecting pages is classification problem we have to choose some features. We decided to select following features:

- number of documents in domain - domains with less than 3 documents usually does not describe scholar documents,
- using too much keywords - this is typical web spam,
- using the same keyword set for each page in the domain, as it implies, that these keywords describe rather content of the whole service, than single article,

- using highwire tag set in meta section - as meta tags with prefix "citation" are promoted by Google for describing scholar pages, most of the publishers use this tag set, so it is very likely that page using this tag set is scholar page. Hardly any non-scholar page use this tag set

At the initial stage of project we have decided that these features are independent. One module was responsible for checking each feature. A module as a result of the check could assign some positive value representing probability that the web page is scholar, and/or another value expressing probability that that the same web page is not scholar. It could also assign no value at all and abstain from the voting.

After running all modules for a single web page, we used the equation for success in set of independent events, and as a result we get two numbers: probability that webpage represents scholar publication and probability that the web page does not represent it. It is also important, that we deliberately assumed that for some webpages decision is not clear, even for human researcher.

Finally we choose webpages with probability for being scholar higher than 50% and decide that all other are not scholar.

Used tools

The set of the data used in this research is not extremely large, but it represents less than 5% of totally available data in Common Crawl. Therefore to achieve reasonable performance we have decided that we need a good scalability. Therefore we have decided to use Apache Hadoop framework to perform computations. Apache Hadoop is an open java implementation of the framework for computation using MapReduce paradigm in massively parallel environment. Apache Hadoop has been used both in the filtering stage while running processing on Amazon Elastic Cloud, and later on in the scientific document identification phase.

The MapReduce paradigm has been brought from functional languages to large scale computing by Google (Dean & Ghemawat (2004)). It allows extremely scalable processing on large computer clusters. The computation is split into two steps: map() and reduce(). In map() phase for every input element some key-value pairs are emitted. In reduce() phase pairs sharing the same key are processed together producing final result. This final result is also emitted as key/value pairs. Such definition of the job allows to run multiple parallel processes both for map and reduce steps, as no information has to be exchanged between processes during step. In page filtering job during map() step each page has been analysed, and if it contained proper set of tags it has been emitted, using URL as a key. Reduce step was simple identity operation, and results were filtered out pages. The second task - detailed identification was opposite: in map() phase records has been mapped using domain names as keys, and all other steps have been done in reduce() phase.

The big advantage of using Hadoop is scalability, which allows processing huge amount of data parallely. While for test set it is not so important, in future we plan run bigger collections of data when time would be crucial.

Unfortunately Apache Hadoop is not very user friendly, composing tasks and workflows of multiple map-reduce step requires a lot of effort. To simplify this task we have used Oozie workflow management tool. Oozie allows declaring job in XML format, much easier to manage than traditional Hadoop jobs. It allowed fast re-running workflow after making changes, and fixing bugs in our software. It also allows ease of parameter configuration.

Results

All modules assigning features to web pages, were MapReduce jobs. These jobs were composed into Oozie workflow. Our testbed was hadoop cluster containing 4 nodes and master node. Each worker node has four AMD Opteron 6174 processors (48 cores in total), 192 GB of RAM, four 600 GB disks connected in RAID 5 array with an access to 7TB LUN of NetApp disk storage over FC. The master has 8-core CPU, 32 GB of RAM and 64 GB storage. Filtering job of 2,014,937 web page headers took around 15 minutes.

Below is presented table with result of classification of 77 web pages test set.

One incorrectly classified web page was due number of repeating keywords sets in their domain. We can work it out however this would lead to overtraining.

This result shows 100% of precision, and 98,7% of accuracy, which is very good result.

Conclusions and Future work

We have achieved surprisingly good results, on the data from the test set. It shows that in some cases really simple methods can give very good results. A big advantage of our method is generation of classifiers in human understandable

Table 1. Results of classification of the web pages from the manually created test set

Number of web pages that:	should be classified as scholar	should be classified as not scholar
were classified as scholar	57	0
were classified as not scholar	1	19

form. Chosen classification method gave us good results and were very easy for debugging and searching for errors.

Another conclusion could said something about evolution of web pages in Internet. Normal metatags are usually ignored by search engines. However the most of bad SEO (search engine optimization) is not targeted to Google Scholar. Thanks to that quality of tags used for describing scholar pages is relatively high, and they are not for other pages - like highwire citation meta tags. On the other hand we find out that dublin core metatags are used for all kinds of web pages, and one cannot trust on them.

In future we plan to test our methods on bigger test set. We plan to use larger set of urls for testing. We also plan to gather newer data sets from CommonCrawl and test how much scholar web pages were removed by preliminary selection.

Bibliography

[1] Dendek, P. J., Czeczko, A., Fedoryszak, M., Kawa, A., Wendykier, P. & Bolikowski, L. (2013). Taming the zoo - about algorithms implementation in the ecosystem of Apache Hadoop, 12. Information Retrieval; Digital Libraries. Retrieved from http://arxiv.org/abs/1303.5367

[2] Zamlynska, K., Bolikowski, L. & Rosiek, T. (2008). Migration of the Mathematical Collection of Polish Virtual Library of Science to the YADDA platform. Towards Digital Mathematics Library. Birmingham, United Kingdom, July 27th, 2008, 127-130.

[3] Kosala, R. & Blockeel, H. (2000). Web mining research. ACM SIGKDD Explorations Newsletter, 2(1), 1–15. doi:10.1145/360402.360406

[4] Turner, T. P. & Lise, B. (1998). Rising to the Top: Evaluating the Use of the HTML META Tag to Improve Retrieval of World Wide Web Documents through Internet Search Engines - Library Resources & Technical Services - Volume 42, Number 4 / 1998 - American Library Association. Library Resources & Technical Services, v42 n4 Oct 1998. Retrieved July 5, 2013, from http://alcts.metapress.com/content/gq8151m1l 8515845/

[5] Gyongyi, Z. & Garcia-Molina, H. (2005, April 1). Web Spam Taxonomy. First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb 2005). Retrieved from http://ilpubs.stanford.edu:8090/771/1/2005-9.pdf

[6] Beel, J. & Gipp, B. (2010). On the Robustness of Google Scholar Against Spam. Proceedings of the 21st ACM conference on Hypertext and hypermedia - HT '10 (p. 297). New York, New York, USA: ACM Press. doi:10.1145/1810617.1810683

[7] Ardö, A. (2010). Can We Trust Web Page Metadata? Journal of Library Metadata, 10(1), 58–74. doi:10.1080/19386380903547008

[8] Dean, J. & Ghemawat, S. (2004). MapReduce: Simplified Data Processing on Large Clusters. Communications of the ACM, 51(1), 1–13. doi:10.1145/1327452.1327492