A Dynamic Approach To The Website Boundary Detection Problem Using Random Walks

Ayesh Alshukri
Department of Computer Science,
University of Liverpool, Ashton Building,
Ashton Street, L69 3BX Liverpool, UK.
Email: A.Alshukri@liverpool.ac.uk
http://www.csc.liv.ac.uk/∼ash

Frans Coenen
Department of Computer Science,
University of Liverpool, Ashton Building,
Ashton Street, L69 3BX Liverpool, UK.
Email: Coenen@liverpool.ac.uk
http://www.csc.liv.ac.uk/∼frans

***Abstract*—This paper presents an investigation into the Website Boundary Detection (WBD) problem in the dynamic context. In the dynamic context (as opposed to the static context) the web data to be considered is not fully available prior to the start of the website boundary detection process. The dynamic approaches presented in this paper are all probabilistic and based on the concept of random walks; three variations are considered: (i) the standard Random Walk (RW), (ii) a Self Avoiding RW and (iii) the Metropolis Hastings RW. The reported evaluation demonstrates that the proposed technique produces good WBD solutions while at the same time reducing the amount of "noise" pages visited. The best performing variation was found to be a Metropolis Hastings RW.**

I. INTRODUCTION

The nature of the world wide web (or simply the web) is such that information of almost any type can exist, connected in a multitude of ways, which can provide us with information on many subjects. The web is thus a complex interconnected data structure of great diversity. However, there are intuitive notions associated with the information on the web that are used to add meaning to the underlying structure, that are not explicitly described. The notion of a "website" is one of these. The concept of a website is significant with respect to applications such as automatic website map generation, digital preservation and web spam identification [1]. Thus, although there is no agreed definition associated with the term "website", despite its common usage, we are interested in identifying websites. Hence, the Website Boundary Detection (WBD) problem. In the context of the work presented in this paper the WBD problem is defined as follows: *The problem of automatically learning the set of web pages/resources/media that are part of a single website* (leaving aside, for the time being, any specific notion of what a website might be).

This paper presents an investigation into the WBD problem in the dynamic context. In the dynamic context the web data is not fully available prior to the detection of a website's boundaries, whereas in the static context the web data of interest has been collected a priori. The advantage offered by the dynamic context, over the static context, is that the static context requires the prior collection of a large number of www resources, substantially greater than the expected size of the web site of interest, so as to ensure that "nothing is missed". The proposed solution presented in this paper uses various random walk graph traversal techniques to gather portions of web data, which are then incrementally clustered, to produce a WBD solution using many fewer web pages than in the case of the static context. The random walk concept was selected, with respect to the work described in this paper, because of its beneficial application in related areas of computer science [13], [17]. More specifically we consider three different random walk approaches: (i) the standard Random Walk (RW), (ii) the Self Avoiding Random Walk (SARW) and (iii) the Metropolis Hastings Random Walk (MHRW). These approaches were selected as they provide a diverse range of methods for randomised graph traversal.

The main contributions of this paper are thus: (i) an investigation of the "power" of random walk based methods of graph traversal for the purpose of WBD in the dynamic context, (ii) an evaluation of the most appropriate web page feature representation and (iii) an evaluation of the most appropriate random walk traversal technique in the context of the dynamic WBD problem.

II. RELATED WORK

The Website Boundary Detection (WBD) problem is an open and difficult problem to solve [6], [5], [11], [2], [3], [4], [12], [1]. As already noted, the WBD problem is concerned with the task of identifying the complete collection of web resources that are contained within a single website. The field of "website mining" [10], [19], [7], [15] describes an area of research that falls within the field of "web mining", but directed at specific applications with respect to a website's structure (in contrast to web page structure or content for example) such as WBD. Website mining is broadly directed at the discovery of hidden patterns or knowledge in websites. Website mining techniques utilise web structure (hyperlinks) and web content (attributes of a page) to solve problems such as the WBD problem. This section presents some selected related work which discusses: link block graphs, logical websites, compound documents, directory based websites and website hierarchies with respect to the WBD problem.

