Mining the Information Architecture of the WWW using Automated Website Boundary Detection

Ayesh Alshukri and Frans Coenen
Dept of Computer Science, University of Liverpool, Ashton Building, Ashton Street, L69 3BX, Liverpool, UK.
E-mail: a.alshukri@liverpool.ac.uk, coenen@liverpool.ac.uk

Abstract. The world wide web has two forms of architecture, that explicitly encoded into web pages, and that implied by web content and look and feel. The latter is exemplified by the concept of a website, a concept that is only loosely defined although users intuitively understand it. Whatever the case the concept of a web site is used with respect to a number of application domains, including website archiving and analysis. In the context of such applications it is beneficial if a website can be automatically identified. This is usually done by identifying a website of interest in terms of its boundary, the so called Website Boundary Detection (WBD) problem. In this paper seven WBD techniques are proposed and compared, four statical techniques where the web data to be used is obtained apriori, and three dynamic techniques where the data to be used is obtained as the process progresses. All seven techniques are presented in detail and evaluated in this paper.

Keywords:
Web structure mining, Web Graphs, Web Site Boundary Detection, Random Walk Techniques, Web page Clustering, Web Archiving, Digital Preservation

1. Introduction

A disconnect exists between the underlying structure of the Web and the structure perceived by humans. Humans can intuitively identify an implied organisation of the Web that is not obvious from the underlying explicit structure. More specifically, humans are able to infer website boundaries in a manner that is not necessarily explicitly encoded in any underlying structure. Conceptually, this human perceived architecture exists in a layer above the encoded structure of the web. The process whereby humans derive this architectures is founded on relationships and/or similarities between features encoded in the web, such as the topic or subject of web pages, layout, navigation menus, imagery and so on. Although humans are able to interpret the WWW in this way, it is a difficult task for a machine to achieve. The ability for a machine to identify this underlying architectures remains an open problem [10].

The standard graph model of the web assumes that each web page is a single node within a graph, with hyperlinks (edges) connecting it to other nodes. This model has been used for various web analysis tasks [11,18,32]. The advantage of this standard model is that it is explicitly encoded in the Web and it can be easily extracted using simple web crawls, however it conveys very little information about the higher level web architectures that humans can perceive.

The work presented in this paper is motivated by a number of applications whereby knowledge of the higher level web architecture, comprised of collections of websites each delimited by a web site boundary of some kind, is required. More specifically the work is motivated for a need for automated Website Boundary Detection (WBD), knowledge that cannot be directly obtained from simple web graph analysis. Typical applications where website boundary knowledge is required include automated: (i) Website archiving and digital preservation [49,1], (ii) Web directory generation [30], (iii) Website map generation [18,33,34] and (iv) Web spam detection [8,9,13,31,37,51,52,53]. Further more, the study of the WWW at the website level, rather than the web-page level, provides benefits with respect to a variety of web analysis activities [10]: (i) Content authorship analysis [17], (ii) Inter (between) and intra (within) website connectivity analysis [32], and (iii) Statistical analysis of dynamic website growth and website generation on a website level [11]. Finally, arguably the most prominent motivation for WBD is for the purposes of digital preservation; the identification of complete web (site) content and its archiving for future generations [1,12,16,49].

The main challenge of WBD is the resource cost required to access, store and process web content. In terms of accessing web content there is an overhead associated with the retrieval of content; as content is stored on servers around the world there is a time and bandwidth cost associated with accessing pages/resource on the web. In the context of WBD, so as to use computational resources effectively, we wish only to consider relevant pages. The process of WBD can be considered in either a static or dynamic context. The distinction is that in the static context all the www pages to be considered are first downloaded. In the dynamic context only portions of the Web are considered in turn, the analysis is thus conducted in a step by step manner. The advantage of the first is that decisions on what groupings to classify web pages into, either contained within a website (target pages) or outside of the website (noise pages), can be made based on all data apriori. The advantage of the second is that the amount of irrelevant (noise) pages that are requested, stored and processed can potentially be reduced, as the gathering of pages is considered at the same time as the classifying of pages which in turn can be used to inform the decision of whether to access subsequent pages or not.

