Extracting Information from Web Content and Structure

Dalibor Fiala[*+]

Roman Tesař[*]

Karel Ježek[*]

{dalfia,romant,jezek_ka}@kiv.zcu.cz

François Rousselot[+]

francois.rousselot@insa-strasbourg.fr

Abstract: Web is a vast data repository. By mining from this data efficiently, we can gain valuable knowledge. Unfortunately, in addition to useful content there are also many Web documents considered harmful (e.g. pornography, terrorism, illegal drugs). Web mining that includes three main areas – content, structure, and usage mining – may help us detect and eliminate these sites. In this paper, we concentrate on applications of Web content and Web structure mining. First, we introduce a system for detection of pornographic textual Web pages. We discuss its classification methods and depict its architecture. Second, we present analysis of relations among Czech academic computer science Web sites. We give an overview of ranking algorithms and determine importance of the sites we analyzed.

Key Words: Web mining, information retrieval, classification, ranking algorithms

1 Introduction

Internet is an immense resource of data. There are billions of documents in various formats – text, image, audio, video, etc. Many of these documents represent useful knowledge that must be extracted out of them first. This extraction (mining) is the subject of a scientific field called Web mining. Recent advances in Web mining have concentrated on content, structure, and usage mining. Both content and structure mining techniques may help us distinguish between relevant and irrelevant Web documents. The first by categorizing into on-topic and off-topic documents, the latter by determining important (authoritative) documents via analysis of their relations to other documents. Of course, the heterogeneous and decentralized nature of the Web causes many useless, harmful or even criminal Web pages to appear. Web mining may also be applied to their detection and elimination. In chapter 2, we are concerned with Web content mining and we introduce a system for filtering textual pornographic Web pages. In chapter 4, we deal with Web structure mining and we present our analysis of Czech academic computer science Web sites.

2 Web Content Mining

2.1 Indecent Web Pages

By far the biggest problem of the Internet is, without any doubts, the freely accessible and always expanding pornographic content. Unlike other topics defined by the Penal Code of the Czech Republic (no. 140/1961 art. 205 (immoral offence), 260 (suppression of rights and

[*] Dept. of Computer Science and Engineering, University of West Bohemia, Univerzitni 22, CZ-30614 Plzen

[+] INSA Strasbourg, 24 bd de la Victoire, F-67084, Strasbourg Cedex freedom of man), 261, 259 (genocide) and 261a), the recognition of pornographic textual content in terms of indecency is not difficult. It is much more complicated with suppression of rights or with genocide. When visiting a Web page on these topics, it is often not very clear whether the page represents a violation of law or whether it is just a pure description of the domain without any hidden intentions. We often need a domain expert for an exact evaluation. Of course, there is a difficulty with accessibility of such pages as well. It is not only complicated to recognize them but also to find them. They are by far not as frequent as pornographic sites. It is a tedious task. However, a possible solution was outlined in [3].

Therefore, at the first stage of our project we created a collection of pornographic textual documents (by means of on-topic hub pages) definitely violating the law according to the immoral offence article. Further below, we will use the term indecent instead of pornographic.

2.2 Comparison of Classifiers

Text classifiers are generally very sensitive to training data. For the training phase we always need positive (indecent) and negative (decent) data if we categorize into two classes, which is our case. So, we have implemented a few well-known and frequently used classification methods and we have modified their functionality for our purposes.

In Table 1 we show the properties of collections used. For testing, we utilized decent documents from standard classification sets. Indecent data was represented by collections assembled from Internet resources (see Section 2.1).

	# documents				
	indecent	decent	# distinct words	min # words in document	max # words in document
Reuters & porn sites	400	400	22 501	76	10 633
7 Sectors[+]	0	4 581	36 653	18	7 615
20 Newsgroups[+]	0	16 330	104 785	45	22 568
WebKB[+]	0	8 273	63 675	31	30 853
Internet porn sites	18 323	0	75 042	186	14 226

Table 1: Features of testing collections

As was shown in [9], applying K-itemsets for classification (for $K > 1$, where $K = 1$ means just individual words) does not result in a significant improvement of classification accuracy. Thus, we have decided to try out word phrases (also referred to as sequences or n-grams) in addition to this approach. To obtain them from the texts, we made use of the suffix tree structure (see [8]), which is less computationally expensive than the well-known Apriori algorithm [4] destined for finding K-itemsets.

