Identifying the subject of small, sparsely linked collections from a web community

Deepak P. and Jyothi John

Model Engineering College, Kochi, Kerala, India
E-mail: deepak-p@eth.net E-mail: jyothijohn@mec.ac.in

Abstract: This work deals with the problem of identifying the subject of small, sparsely linked collections of web documents from a web community. In the course of attempts to find solutions for many problems concerning the web, we are often left with a handful of pages dealing with something in common, but with very few links within them. This paper presents algorithms which work on such collections and output a set of words ordered in the decreasing order of relevance. The most relevant words provide a close approximation of the topic that the collection deals with.

Keywords: subject; identification; web communities; sparsely linked.

Reference to this paper should be made as follows: P., Deepak and John, J. (2004) 'Identifying the subject of small, sparsely linked collections from a web community', *Int. J. Web Based Communities*, Vol. 1, No. 1, pp.35–45.

Biographical notes: Deepak P. is currently studying for BTech (Computer Science and Engineering) at Model Engineering College, Kochi, India. He is extremely interested in software engineering, computational complexity and web technologies. He aspires to be a researcher in computer science.

Jyothi John is currently a Professor in Computer Science and Engineering at Model Engineering College, Kochi. He graduated from the TKM College of Engineering, Kerala University and got his postgraduate degree in Computer Science and Engineering from the Indian Institute of Technology, Mumbai, India. He worked as a Lecturer at the TKM College of Engineering from 1981 to 1993 after which, he joined Model Engineering College as an Assistant Professor in Computer Science and Engineering. Since 1997, he has been working as a Professor at Model Engineering College. His research interests include cluster computing and information retrieval.

1 Introduction

This work presents algorithms, which, when supplied with 'small sparsely linked web collections' from a web community (or having something in common), return a list of words which try to approximate the common subject that the collection deals with. Sparsely linked web collections which are encountered often, contain tens to a few hundreds of documents with the number of links within them ranging anywhere from a couple to a dozen. The collection of relevant pages from search results, the collection of pages browsed by a child (for parental monitoring systems), etc. fall in the category of such collections. Algorithms that deal with web communities often use both link-based

and text-based information. Hypertext, the language of the web, is a collection of text and hyperlinks. Thus, usage of information from both sources in the same proportion may well be justifiable in many cases, but certainly not when all that we get is a 'small sparsely linked web collection' where the link-based information may be too little to use. The information relies too much on something that could well be catastrophic if the links are accidental. The algorithms presented here use link-based information, but only to supplement the text-based information which is given far more weightage. The set of words returned by the algorithm for a given collection may be called the 'subject set'. Unfortunately, not much literature was found on the problem dealt with by the paper.

We recognise the possibility of doubting the relevance of such a paper and as to whether sparsely linked collections are so relevant to be dealt with separately. A study [1] confirms that the fraction of pages with in-degree or out-degree i is proportional to $1/i^x$ for $x = 2.1$ for in-degree and $x = 2.72$ for out-degree. Thus, the number of pages decreases exponentially with the increase in degree. So, the probability of having a sizeable amount of links within a small collection is very low. Hence, we have to deal with sparsely linked collections very often, and that is exactly why this paper proposes dealing with such collections separately.

The rest of the paper is organised as follows: Section 2 explores the applications of such techniques; Section 3 lists down the possible ingredients of algorithms for the problem at hand; and Section 4 lists the algorithms proposed and their descriptions. The next section describes the test setup along with short descriptions of the collections on which the algorithms were tested. Section 6 lists the test results followed by conclusions in Section 7, references in Section 8.

2 Applications of such techniques

2.1 To refine web searching

Currently, web search engines in response to a query generate a list of web pages with automatically generated descriptions for each page. It is a common experience of users that 'good' search results often occur only in the first few result pages. The first 20–30 results often can be best described as a small sparsely linked web collection. A search for an event usually lists different news reports on the event and that for a name usually lists pages on different real world people having that name. Similarly, the search for a field of study displays links to research papers on it and departments which teach it. Thus, more often that not, we have pages from different sites which may even be authored by people from geographically very distant locations, who otherwise have no reason to link to each other. A search for a product may list pages of different vendors who would definitely not risk placing links to competitors on their pages. Having conjectured that search results often present us with small sparsely linked web collections, we can explore methods of using the subject set of such collections to enhance web searching.