The WBD problem has been approached by using methods based on the link block graph representation. A link block graph is created from web pages by segmenting individual pages into "blocks" representing navigation menus or sets of coherent links. This is typically done by segmenting the HTML Document Object Model (DOM) of web pages. Once segmented a graph structure can be used to represent the link blocks (as a link block graph). In the work by Rodrigues [16], and related work by Keller [12], the main emphasis was directed at link blocks that represent structural menus (s-menus) of web pages within a web graph. A link between web pages exists when a link from a s-menu can be used to navigate to another page that contains the same s-menu. Web pages that contain common s-menus can be used to define the boundaries of a website.

In the work by Senellart [18] the aim was to find web pages that are contained in "logical websites". A logical website is described in terms of the link structure and is defined as a collection of nodes (or web pages) that are significantly more connected than other nodes (a definition also used later in this paper). The web is thus modelled using flow networks. An assumption is made that if a flow is pushed around the network "bottlenecks" will occur in the less connected noise pages, but flow more freely around the highly connected target pages of the website. To detect the boundaries of a website flow is pushed from a set of seed pages until a bottle neck is created around the boundaries of the website. The set of nodes that have flow pushed to them between the seed and the "bottle neck" are then identified as part of the website.

Research by Eiron [9] proposed the notion of *compound documents*; a set of web pages that can be aggregated in to a single coherent information entity. A simple example of a compound document is an online multi-media news article. In the work by Dmitriev [8] a method to discover compound documents is proposed. The method uses supervised learning techniques to train a model using manually labelled compound documents.

In [5] a "directory based (web) site" is described as a section of a web server over which an author has full control. The method proposed to identify a website uses various filters based on characters in the URLs of web pages so as to categorise which child pages fall under the control of a certain author and thus identifying a website's boundaries in terms of authorship.

A website hierarchy is a tree structure that can be used to represent the organisation of a collection of web pages based on the link structure. A method of extracting a hierarchy from a collection of web pages describing a website is proposed in [20]. The method uses a machine learning approach to weight edges based on the different predetermined categories of links. A shortest path algorithm is then used, rooted at the home page, to build the tree structure to represent the hierarchy of the website.

III. Formal Definition

In this section a formal definition of the WBD problem is presented. Given a collection of web pages W, comprising n individual pages such that $W = \{w_1, w_2, \cdots, w_n\}$, where a particular page w_s is denoted as the seed (home) page; a website boundary (ω) is said to be the bounded subset of pages in W that form the website related to w_s. Note that each of the individual web pages can be described using a m dimensional feature vector $V = \{v_1, v_2, \ldots, v_m\}$ and that a collection of pages can be modelled using a graph $G = (W, E)$, where W is the collection of web pages (as noted above), and E is the set of directed (hyper) links connecting pairs of pages in W. Recall that the key characteristic of the work presented in this paper is that the WBD problem is considered in a dynamic context. This implies that the graph $G = (W, E)$, and consequently the sets W and E, is not known apriori. Thus at commencement of the dynamic approach to identify a solution to the WBD problem, all that is known is the seed page w_s, and its associated directed edges. As the process proceeds we wish to distinguish between "target pages", pages that belong to the website of interest (the set/cluster of pages K_T) and "noise pages" (the set/cluster of pages K_N). It is the set K_T that we wish to identify. The following section, Section V, presents the proposed dynamic approach to the WBD problem using Random Walks.

IV. The Dynamic Approach

The three proposed dynamic approaches considered in this paper are founded on an incremental approach to producing a solution to the WBD problem. The fundamental methodology in each case comprises the template presented in Table I. At the start (line 1) the set of target pages contains only the seed page ($K_T = \{w_s\}$) and the set of noise pages is empty ($K_N = \{\}$). The "process internal states" referred to in line 2 is concerned with the clustering parameters, thus in the case of k-means the adjustment of the cluster centroids (prototypes). Lines 4 and 5 are concerned with the two most important aspects that underpin the proposed dynamic approaches: (i) Graph Traversal (line 4) and (ii) Incremental Clustering (line 5). The adopted graph traversal method, the random walk method, dictates the process whereby web graph nodes (web pages) are selected and visited, starting from the initial seed page w_s. Further detail concerning the graph traversal techniques presented in this paper is given in Section V below.