In this paper both static and dynamic WBD algorithms are considered. Recall from the foregoing that the distinction between the two is that in the first case the search space, a collection of web pages, is obtained in advance, while in the second case the search space is explored dynamically. Two static WBD approaches are considered: (i) content based and (ii) structure based. The focus of the first, as the name suggests, is the content of web pages. The idea is to identify the most appropriate attributes for describing web page content and to use these as features to represent web pages. Web pages are then clustered using these features in order to identify a websites boundary. Two static content based techniques are considered: (i) the Composite Feature (CF) technique and (ii) the Discriminant Feature (DF) technique. The focus for the static structure based WBD approach is the hyperlink structure of web pages, as such the approach is founded on graph partitioning methods. Two graph partitioning techniques are considered in the context of the static structure based WBD approach: (i) Hierarchical Graph Partitioning (HGP) and (ii) Mincut Graph Partitioning (MGP). The proposed dynamic technique is founded on the use of a *Random Walk* (RW) web graph traversal coupled with an incremental k-means clustering mechanism. Three variations are considered: (i) Random Walk (RW), (ii) Self Avoiding Random Walk (SARW) and (iii) Metropolis Hastings Random Walk (MHRW).

The remainder of the paper is organised as follows. In Section 2 a literature review of some existing work on the resolution of the WBD problem is presented. Section 3 then presents a formal description of the WBD problem. Section 4 presents the first of the static approaches, namely the content based approach (two alternative techniques). The second static approach, the structure based approach, is presented in section 5 (two alternative techniques). Section 6 then presents the proposed dynamic approach to the WBD problem based on Random Walks (three alternative techniques). A complete evaluation is then presented in Section 7 and some conclusions in Section 8.

2. Literature Review

The Website Boundary Detection (WBD) problem is recognised as an open and difficult problem to solve [4,5,6,7,10,27,28,29]. 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. WBD broadly falls within the domain of "web mining" [14,21,42,50]. Web mining is directed at the discovery of hidden patterns or knowledge in web data using techniques that utilise web structure (hyperlinks) and web content (attributes of a page). This section presents some selected related work which discusses a number of alternative techniques, to that presented in this paper, that have been used to address the WBD problem, namely: (i) link block graphs, (ii) logical websites, (iii) compound documents and (iv) directory based websites.

Starting with the link block graph technique, a link block graph is a set of vertices and edges that represent certain elements of a web page referred to as a block (a coherent sets of information). Examples of such blocks are a navigation menu across the top of a page, a set of related links at the side of a page or a section of text and images. Given a set of web pages a link block graph is typically created from these pages as follows. First the individual pages are segmented into "blocks". These individual blocks are created by segmenting the HTML Document Object Model (DOM). Second a graph structure of vertices and edges is created from the blocks. Each block is a vertex. An edge between blocks is created based on a notion of similarity. The results is a link block graph. In [44], and in related work presented in [29], the link block graph concept was used to model web data, more specifically link blocks were used to represent structural menus (s-menus) of web pages; an s-menu is a link block graph, as described above, but describing only the internal navigation elements of web pages. A relationship 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 [48] the aim was to find all web pages that are determined to be contained in a "logical website". A logical website in this context is a collection of pages that is said to represent a logical structure and content, such that they are more connected in terms of hyperlinks than other pages. An idea also adopted with respect to work presented later in this paper. The work by Senellart used the concept of flow networks in order to discover web pages that made up a logical website. 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. The significance of the work by Senellart is that the concept of flow networks is also adopted with respect to the work presented later in this paper.

Research by Eiron [20] 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 [17] a method to discover compound documents is proposed. The method used supervised learning techniques to train a model using manually labelled compound documents. This method was then applied to unlabelled web pages in order to determine which of these pages should be aggregated to represent a compound document. This is a similar problem to the WBD problem, but applied in a supervised learning environment.