The subject set that the algorithm returns when presented with documents obtained from a query on a search engine can be expected to contain the terms in the search query itself, as the most highly ranked terms. Search queries are often found to contain not more than a couple of words on the average. The words in the subject set, apart from those in the search query, can provide an indication of topics closely related to the search query topic. Thus, listing them could provide valuable clues for the user to refine his

search. For example, a search for Kumarakom, a tourist destination in Kerala, India would inevitably contain pages relating to Kerala tourism in the results. The subject set of the collection from the results was found to have tourism and boating as among the highly ranked words. Displaying these words would enable the user to refine his search by incorporating tourism and/or boating to his search query. The former would benefit a user inquiring about Kumarakom or finding tourist destinations and the latter would benefit someone searching for tourist attractions in Kumarakom.

2.2 Reducing the web search process to just a lookup

A search engine almost always must have an updated copy of the whole web graph [2]. There are proven algorithms for detecting communities in the web graph, as by finding a dense bipartite graph [3], a k-clique, NK-clan [4] or a structure where nodes have more links to pages within the structure than to outside. The search engine could identify the communities from the web graph by such techniques and associate each community with its subject set. When a search query comes in, it can be matched with the subject set of communities and the pages that belong to the communities whose subject sets contain the issued query could be the search results (after applying ranking algorithms for appropriate ordering of results) [5]. Although this would seem to be a time-consuming process, caching and other such optimisations could be used to speed up this matching process. The accuracy of such a plain lookup technique has to be subjected to further investigations. Further, it is a common observation that web communities have a lot of links within them. So the algorithm presented in this paper (which is directed towards dealing with sparsely linked collections) might have to be adapted to enable it to use or give more weightage to link-based information before usage in such a scenario.

2.3 Parental monitoring systems

Parental monitoring systems typically look to identify the amount of obscenity in a collection of web pages that the child (or more generally, the person under observation) has browsed. A common child tendency is to search using search queries like, ‘free videos’ and ‘hot pics’ and browse the pages returned. As can be inferred, the collection of pages returned for such searches would have ‘porn’ or ‘sex’ in the subject set.

Further, many porn-related sites popup windows of similar sites, in which case, the pages would be diverse (sparsely linked), but dealing with a common topic. In both cases described above, the amount of obscenity among terms in the subject set could well serve as an indicator of the intentions of the person under observation.

2.4 Other possible applications

In this era of proliferation of web services, useful web services can be planned based on subject set information. Two words appearing in the subject sets of the same community is a good indication that they are closely related. e.g., Communism and Marx might appear together in the subject sets of many communities. Interesting clues to the web identity of a person can be found by examining the subject set of pages related to the person.

3 Possible ingredients of such algorithms

Algorithms for subject identification have to deal with many issues. As the algorithm outputs a set of words (the subject set), possibly in the order of preference, such algorithms would naturally have to do a lot of text analysis. Thus, having conjectured that text analysis would be the integral part of this algorithm, we have to identify how to do it. We can possibly choose to deal with hyperlink anchor text as different from usual text.

3.1 Hyperlink anchor text analysis

Hyperlinks are considered to convey better information, both in terms of quality and accuracy, than web page text. It is also believed that it is less practical to forge hyperlink information. Further, it has been found that there are usually 2.89 words per anchor [6] and thus anchor text information may convey a lot. Even in sparsely linked web collections, we can expect to find some links between pages. Considering a sparsely linked collection of 20 documents, we can well expect to see around a couple of hundred or more. Out of them, only less than ten could be linking to pages within the set. Thus, it is an important decision to make whether to use the information from the links which lead to pages that are not in the collection. Advertiser links usually fall in that category, but there may be many more useful links as well. Manual inspections of such links led us to the conclusion that considering such links would be more harmful than useful, the algorithms presented here do not use information from such links. Even though the possibility of 'noisy' links exist, the algorithms presented here use the information from links that have the target within the collection at hand.