As the traversal progresses, for each identified node w (web page), we determine whether the current node belongs to K_T or K_N. To do this an incremental clustering method was adopted; more specifically the Incremental Kmeans (IKM) algorithm was used with respect to the evaluation presented later in this paper. IKM is based on the classic k-means algorithm [14] adapted to fit the incremental template given in Table I by assuming that:

1) The state of the process contains information about the (two) clusters centroids (K_T and K_N) based on all clustered items sofar.
2) The pages in G are inspected one at a time in some order.
3) The state update (centred calculation) is performed after some suitable number of pages have been visited.

A key feature of IKM, in the context of dynamic WBD, is that the order in which records (nodes) are considered is subject to the nature of the traversal of G, thus the list of nodes visited will change over subsequent iterations and will "sooner or later" include repetitions. A new page w is added to cluster K_T or K_N according to its closest similarity with the current cluster centroids. Note that if a previously visited page is traversed, it may be reassigned to a different cluster. The process continues (the loop from lines 3 to 7 in the template presented in Table I, until the system state (cluster centroids) does not change, or a termination criteria is reached. Due to the differing operation of the graph traversal methods considered (see below) the clusters produced can vary greatly. Hence the notion of a 'step' is incorporated into the experimental analysis of the techniques. Referring to the template presented in Table I a step is considered to be a single iteration of the loop from lines 3 to 7. The number of steps required for an algorithm to identify a WBD solution is thus a useful comparison measure.

```
Algorithm clustering_template ($w_s$)
1: $K_T = \{w_s\}$; $K_N = \{\}$;
2: set up the process internal state;
3: repeat
4:     select web graph node $Q$ to visit next;
5:     add $Q$ to $K_T$ or $K_N$;
6:     update the process state;
7: until convergence;
8: return $K_T$;
```

TABLE I: Template for a dynamic approach to WBD.

V. Graph Traversal

This section details the three random walk graph traversal methods considered: (i) standard Random Walk (RW), (ii) Self Avoiding Random Walk (SARW) and (iii) the Metropolis Hastings Random Walk (MHRW). These are all probabilistic approaches. The distinction between a deterministic and a probabilistic approach is that, given a web graph G, the first will always traverse the graph in the same manner while the second will (at least potentially) traverse the graph in a different manner each time the algorithm is applied to G. Deterministic approaches use a heuristic process to determine which node to visit next thus the ordering of web pages accessed using a deterministic method remains fixed for every traversal of the same graph. In the probabilistic approach there is an element of choice and unpredictability involved. Each graph traversal method is given in detail in the following subsections, with respect to the dynamic approach template that was presented in Table I.

A. Random Walk (RW)

The Random Walk (RW) method of graph traversal is described by the pseudo code presented in Table II. Given a current page Q (which at start up will be the seed page) the page to be visited next is selected at random from the immediate neighbours of Q in G. Note that the walk can revisit nodes multiple times and consequently reassign nodes to either K_T or K_N. Note also that using the RW approach the process state is recomputed after each step. Clearly any random walk on a finite connected graph will eventually visits all the vertices in the graph. Thus, in principle, the process could run until convergence is achieved (the K_T and K_N clusters become stable). However, experiments conducted by the authors (and not reported here because of space considerations) have indicated that stopping the process after a given maximum number of steps (MaxIterations) is more efficient and still results in a good quality WBD solution.

B. Self Avoiding Random Walk (SARW)

The Self Avoiding Random Walk (SARW) "crawls" the graph in a random manner without returning to previously visited nodes. It does this by maintaining a list R of nodes visited so far. The pseudo code is shown in table III. The process continues until some randomly generated number (between 0 and 1) is greater than $(reset)^j$ (were $reset$ is a user supplied "seed" between 0 and 1, and j is a measure of the number of noise pages that have been visited so far mitigated by the number of target pages visited so far) at which point the process is resumed with $R = \{\}$. Thus the more noise pages that are visited the more likely that the algorithm will reset. Note also that the reset parameter r can be adjusted to control the sensitivity of the walk.

```
Algorithm RW ($w_s$)
  $K_T = \{w_s\}$; $K_N = \{\}$;
  set $Q$ to $w_s$; set a counter to one;
  set up the process internal state;
  repeat
      redefine $Q$ to be a random neighbour of $Q$ in $G$;
      add (or reassign) $Q$ to $K_T$ or $K_N$;
      increase the counter;
      update the process state;
  until counter goes past MaxIterations;
  return $K_T$;
```