We were performing classifications with the phrases obtained (for details of phrase-based classification, see [9]) and we compared the results not only with the Itemsets technique [4] but also with other common classifiers. We set the maximum length of phrases to three as longer phrases have only some insignificant influence on classification due to their rare occurrence. The number of the phrases we employed in the classification was selected as 700 for each length (1, 2 and 3), based on the chart in Figure 1. The accuracy of classification is

[+] Available from http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb/index.html represented by the break-even-point (BEP), i.e. the value in which recall and precision are the same, which is a frequent approach.

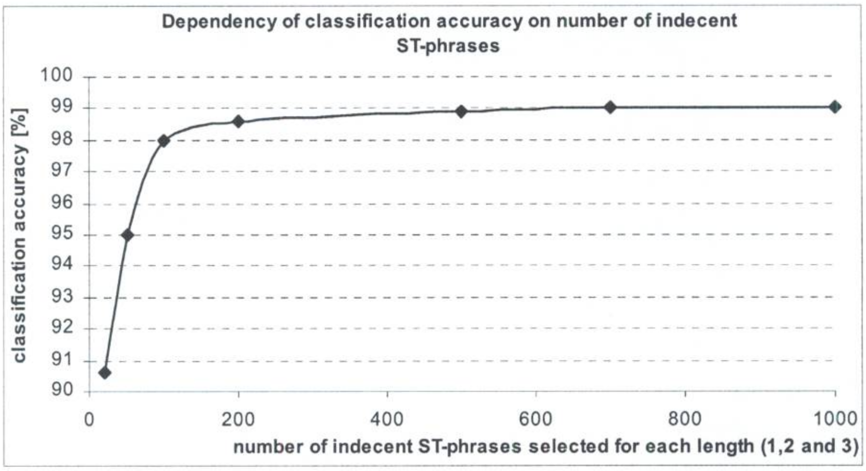

Figure 1: Dependency of classification accuracy

Since individual classifiers require positive as well as negative training data, we took all indecent documents from all collections in Table 1 as indecent training samples and all decent documents from all collections as decent training objects. In the preprocessing stage, we applied stemming to those documents [7] and we removed all numbers and stop words. Table 2 summarizes the results of particular classifiers for each data collection separately.

	Itemsets	Naive Bayes	TFIDF	C4.5	ST-phrases
Reuters & porn sites	99.75	97.88	99.25	99.38	**100.00**
7 Sectors	99.39	94.56	88.73	99.19	**99.50**
20 Newsgroups	99.82	**99.96**	90.92	97.89	99.31
WebKB	**99.14**	97.04	92.28	**99.14**	98.97
Internet porn sites	97.23	96.88	**98.19**	95.80	97.80

Table 2: Accuracies of classification of test collections from Table 1 [%]

As we may see from the results in Table 2, the classifiers have no problems with categorizing into two very distinct classes. However, application of suffix tree phrases of lengths 1, 2, and 3 yields very good outputs compared to the Itemsets method and it results in an improvement of the overall classification accuracies (column *ST-phrases*), which confirms the conclusions taken in [9].

Classification based on suffix tree phrases is very fast, similarly to the classification phase of the Itemsets method. It merely consists in searching for learned (indecent) phrases in the document being classified, which makes it usable for real-time work. See [9] for a more detailed description of this classification method.

The next step was the integration of our phrase-based classifier within a system for filtering Internet content and detecting indecent Web pages.

2.3 Web Filter

We have decided to realize our application of filtering Internet content as an HTTP proxy server. Initially, we were trying to implement a proxy server of our own. But for the sake of adhering to general communication protocols, a backward compatibility and the overall complexity of the resulting application, we have preferred adopting an existing solution. At last, we chose the MUFFIN proxy server [12], which had a number of already implemented features. An overview schema of our system for filtering HTML pages is depicted in Figure 1.

In the block diagram in Figure 1 there is a Porter's stemmer module [11] whose aim is to extract word stems and thus enhance classification. Usage of this module is still an open issue because some experiments show that word stemming does not always contribute to classification accuracy.