In [7] the concept of a "directory based (web) site" is described; the partition of a web server over which some individual (website author/owner) has full control. The method proposed to address the WBD problem in this case used various filters based on characters and directories in the URLs of web pages so as to categorise which child pages fall under the control of which individual and thus identifying a website's boundaries in terms of authorship. The disadvantage of this technique is that it assumes that all pages contained within a website are located within the same directory hierarchy as one another. In this paper it is argued that this is often not the case, as individual web pages, images and other web resources can be located using different urls and servers but can still make up a single website.

3. Formal Description

The general WBD problem can be formally described as follows. A web page is defined as a WWW resource, written in HTML format, and denoted by w. A collection of n web pages is given by W such that $W = \{w_1, w_2, \cdots, w_n\}$. A web page can be represented according to both its (hyperlink) structure and its (HTML) content. In the remainder of this paper the terms page, web page, node and vertex are used interchangeably, but in all cases we are referring to an element w_i in W.

With respect to the work presented in this paper the hyperlink structure of a set of web pages is represented as a (web) graph G such that $G = (V, E)$, where $V = \{v_1, v_2, \dots v_n\}$ is a set vertices each representing a web page (V thus corresponds to W) and $E = \{e_1, e_2, \dots, e_p\}$ is a set of edges representing connections between www pages (the presence of at least one link). A specific edge is indicated using the notation (v_i, v_j) where $v_i, v_j \in V$. A web page can of course reference it self, v_i, v_i. Without loss of generality we assume that G is connected.

The HTML content of a page w_i can be conceptualised as a set of features whereby each page has a numerical feature vector associated with it. There exists a function $f : W \rightarrow \mathbb{R}^k$, for some fixed integer $t \geq 1$, such that for any $w_i \in W$, $f(w_i)$ is an ordered sequence of t real numbers characterising the page w_i. A common method to represent a set of numerical features, and the one used in this paper, is the vector-space model [45]. Using a vector space model each web-page, w_i, is described in terms of a feature vector, $F_i = \{f_1, f_2, \ldots, f_k\}$ of length k, which resides within some k dimensional feature (vector) space. Each dimension of the feature space f_i describes a feature. We use the notation $\mathbf{F}$ to indicate the complete set of feature vectors associated with a set of web pages W, $\mathbf{F} = \{F_1, F_2, \ldots, F_n\}$. Note that there is a one to one correspondence between W and $\mathbf{F}$ ($|W| = |\mathbf{F}|$).

3.1. WBD problem

Using the above formalism, the WBD problem can be defined as follows. Given a collection of web pages W, comprising n individual pages such that $W = \{w_1, w_2, \cdots, w_n\}$, we wish to find the sub-set of W that describes a "bounded" set of web pages (ω) in W, in other words a website, centred on some given seed page w_s (typically a known home page, entry or landing page). The desired set ω is thus a collection of web pages that are in some sense similar to the seed page w_s. The set of web pages, B, located on the "circumference" of ω then marks out the boundary of the web site centred on w_s ($B \subseteq \omega$). The boundary set B is thus a collection of web pages where, for each $w_i \in B$, w_i has at least one hyperlink that connects to a web page that is not included in ω.

A basic WBD solution used later in this paper is founded on a clustering process. This subsection is thus concluded with a brief description of this process. Broadly, clustering is used to groups similar web pages in W together based on their content, more specifically on the values in an associated set of feature vectors F. A cluster configuration $K = \{g_1, g_2, \ldots, g_k\}$ is essentially the set of pages W divided into groups (two or more) of related or similar pages as determined by a particular clustering algorithm. One of the clusters in the set K will contain w_s and is therefore identified as the target cluster K_T ($K_T \in K$). Cluster K_T will therefore contain pages most similar to w_s. The remaining clusters, identified as K_N, are the noise or irrelevant clusters ($K = K_T \cup K_N$ and $\{\} = K_T \cap K_N$). The target cluster equates to ω, from which in turn we can identify B and resolve the WBD problem.