TABLE II: Pseudo code for Random Walk (RW)

C. Metropolis Hastings Random Walk (MHRW)

Intuitively the RW and SARW methods are based on a random traversal of the graph which can be effected dramatically by the underlying graph structure. If a node has a high degree, then it is more likely to be chosen randomly, as it has an increased number of neighbours. The Metropolis Hastings Random Walk (MHRW) aims to reduce this behaviour. The approach taken is to use a calculation which effectively produces an inverse probability of choosing a neighbour which has a high degree. The pseudo code for MHRW is shown in Table IV. The function $Deg()$ simply returns the degree of the given node. Thus as the value of $Deg(Q)/Deg(Q_{new})$ increases, the likelihood of $random\ number > Deg(Q)/Deg(Q_{new})$ decreases, and thus there is a decreasing chance of not moving (staying at the current node).

VI. Evaluation

This section presents a comparison of the operation of the three proposed random walk approaches with each other and with respect to straightforward Depth First (DF) and Breadth First (BF) approaches. The DF and BF approaches are deterministic graph traversal approaches which visit graph nodes (web pages) in a fixed order. The DF and BF approaches were included in the evaluation so as to provide for a "base line" with which the proposed probabilistic random walk approaches could be compared. The rest of this section is organised as follows. Details concerning the datasets used for the experiments are presented in Subsection VI-A, whilst the adopted evaluation criteria is discussed in Subsection VI-B. The performance of the proposed approaches was evaluated in two respects. Firstly a comparison using five different feature representations (Body text, Title text, Script links, Resource links and Image links) was considered in terms of WBD performance. The results of this set of experiments are presented in Subsection VI-C. Secondly the WBD performance of the proposed approaches (RW, SARW and MHRW), and BF and DF, were compared with each other using the best performing feature representation from the previous set of experiments. The results of this set of experiments are presented in Subsection VI-D.

```
Algorithm SARW ($w_s$, $r$)
  $K_T = \{w_s\}$; $K_N = \{\}$;
  $reset = r$;
  set $Q$ to $w_s$; set a counter to one;
  set up the process internal state;
  loop
    increase counter;
    define $v_0$ to be an element of $K_T$;
    $i = 1$; $j = 0$; $R = \{v_0\}$;
    repeat
      $v_i \leftarrow$ random neighbour of $v_{i-1}$ NOT in $R$;
      $Q = v_i$;
      add $Q$ to $R$;
      add $Q$ to $K_T$ or $K_N$;
      update the process state;
      if $Q$ added to $K_T$ then
        decrease $j$ (if positive);
      else
        increase $j$;
      end if
      increase $i$;
    until random number $> (reset)^j$;
  end loopcounter goes past MAXITERATIONS
  return $K_T$;
```

TABLE III: Pseudo code for Self Avoiding Random Walk (SARW)

A. Datasets

The four data sets used for the evaluation were created using departmental web pages hosted by the University of Liverpool (www.liv.ac.uk). The web pages collected were manually labelled by an assessor with respect to the WBD solution. Each data set contained approximately $450 - 500$ web pages. The four departments were: (i) Chemistry (LivChem), (ii) tHistory (LivHistory), (iii) Mathematics (LivMaths) and (iv) Archaeology, Classics and Egyptology (LivAce)[1]. These departments were selected so as to provide non-trivial examples of the WBD problem. This was in contrast to selecting very dissimilar web pages in which the WBD problem becomes a trivial task.