3.2 Plain text analysis

As the anchors themselves make up only a fraction of the text in web documents, an algorithm that aims to output a list of words would inevitably have to analyse non-anchor text. As per the conjecture that anchor texts of hyperlinks contain information about the target pages, it may well be inferred that good hubs point to a lot of good authorities [7]. Listing good authorities highly is the aim of any search engine and many text based search engines do have good authorities high up in the list, although they do not use any linkage information. Thus, good authorities may also have information about the subject embedded on them. So, the weightage that has to be given to the text of a page could well be made to depend upon that number of inward links to that page and outward links from the page. But, as the links in the collections we deal with are very rare, we should be careful not to rely on the links from and to a page to evaluate it beforehand. About 50% of the links in the collection being 'noisy' should not spoil the entire subject set.

3.3 Markup language based-techniques

HTML is (or was, until recently) the language of the web. HTML contains a lot of structures called tags which commonly begin with <tag-name options> and end with </tag-name>. Certain tags like title and meta can be used to derive a lot of semantic information. Examples of such heuristics may include techniques such as giving higher weightage to table headings, huge bold texts or giving low weightage to texts with small

fonts at the bottom of a page (which is very likely to be an advertisement text). But, such techniques have their own disadvantages in that they are based on 'markup language. They may quickly become obsolete and useless with new languages like XML (with customisable tags) gaining in popularity.

4 Algorithms and their descriptions

4.1 Algorithm A

This algorithm is centred on page text analysis, more specifically on the heuristic that the page text contains the subject. It is based on the discussions in Section 3.2. It has three variables, which can be assigned different values to obtain different flavours.

4.1.1 Algorithm

Table 1 Algorithm A

For each page, page$_i$
{
*Score of page$_i$ =(***a*** *number of links targeting page$_i$) + (***b****number of hyperlinks from page$_i$) +* **c**
}
Score of a word= Σ *(frequency of the word in page$_i$)*(score of page$_i$)*

Before pages are subjected to the algorithm, they have to be stripped off the HTML tags and other syntactical content. The values of a, b and c determine the relative weightage given to authorities, hubs and linkage information respectively. An instance of the algorithm used for the tests is presented below. It uses c = 1. With c = 1, isolated pages or pages not linking with any other page in the set are given some weightage. Page text analysis, unlike anchor text analysis, is a time-consuming process. This algorithm does a single pass on the whole text of each page.

Table 2 Algorithm A1

For each page, page$_i$
{
Score of page$_i$ = (number of hyperlinks from page$_i$)+1
}
Score of a word= Σ *(frequency of the word in page$_i$)*(score of page$_i$)*

The set of values used in this case is a = 0, b = 1 and c = 1.

4.1.2 Issues related to the algorithm

Page text analysis is a very exhaustive process. But, this algorithm does rely on it due to the need for the result as a set of words. If the set of pages used is not a heavily linked one, there is bound to be a lot of time difference between setting c = 0 and otherwise. Although there is no proven information about the average size of web pages on the web,

it can be assumed to be around 20K. Thus, even a set of 100 pages would have close to 2M data, detailed analysis of which is bound to be costly. The main disadvantage of this algorithm is its speed (or lack of it), although caching techniques can be used to speed it up if implemented on a large scale. Since small sparsely linked web collections do not provide too much link-based information for explotation, we resort to a full pass over the page texts.

Another major disadvantage of this algorithm is that there is no provision for limiting the influence of a page. Thus, the score of a word may be boosted by a single page (if the word frequently occurs in it) to a value much beyond the reach of other words. This drawback may cause this algorithm to be fooled or misled by forgers who place a lot of invisible text to get a higher rank in search listings and other web services.

4.2 Algorithm B

This is a variant of algorithm A and like algorithm A is based on page text analysis. This algorithm differs from algorithm A in that it limits the influence that can be exerted by a single page. As opposed to algorithm A, it has five variables that can be adjusted according to needs.