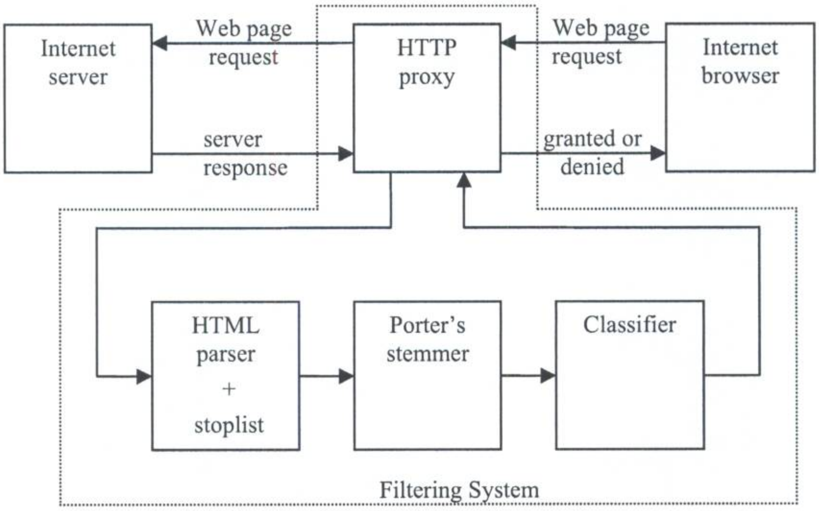

Figure 1: Basic system architecture

The principle activities of the system are very simple. After launching the HTTP proxy server, network traffic is monitored at a selected port. When a Web browser connects to the proxy server at this port and requests a Web page, the IP address of the server hosting this page is find out first and, subsequently, it is looked up in the list of forbidden hosts that may be defined in a configuration file. Access to a server found in this list is denied immediately. Likewise, a list of permitted hosts is checked so as to grant access without any further filtering. If the address has not been found in these lists, the request is sent to the Web server. Its response is captured by the proxy server and dispatched to the HTML parser [10]. The parser removes HTML tags and transmits the resulting text obtained from the Web page to the stemmer which reduces individual words to their stems. The text modified in this way is classified by the suffix tree classifier which decides whether or not the page is indecent and, accordingly, it allows or refuses it to be displayed in the Web browser. In case that the Web page is classified as indecent, it is stored for further analysis.

The application is implemented in Java and thus it is able to work under various operating systems. Furthermore, its modular architecture allows for an easy use of its particular components in other systems and, at the same time, for future extensions of its functionality. At present, we conduct experiments that should verify the system's capability of processing a large number of Web page requests in real time.

3 Web Structure Mining

Recently, it has become clear that studying the structure of the Web helps extract much useful information. The knowledge of structure is sometimes even more valuable than the knowledge of content. Since the Web is mostly modeled as a graph, the importance of its structure (or topology) is indisputable. A common task, performed by Web search engines among others, is to determine importance of a Web site or Web page. This is often done by exploring its relations to other Web pages in terms of hyperlinks among pages in analogy to counting bibliographic citations in scientific literature. There are a couple of common ranking algorithms that assign quality rankings to Web pages, based on the structure of the Web graph. We will briefly introduce three main ranking algorithms, which requires some necessary formalization.

3.1 Ranking Techniques

Let $G = (V, E)$ be a directed, edge-weighted graph, V a set of nodes, and E a set of edges. G is a Web graph representing a portion of the Web, V a set of Web pages (or Web sites; it depends on the level of abstraction), and E a set links among these pages. Of course, we can set all edge weights to one when necessary.

An intuitive, first-order ranking method is to count in-links for each node. Analogically to bibliographic citations, the node with the largest number of in-links is considered as the greatest authority. If all edge weights in the graph are equal to one, the number of in-linking edges of a node is the same as its in-degree, which is computed as follows:

$$D_{in}(u) = \sum\nolimits_{(v,u)\in E} w(v,u) \tag{1}$$

where u and v are nodes in the graph G and $w(v, u)$ is the weight of the edge linking from node u to v. Note that only direct neighbours can influence each other. If node A cites node B and node B cites node C, the rank of C is the same as if there was only node B citing node C. The citations always have the same impact. It is in contrast with the traditional notion of prestige where some citations have a larger importance than the others. The next two algorithms are recursive in nature and reflect more complex relations among nodes.