B. Evaluation Criteria

This section details the criteria used for the evaluation of the dynamic approaches presented in this work. A WBD solution is given as a partition of the set of web pages into a target cluster K_T and a noise cluster K_N where K_T contains the given seed page. Standard evaluation metrics taken from the domain of data mining were used: accuracy, precision, recall and the Fmeasure. Two additional measures, score and coverage, were also used. The "score" for a particular WBD solution is the average of the measured accuracy, precision, recall and Fmeasure. The "coverage" for a solution (described by the noise cluster K_T) is a measure of the number of target pages that should be included in K_T versus the number of target pages actually allocated to K_T. More formally $coverage(K_T) = \frac{M_{TP}+M_{FN}}{|K_T|}$, where: (i) M_{TP} is the number of pages/elements correctly allocated to the target cluster (True Positives) and (ii) M_{FN} is the number of pages/elements that should be allocated to K_T but are wrongly allocated to the noise cluster K_N (False Negatives). The coverage of the noise cluster K_N is calculated in a similar manner, $coverage(K_N) = \frac{M_{TN}+M_{FP}}{|K_N|}$, where: (i) M_{TN} is the number of pages/elements correctly allocated to the noise cluster (True Negatives) and (ii) M_{FP} is the number of pages/elements that should be allocated to K_N but have been wrongly allocated to the target cluster K_T (False Positives). A higher K_T (K_N) coverage value indicates that a high number of target pages are included in K_T (K_N) together with a low number of noise (target) pages. We also use the number of steps (see end of Section IV) as an additional evaluation metric. The number of steps required to achieve a WBD solution is an indicator of "time complexity".

```
Algorithm MHRW ($w_s$)
  $K_T = \{w_s\}$; $K_N = \{\}$;
  set $Q$ to $w_s$; set a counter to one;
  set up the process internal state;
  repeat
    redefine $Q_{new}$ to be a random neighbour of $Q$ in $G$;
    if $random\ number > Deg(Q)/Deg(Q_{new})$
    then
      {dont move} $Q = Q$;
    else
      {move} $Q = Q_{new}$;
    end if
    add $Q$ to $K_T$ or $K_N$;
    increase the counter;
    update the process state;
  until counter goes past MAXITERATIONS;
  return $K_T$;
```

TABLE IV: Pseudo code for Metropolis Hastings Random Walk (MHRW)

Fig. 1: Performance score with respect to the five different web page feature representations considered.

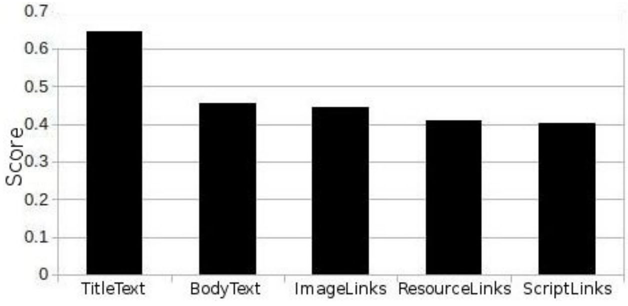


TABLE V: The WBD performance for the dynamic approaches considered

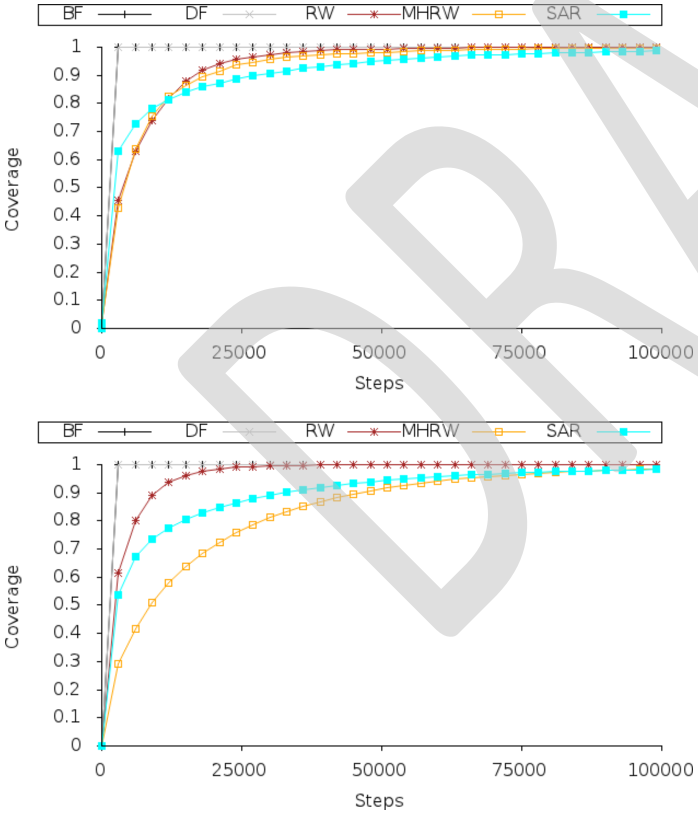
	Score	**Coverage** K_T	**Coverage** K_N	**Time (ms) / Step**	**Total Steps**
MHRW	**0.791**	0.941	0.768	4.74	25000
BF	**0.717**	1	1	314.801	842
DF	**0.714**	1	1	320.561	842
RW	**0.632**	0.96	0.992	5.957	25000
SARW	**0.506**	0.89	0.869	5.611	25000