3.2. Static vs. Dynamic

As noted in the introduction to this paper the WBD problem is considered in both the static and dynamic contexts. In the static context, the entire vector space is made available prior to the commencement of the WBD process. Thus the collection of web pages $W = \{w_1, w_2, \cdots, w_n\}$ and the set of features for the web pages $F = \{f_1, f_2, \ldots, f_m\}$ is known apriori. In this case clustering of the form described above, to produce K ($K = \{K_T, K_N\}$), is conducted using all the available data.

In the dynamic context the entire vector space is not known in advance. The web pages in the vector space are populated in a dynamic manner as part of a web traversal (crawling) process. This essentially means that the vector space is updated at each "step" of the traversal process. The seed page w_s is used as the starting point for the traversal. Prior to the start $V = \{\}$ and $K = \{\}$. In the dynamic context, unlike the static context, during the early stages of the process the vector space will contain no, or only partial information, concerning the web pages of interest. In other words the size of the search space, in terms of web pages, is not known in advance. This has implications for the clustering process. The clustering algorithm has to make decisions based on the web pages collected so far. This means the clustering method has to operate in an incremental manner.

4. Content Based Static WBD

This section presents the first of the static WBD approaches, namely the content based static WBD approach. Recall that in the static context the search space is identified apriori. The content based static WBD approach uses a range of features, extracted from the content and meta-data of web pages, to define individual web pages in the set W. A clustering algorithm is then applied, as described above, and the individual web pages in W allocated to a set of clusters K; as noted above the cluster holding w_s will then be identified as K_T. The accuracy of the WBD solution produced is thus subject to: (i) the features that are selected and (ii) the nature of the adopted clustering. The features that may considered with respect to the content based static WBD approach are discussed in Subsection 4.1 below. Two variations of the static content based approach are considered here, (i) the Composite Feature (CF) technique and (ii) the Discriminant Feature (DF) technique; both are discussed in further detail in Sub-section 4.2.

4.1. Feature Representations

There are many features that can be extracted from web pages that can serve to describe content and thus be utilised with respect to the static WBD approach. The features considered with respect to the work presented in this paper are listed below. In each case we consider our motivation for selecting the feature and how the feature may be represented in terms of a binary valued feature space representation. The features considered are categorised into two groups: (i) link (URL) based features and (ii) textual based features.

The link based features were extracted and represented as follows. The HTML content for each page was first downloaded and validated into a well formed Hyper Text Mark-up Language (HTML) Document Object Model (DOM) object. This was done because HTML documents are notoriously malformed and subsequently difficult to parse, for example documents often have missing tags. The steps required to validate the HTML into a well formed document are therefore a necessary precursor to the successfully extraction of link based features. Each link (URL) in each web page was then parsed and extracted from the HTML by splitting it into "words" using the standard delimiters found in URL's. For example the URL `http://news.bbc.co.uk` would produce the set of words $\{news, bbc, co, uk\}$. Non-textual characters were removed (no stop word removal was undertaken). Each word then represented a binary valued feature in a feature vector representation.

The textual based features were extracted from each web page using a html text parser/extractor to extract text as it would be rendered by a web browser. This was deemed to be the same text that a user would use to judge a pages topic/subject. Note that for this purpose non-textual characters were removed, along with words contained in a standard "stop list". Thus the identified textual feature values comprised only what were deemed to be the most significant words. Each word represented a binary valued feature in our feature space.

The link based features that were considered in this work were as follows:

Hyper links The motivation for using hyper links as one of our features is the observation that web pages that are related tend to share many of the same hyper links. The shared links may be other pages in the same website (e.g. the website home page) or significant external pages (most links point to a related set of pages with some common topic).

Image links The use of image links was prompted by the observation that web pages that link to the same images were likely to be related, for example a common set of logos or navigation images could imply a relationship.

Mailto links If a set of web pages includes identical email links (`mailto`) this might also indicate that a relationship exists.