PageRank[*] [1, 6] is computed like this:

$$PR(u) = \frac{1-d}{|V|} + d\sum\nolimits_{(v,u)\in E} \frac{PR(v)}{D_{out}(v)} \tag{2}$$

where d is a constant usually set between 0.8 and 0.9 and D_{out} is an out-degree calculated analogically to (1).

HITS [5] defines authority and hubness for each node:

$$A(u) = \sum\nolimits_{(v,u)\in E} H(v)$$

$$H(u) = \sum\nolimits_{(u,v)\in E} A(v) \tag{3}$$

The principal idea is that a good authority is pointed to by many good hubs and a good hub points to many good authorities.

We have no space to explain the higher-order ranking methods in detail and refer to the above publications. The recursion has such effect that, for instance, page A makes bigger the importance of page B, which contributes to the significance of page C, which improves the

[*] PageRank is used in the Google Web search engine for evaluation of pages.

authority of page A again. Thus, a citation from a popular Web page has a larger impact than from an insignificant one. In practice, we solve those recursive equations iteratively. Usually, a few dozens of iterations are enough for the ranking of nodes ordered by importance to stabilize. Interestingly, recent studies [2] have shown that the results of all three ranking methods are highly positively correlated, so we can content ourselves with counting in-degrees in common cases.

3.2 Experiments

Our task was to determine authoritative institutions within a set of Czech academic computer science Web sites. We chose this area because we knew it well and we could expect that there would be enough data on the Web to analyze. However, the methodology we have developed is sufficiently general and it can be applied to a different domain and scope. In a Web directory we selected 17 computer science departments at last (see Table 3). Unfortunately, what makes the selection a little bit complicated is the fact that not all Czech Universities have the same structure and the same hierarchy of faculties and departments. Therefore, there are also faculties that have a similar profile like our home department.

Site	Institution
cs.felk.cvut.cz	ČVUT Praha
iti.mff.cuni.cz	UK Praha
kam.mff.cuni.cz	UK Praha
ki.ujep.cz	UJEP Ústí n. L.
kit.vse.cz	VŠE Praha
kocour.ms.mff.cuni.cz	UK Praha
ksvi.mff.cuni.cz	UK Praha
ktiml.ms.mff.cuni.cz	UK Praha
ufal.mff.cuni.cz	UK Praha
www.cs.vsb.cz	VŠB-TU Ostrava
www.fi.muni.cz	MU Brno
www.fit.vutbr.cz	VUT Brno
www.inf.upol.cz	UP Olomouc
www.kai.vslib.cz	TU Liberec
www.kin.vslib.cz	TU Liberec
www.kiv.zcu.cz	ZČU Plzeň
www.cs.cas.cz	AV ČR

Table 3: Web sites involved in our experiments

3.3 Results

One condition that limited our selection of experimental Web sites was that each department should have its home page on a Web server of its own and not of its faculty or University. Separate servers can be more easily processed by automated Web agents because the robot immediately recognizes whether or not a link is internal (within the server) or external. Some well-known heuristics also say that longer URLs are less important documents than the shorter ones. So, from this point of view, we left out less significant sites right from the start. In December 2005, we let the Web robot crawl all of the seventeen Web sites and store all appropriate information to a database. We repeated the experiment two more times in several days and the results (see Table 4) were almost the same. We considered only documents in certain formats that were accessible via HTTP protocol. For instance, we ignored audio and video files, which is natural, but also doc, ppt, rtf, and txt documents, which may be arguable. (Omitting these formats in one of the experiments caused only one change in the middle part of the chart in Table 4.) To prevent the Web agent from getting stuck in Web traps, we decided the maximum height of the document tree to be eight, which is a good estimate by experience. (Documents in greater heights are usually duplicates with different URLs.)