[1] Data sets available at www.csc.liv.ac.uk/~ash

C. *Web Page Feature Representation Evaluation*

The performance of the proposed approaches with respect to a range of feature representations (Title text, Body text, Script links, Resource links and Image links) and in comparison with DF and BF is presented here. Each of the representations were created by parsing the HTML source of the web pages and creating a vector space model. The results obtained in terms of the score value are presented in Figure 1. Inspection of the figure indicates that the Title text representation exhibited the best performance in terms of solutions to the WBD problem. It is suggested that the Title text representation produced the best performance because: (i) the title of a web page is written by the author/publisher to convey a summary of the page that is quickly digestible by users and (ii) the title typically rounds up the subject or context of a web page in a concise manner. The Script, Resource and Image links representations produced the worst performance. Also, the Body text representation, contrary to expectations, did not serve to produce a good result (although it was second best).

Fig. 2: The average coverage with respect to: (up) K_T the target set of web pages and (down) K_N the noise set of web pages (Title text representation used in both cases)



D. *Dynamic Approach Evaluation*

This section presents an evaluation of the WBD solutions produced using RW, SARW, MHRW, DF and BF and the Title text web page representation (the best performing representation from the previous experiments reported above). The results are shown in Table V. The table gives the Score, Coverage of K_T and K_N, Time (ms) per steps (the run time required to complete a single step in the graph traversal) and the total number of steps (the stopping criterion in terms of steps). From Table V it can be seen that MHRW produced the best WBD performance in terms of Score. It also has a high value of K_T coverage indicating that the approach visited a larger number of target pages while at the same visiting a low number noise web pages as shown by K_N coverage. The coverage of BF and DF was 1 for K_T and K_N, this indicated that all nodes were visited (thus the number of step is equivalent to the total number of nodes in the graph).

Figure 2 presents the coverage results obtained, with respect to K_T and K_N (reported as an average of the datasets using the Title text representation) as the different approaches proceed (measured in terms of the first 10,000 steps). Figure 3 in turn gives the Accuracy, Fmeasure, Recall and Precision results obtained as the different approaches proceed (measured in terms of the first 1,000 steps). From Figure 2 it can be observed that the K_N coverage value increases at a much slower rate compared with coverage of K_T using approaches MHRW, SARW and RW respectively. This is shown over 10,000 steps to illustrate how the crawls progressed. From Figure 3 the comparative performance of the approaches can be observed with respect to the initial stages of the walks (the first 1,000 steps). From the figure it can also be seen that the RW and MHRW methods provide much more stable results in terms of precision and recall than BF and DF. This is due to the repeated traversal of higher connected nodes of the website, in contrast to BF and DF which consider and new node at each step making more volatile cluster assignments.