4.2.1 Algorithm

Table 3 Algorithm B

Global score of every word=0;
For each page, page$_i$
{
*Score of page$_i$ =(**a*** *number of links targeting page$_i$) + (**b***number of hyperlinks from page$_i$) + c;*
*Local score of a word= frequency of the word in the page$_i$ * score of the page$_i$;*
Pick d words having the highest scores and let their scores be {s[0],s[1]…s[d-1]};
*Scale their scores so that the sum of their scores becomes equal to page-score*e;*
Add these scores (of the first d words) to the global scores of those words;
}

The variables a, b and c determine the relative weightage given to authorities, hubs and linkage information. The subject set is the set of words in the order of their global scores. The value of the variable d is an important design issue and it determines the number of words whose global score can be influenced by a single page. The value of the variable e does not play any important part in the whole process except that it just scales the scores of each word by the same amount. If d is set to a very low value, only words with utmost significance will appear in the subject set, but the possibility of important words getting excluded exists. A high value for d increases the computation overhead but decreases the probability of words of average significance getting excluded. Thus, the choice of d is a tradeoff between safety and computation overhead. The issues in setting the values of a, b and c are the same as those discussed in 4.1.2. The instance of the algorithm used for testing is given below.

Table 4 Algorithm B1

Global Score of every word=0;
For each page, page$_i$
{
Score of page$_i$ =(number of links targeting page$_i$)+1;
*Local score of a word= frequency of the word in the page$_i$ * score of the page$_i$;*
Pick 5 words having the highest scores and let their scores be {s[0],s[1]…s[4]};
*Scale their scores so that the sum of their scores becomes equal to page_score*100;*
Add these scores (of the first 5 words) to the global scores of those words;
}

This is the instance with a = 1, b = 0, c = 0, d = 5 and e = 100.

4.2.2 Issues related to the algorithm

This variant of algorithm A does away with the disadvantage of a page being able to influence the scores without bounds. A page can add utmost page_score*e to the array of global word scores and can influence global word scores of most d words. Other issues are very similar to those presented under Section 4.1.2.

5 Test setup and descriptions of test sets

For evaluation of the performance of the above algorithms, the following strategy was adopted. A set of pages from among the results to a query in a popular search engine was gathered. Let the query string used be A. The gathered set of pages is given as input to the algorithm. Let the subject set returned be B. The performance of the algorithm used is taken to be the degree of matching between the subject set B and the query string A. The intimacy of other words in the subject set is with the topic of the search query which is useful in cases that are described in Section 2.1.

The search engine used for gathering the sets was Google (http://www.google.com/). The test sets used are further described below:

Sets 1 to 10 were small sets of pages, typically containing between 15 and 20 pages and with 0 to 8 links between them. Then, three more sets were used to investigate the performance of the algorithm on larger collections. They are briefly described as follows.

Some description of the choices of some queries may well help. Edsger Dijkstra was a well known computer scientist who invented the single-source shortest path algorithm for graphs and was an ardent hater of the ‘goto’ statement. HEERA stands for Higher Education Employer–Employee Relations Act and is also the name of an Indian actress. Kochi is a city in the state of Kerala in India. Padma Bhushan is the highest civilian honour in India and famous recipients include musicians like Yesudas. Telgi is the prime accused in a recent fake stamp paper scam in Mumbai, India.

Table 5 Test collections

Set	*Query used*	*Pages*	*Links within*
1	birthday	020	001
2	congress	019	006
3	Dijkstra	020	004
4	HEERA	017	001
5	informatics	020	003
6	Kerala	015	008
7	Kochi police	019	000
8	Padma Bhushan	016	001
9	Siamese	020	002
10	Telgi	018	000
11	Kerala politics	120	006
12	cricket world cup	180	034
13	Bill Gates	215	040

6 Test results

The most heavily weighted five words from the subject sets returned are presented in the results tables below. The more common English words like ‘the’ or ‘of’ were eliminated from consideration using a stop list. The HTML tags were obviously not considered.

6.1 Algorithm A1

Table 6 Algorithm A1 test results

Set-1	*Set-2*	*Set-3*
birthday	Congress	Dijkstra
family	House	computer
party	Congressional	Edsger
child	Information	algorithm
year	Senate	statement
Set-4	*Set-5*	*Set-6*
HEERA	Informatics	Kerala
employee	Volume	document
employees	Information	India
relations	Medical	university
university	Health	music

Table 6 Algorithm A1 test results (Continued)