Page Anchor links Page anchors are used to navigate to certain places on the same page, these can be useful with respect to the resolution of the WBD problem as they often have meaningful names. It was conjectured that if the same or related names are used on a set of web pages it could imply related content.

Resource links The motivation for using resource links was that the styling of a page is often controlled by a common Cascading Style Sheet (CSS) which could therefore imply that a collection of pages that use the same style sheet are related.

Script links It was observed that some scripting functions that are used in web-pages can be written using some form of common script file; if pages have common script links then they could also be related.

URL URLs are also likely to be an important factor in establishing whether subsets of web-pages are related or not. A URL is used to request the actual page from the WWW, it contains a whole host of information like server and directory information.

The textual based features considered were then:

Title text It is conjectured that the title text used within a collection of web pages belonging to a common web site is a good indicator of "relatedness".

Body text Another key indicator of web page "relatedness" is content; the text contained in individual WWW pages.

4.2. Static Content Based WBD

This Sub-section presents the two static content based WBD techniques: (i) the Composite Feature

```
Template:
  1: Select feature(s)
  2: Apply clustering algorithm
  3: Extract WBD solution
```

Table 1
Process template for the static approach to WBD.

(CF) technique and (ii) the Discriminant Feature (DF) technique. Both comprises three inter-related processes: (i) feature selection, (ii) clustering and (iii) web site extraction, as illustrate by the process template given in Table 1. The first technique, CF, is the simplest of the two variations. The CF technique operates by simply using a user prescribed individual feature, or combination of features, to represent the web pages in W. The features (see previous section, Section 4.1) are extracted from web pages and represented as a vector space model. The clustering approach described above is then applied and the cluster containing the seed page (w_s) identified as the target cluster ω (the web site of interest) from which the website boundary B can be extracted.

The second technique, DF, is founded on a discrimination method whereby individual features, and combinations of features, are generated and evaluated using a small amount of prelabelled training data (both target and noise web pages). The idea is to learn a "best" set of features in terms of a given WBD problem. A scoring mechanism is used for the evaluation. This is done using the score metric presented later in this paper in Section 7.1. The discrimination process is an iterative one, commencing with single features and then on each iteration adding additional features. In this manner a "combination hierarchy" of features is produced. Given a set of features the same clustering approach as described above can be used to identify ω and consequently B. The process is illustrated in Table 2 with an example set of features $F = \{a, b, c, d, e\}$. All pairings are considered first (line 1) and a best pairing identified (the paring $\{ac\}$ in the example). This best pairing is then used to form a set of candidate triples (line 3), and so on.

5. Structure Based Static WBD

This section presents the second static WBD approach, the structure based static WBD approach, whereby the web graph hyperlink "structure" is used to identify the desired website boundary B. The approach

```
F = {a, b, c, d, e} 5-element features.
  1: Best {ac} : {a,b} {a,c} {a,d} {a,e} {b,c} {b,d} {b,e} {c,d}
     {c,e} {d,e}
  2: Best {acb}   :          {ac,b} {ac,d} {ac,e}
  3: Best {acbe}  :            {acb,d} {acb,e}
  4: Best {acbed} :               {acbed}
```

Table 2
Example hierarchy generated using the feature discrimination technique.

makes the assumption that a website comprises a set of "strongly" connected components compared to the rest of the web graph under consideration. Two variations of the static structure based WBD approach are considered: (i) Hierarchical Graph Partitioning (HGP) and (ii) Mincut Graph Partitioning (MGP). For the first the Newman hierarchical partitioning algorithm [38,39] was used. For the second the Ford Fulkerson algorithm was used which in turn is founded on the *mincut-max flow* theorem described in [23,24] directed at the concept of flow networks. The two variations are described respectively in Sub-sections 5.1 and 5.2 below; both aim to partition the web graph such that strongly connected pages are segmented from the rest, thus producing a WBD solution.