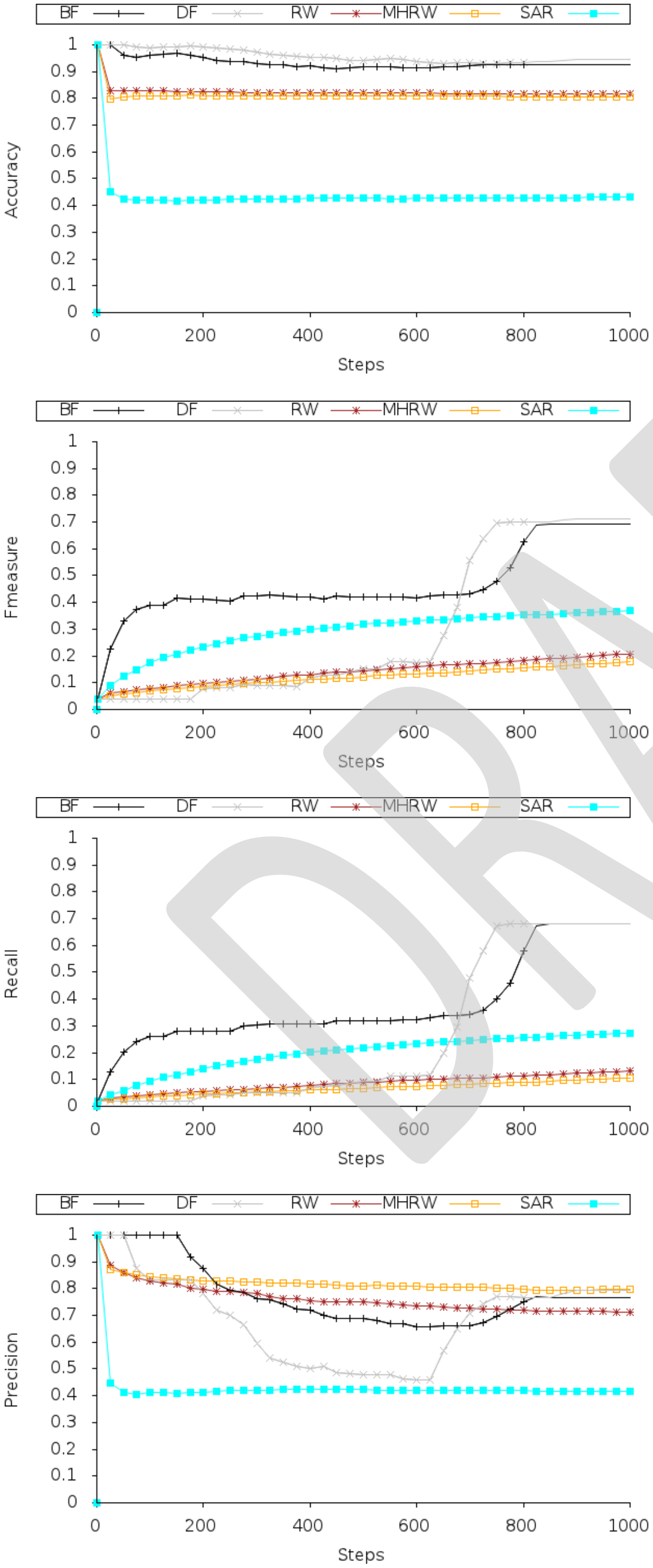
VII. Conclusion

This paper presented an investigation into the WBD problem in the dynamic context. In the dynamic context the web data is not fully available prior to the start of the analysis. Three dynamic approaches were presented in this paper (RW, SARW and MHRW) which used different graph traversal techniques to gather portions of web data, which were then clustered incrementally in order to produce a WBD solution. The distinction between the RW, SARW and MHRW approaches and the DF and BF approach is that they traverses the web graph by selecting pages to be visited in a random order depending on the hyperlink structure. The distinction between MHRW and RW and SARW is that MHRW avoids high degree pages. The conducted evaluation indicated that the MHRW approach, coupled with the Title text web page feature representation, produced the best performance. It is suggested that the MHRW approach produced the best performance because: (i) fewer noise web pages are visited than in the case of the other methods considered and (ii) the random selection of next pages to be visited. The characteristics of the MHRW approach translate into a solution that requires fewer resources as it minimises the downloading and parsing of unnecessary noise pages.