# Docs	# Citations	Ratio
15 438	926	0.0600
632	335	0.5301
18 325	243	0.0133
10 952	76	0.0069
12 309	69	0.0056
16 422	56	0.0034
11 860	43	0.0036
3 226	38	0.0118
148 682	29	0.0002
46	18	0.3913
1 230	13	0.0106
472	13	0.0275
847	3	0.0035
8 316	2	0.0002
2 423	0	0.0000
273	0	0.0000
240	0	0.0000
251 693	1 864	0.0074

Table 4: Authoritative Web sites

In total, the agent gathered over 250 thousand documents of requested types and it created a corpus of about 7 GB. In Table 4, we can see that the Web sites (or departments if we identify institutions with their Web sites) are split into three clusters. At the top, there are three sites distinctly ahead of the others. At the bottom, there are departments that have very few or even no citations. Citations are links from documents on other servers. Also, we removed duplicate citations or self-citations. As quality ranking is somewhat tricky we do not disclose the relationship between Table 3 and Table 4 at present.

3.4 Difficulties

The numbers of citations could be replaced with the ranks obtained from the PageRank algorithm or with authorities gained from HITS. Respecting the correlation mentioned above, however, the ordering of the hosts should be similar. There are some facts, though, that might have a much larger impact. For instance, existence of server aliases is annoying. If there are two host names representing one machine with the same content, references to them should be counted together. There may be a large number of aliases and ignoring them may yield significantly distinct results. Another trouble is dynamically generated Web pages (see the server with a much larger number of documents in Table 4). Two and more URLs (and thus two and more possible citations) may represent one document and citations by them should be counted only once then. This is painful, especially with regard to the low inter-linkage among the sites (0.0074). There is also a difficulty with document formats. If the ignored documents (e.g. rtf) were more frequent on one server than on the others, this host would be disfavoured. Therefore, we must take into account all these possible effects before declaring the most authoritative sites.

4 Conclusions and Future Work

Web is a huge data store that must be further analyzed to transform the data into knowledge. Due to its scope, heterogeneity and dynamic development, much research effort has been recently devoted to finding tools and techniques for exploiting its content, structure and usage. Moreover, some Web documents are not only useless or irrelevant but even harmful. We were concerned with two applications of Web content and Web structure mining. First, we presented a system for filtering pornographic textual Web pages that may be easily extended to include other indecent content. And second, we introduced some Web page ranking methods and we showed their direct application to determine authoritative Web sites from the Web structure. Although concentrated on Czech computer science Web hosts, we hope to apply the same methodology in other areas as well.

References

1. Chakrabarti S.: *Mining the Web: Analysis of Hypertext and Semi Structured Data*, Morgan Kaufmann, 2002.
2. Ding C., He X., Husbands P., Zha H., Simon H.: *PageRank, HITS and a Unified Framework for Link Analysis*, Proc. 25th ACMSIGIR Conference on Research and Development in Information Retrieval, Tampere, Finland, pp. 353 – 354, 2002.
3. Greevy E., Smeaton A. F.: *Classifying racist texts using a support vector machine*, Proc. 27th ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, United Kingdom, pp. 468 – 469, 2004.
4. Hynek J., Jezek K., Rohlik O.: *Short Document Categorization - Itemsets Method*, PKDD 4th European Conference on Principles and Practice of Knowledge Discovery in Databases, Workshop Machine Learning and Textual Information Access, France, pp.14 - 19, 2000.
5. Kleinberg J.: *Authoritative sources in a hyperlinked environment*, Proc. ACM-SIAM Symposium on Discrete Algorithms, 1998.
6. Page L., Brin S., Motwani R., Winograd T.: *The PageRank Citation Ranking: Brining Order to the Web*, Tech. report, Computer Science Department, Stanford University, 1998.
7. Porter, M. F.: *An algorithm for suffix stripping*, Program, 14(3): pp. 130 - 137, 1980.
8. Tesar R., Fiala D., Rousselot F., Jezek K.: *A comparison of two algorithms for discovering repeated word sequences*, WIT Transactions on Information and Communication Technologies, Vol. 35, pp. 121 - 131, 2005.
9. Tesar R., Jezek K.: *Klasifikace Sufix Tree frázemi - srovnání s metodou Itemsets*, Proc. ZNALOSTI 2005, Slovensko, pp. 144 - 153, 2005.
10. http://htmlparser.sourceforge.net/
11. http://www.tartarus.org/~martin/PorterStemmer/index-old.html
12. http://muffin.doit.org/