Set-7	*Set-8*	*Set-9*
police	Padma Bhushan	twins
Kochi	Yesudas	Siamese
news	music	news
cell	India	conjoined
people	Indian	Chang (or Chan?)
Set-10		*Set-11*
Telgi		Kerala
police		left
Scam		document
News		Layer
Stamp		All
Set-12		*Set-13*
world		Bill
cup		Gates
cricket		Microsoft
Australia		Not
more		About

6.2 Algorithm B1

Table 7 Algorithm B1 test results

Set-1	*Set-2*	Set-3
birthday	congress	Dijkstra
party	house	changes
birthdays	congressional	advocaten
web	document	notarison
cards	library	pages
Set-4	*Set-5*	*Set-6*
HEERA	informatics	Kerala
employee	volume	document
relations	Health	family
article	information	window
document	medical	more
Set-7	*Set-8*	*Set-9*
police	Yesudas	twins
Kochi	Padma Bhushan	news

Table 7 Algorithm B1 test results (Continued)

Set-7	*Set-8*	*Set-9*
news	Balachander	Chan (or Chang?)
Japan	Padma	conjoined
Cochin city police	music	girls
Set-10		*Set-11*
Telgi		Kerala
Scam		caste
Police		study
Case		party
Stamp		history
Set-12		*Set-13*
World		Bill
Cup		Gates
Cricket		Microsoft
Australia		Index of
Document		about

7 Conclusions

As can be seen from the test results in Section 6, almost everything worked fine in most cases. Only two results can be classified as disappointing, algorithm B1 on set3 and algorithm A1 on set 11. On further examination, the documents of set 3 were found to be a mixture of Dutch and English, which, might have led to the anomaly. Even with sets having no links within them (set 7 and set 10), the results were very good. An overall evaluation shows that algorithm B1 performed better than A1.

This problem is a broader version of the problem of getting semantic information from hypertext, as it deals with not just extracting semantic information from individual documents but also from sets of documents. Hypertext differs from flat texts in that it is not integrated and is a collection of subunits of flat texts with interlinking. Thus, for hypertext information, links are just as important as text. So, the structuring of information in hypertext documents may be exploited by means of building structures such as HTML struct trees [8]. Such techniques would have to depend heavily on the markup language used and so such techniques have not been used here (due to our anticipation that HTML is going to be replaced soon). The techniques presented here should work as well for any markup language, like they did for the test sets used here which was fully HTML.

References

1 Gibson, D., Kleinberg, J. and Raghavan, P. (1998) 'Inferring web communities from link topologies', *ACM Hypertext*.

2 Kleinberg, J., Kumar, S.R., Raghavan, P., Rajagopalan, S. and Tomkins, A. (1999) 'The web as a graph: measurements, models and methods', *Proc. 5th Annual Intl. Conference on Computing and Combinatorics*.

3 Reddy, P.K. and Kitsuregawa, M. (2001) 'An approach to relate the web communities through bipartite graphs', *Proc. 2nd Intl. Conference on Web Information Systems Engineering*, IEEE Computer Society, pp.301–310.

4 Kemal, E., Raghavan, V., Chu, C.H., Broadwater, A.L., Bolelli, L. and Ertekin, S. (2000) 'Shape of the web and its implications in searching the web', *Proc. Int. Conf. Advances in Infrastructure for Electronic Business, Science, and Education on the Internet*., URL: http://citeseer.nj.nec.com/efe00shape.html.

5 Arasu, A., Novak, J., Tomkins, A. and Tomlin, J. (2002) 'PageRank computation and the structure of the web: experiments and algorithms', *Poster Proc. World Wide Web Conference 2002 (WWW2002)*, Hawaii, May, URL: http://citeseer.nj.nec.com/arasu02pagerank.html.

6 Amitay, E.and Oberlander, J. (1997), 'Convention Says...', *Proc. Flexible Hypertext Workshop Held in Conjunction with the Eighth ACM International Hypertext Conference (Hypertext'97)*.

7 Kleinberg, J. (1999) 'Authoritative sources in a hyperlinked environment', *Journal of the ACM*, Vol. 46, No. 5, pp.604–632.

8 DiPasquo, D. (1998) *Using HTML Formatting to Aid in Natural Language Processing on the World Wide Web*, Senior Thesis, Computer Science Department, Carnegie Mellon University, URL: http://citeseer.nj.nec.com/dipasquo98using.html.