References

[1] A Alshukri. *Website Boundary Detection via Machine Learning*. Thesis, University of Liverpool, 2012.

Fig. 3: The WBD performance (over the first 1000 steps) for the five graph traversal approaches, using the Title text feature representation, reported as an average calculated from across the data sets.



[2] A Alshukri, F. Coenen, and M. Zito. Web-Site Boundary Detection. In *Proceedings of the 10th Industrial Conference on Data Mining*, pages 529–543, Berlin, Germany, 2010. Springer.

[3] A Alshukri, F. Coenen, and M. Zito. Incremental Web-Site Boundary Detection Using Random Walks. In *Proceedings of the 7th International Conference on Machine Learning and Data Mining.*, pages 414—-427, New York, USA, 2011. Springer.

[4] A Alshukri, F. Coenen, and M. Zito. Web-Site Boundary Detection Using Incremental Random Walk Clustering. In *Proceedings of the 31st SGAI International Conference*, pages 255—-268, Cambridge, UK, 2011. Springer.

[5] Y Asano, H. Imai, M. Toyoda, and M. Kitsuregawa. Applying the Site Information to the Information Retrieval from the Web. In *Proceedings of the Third International Conference on Web Information Systems Engineering, 2002. WISE 2002.*, WISE 2002, pages 83–92. IEEE Computer Society, 2002.

[6] K. Bharat, B-W. Chang, M. Henzinger, and M. Ruhl. Who links to whom: mining linkage between Web sites. In *Proceedings 2001 IEEE International Conference on Data Mining*, pages 51–58, Washington, DC, USA, 2001. IEEE Computer Society.

[7] K-W. Cheung and Y. Sun. Mining Web Site's Clusters from Link Topology and Site Hierarchy. In *Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence*, page 271, Washington, DC, USA, October 2003. IEEE Computer Society.

[8] P. Dmitriev. As we may perceive: finding the boundaries of compound documents on the web. In *Proceeding of the 17th international conference on World Wide Web*, pages 1029–1030, Beijing, China, 2008. ACM.

[9] N. Eiron and K. S. McCurley. Untangling compound documents on the web. In *Proceedings of the fourteenth ACM conference on Hypertext and hypermedia*, pages 85–94, New York, USA, 2003. ACM Press.

[10] M. Ester, H-P. Kriegel, and M. Schubert. Web site mining: a new way to spot competitors, customers and suppliers in the world wide web. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 249—-258, New York City, USA, 2002. ACM.

[11] M. Henzinger. Finding near-duplicate web pages: a large-scale evaluation of algorithms. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 284–291. ACM, 2006.

[12] M. Keller and M. Nussbaumer. MenuMiner: Revealing the Information Architecture of Large Web Sites by Analyzing Maximal Cliques. In *Proceedings of the 21st international conference companion on World Wide Web*, page 1025, New York, USA, April 2012. ACM Press.

[13] L. Lovász. Random walks on graphs: A survey. *YaleU/DCS/TR-1029*, 2:1–46, 1994.

[14] J MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley symposium*, 1967.

[15] S. V. Ramnath and P. Halkarnikar. Web Site Mining Using Entropy Estimation. In *International Conference on Data Storage and Data Engineering*, pages 225–229. IEEE, February 2010.

[16] E. M. Rodrigues, N. Milic-Frayling, M. Hicks, and G. Smyth. Link Structure Graphs for Representing and Analyzing Web Sites, 2006.

[17] P Sarkar and AW Moore. Random Walks in Social Networks and their Applications: A Survey. *Social Network Data Analytics*, pages 43–77, 2011.

[18] P. Senellart. Website Identification. Masters thesis dea internship report, Université Paris XI, Orsay, France., September 2003.

[19] Y. Tian. A web site mining algorithm using the multiscale tree representation model. In *Proceedings of the 5th Webmining as a Premise to Effective and Intelligent Web Applications*, 2003.

[20] Christopher C. Yang and Nan Liu. Web site topic-hierarchy generation based on link structure. *Journal of the American Society for Information Science and Technology*, 60(3):495–508, March 2009.