5.1. Hierarchical Graph Partitioning Structure Based Static WBD

This sub-section describes the hierarchical graph partitioning variation of the Structure Based Static WBD approach founded, as noted above, on the Newman hierarchical partitioning algorithm [38,39]; an agglomerative hierarchical mechanism for identifying clusters in graph data according to the similarity of graph vertices. The algorithm produces p graph clusters, where p is determined by the algorithm itself (no need for user input). The hierarchical graph clustering approach is centred around a modularity value Q; a measure of how "good" a given cluster configuration is. The Q value is calculated, according to the interconnectivity of the vertices in each individual cluster, by comparing the connectivity of records within each cluster with respect to the connectivity if the records were connected randomly [40].

More specifically the Q value is calculated as follows. Given: (i) a directed graph $G = (V, E)$ defined as presented above; (ii) a cluster g_i comprised of a set of vertices $g_i = \{v_1, v_2, \dots\}$ (thus $g_i \subset V$); and (iii) a cluster configuration for G ($K = \{g_1, g_2, \dots, g_k\}$). Consider S_i and S_j to be two sets of vertices. A function can be devised, $countEdges(S_i, S_j)$, that counts the number of edges from the set of vertices in S_i to the vertices in S_j. Let E_{ij} equal the set of edges from cluster g_i to g_j in the cluster configuration K representing graph G, then:

$$E_{ij} = \frac{countEdges(g_i, g_j)}{m} \tag{1}$$

Thus E_{ij} represents the set of intra-cluster edges, with respect to g_i and g_j, normalised in terms of the total number of edges m. Note that E_{ii} is the set of intra-cluster edges in a single cluster g_i. Let T equal the total number of edges inside a given cluster configuration k, such that:

$$T = \sum_{g_i=1}^{g_i=k} E_{ii} \tag{2}$$

This value is essentially the number of inter-cluster connections for all clusters. Let a_i be the number of edges within graph G that end in the vertices contained in cluster g_i:

$$a_i = \sum_{j=1}^{j=k} E_{ji} \tag{3}$$

The value a_i^2 is the number of edges in cluster g_i if edges where connected uniformly at random in graph G. This value is simply calculated as $a_i^2 = (a_i)^2$. Given the above, modularity Q is then calculated using:

$$Q = \sum_{i=1}^{i=k} (E_{ii} - a_i^2) \tag{4}$$

Where E_{ii} and a_i^2 are defined as above.

The Newman clustering algorithm is an iterative hierarchical clustering method applied to graph networks (such as social networks); the aim being to generate a set of clusters K, where each individual cluster contains one or more vertices. The input to the algorithm is a configuration K generated by placing each vertex in V in its own cluster. Thus on start up $|K| = |V|$. On each iteration a pair of clusters g_i and g_j, from all possible combinations of clusters in K, is selected to be joined. The selection is made by maximising the Q value (calculated as described above). The algorithm iteratively merges combinations of clusters so that either: (i) the modularity value (Q) is maximised for a cluster configuration, or (ii) a single cluster containing all input vertices V is reached. The resulting cluster configuration produced by the algorithm is thus the cluster configuration that has the highest value Q associated with it.

Using the Newman algorithm a set of clusters K is produced. However, as noted above, the number of clusters is not predetermined, it is derived by the algorithm. The seed page w_s of interest will be located in one of the output clusters, K_T. The aggregation of the remaining clusters, regardless of their number, is then the noise cluster K_N. The aim is to produce a graph partitioning such that K_T consists of as high a number of target vertices and as low a number of noise vertices as possible.

5.2. Mincut Graph Partitioning Structure Based Static WBD

This Sub-section describes the Mincut graph partitioning approach based on the Max-Flow Min-Cut theorem applied to flow networks (recall that flow network theory was also used in [48] for WBD as discussed in Section 2). As noted previously, in graph theory a flow network is a graph of vertices and edges where edges can have a *flow* associated with them, which is an indicator of *material* "flowing" between vertices. Flow networks can be used to model a variety of problems [2,25]: in the context of the work described in this paper they can be used to identify "cuts" (divisions) within a network in order to produce graph partitions [46]. A "minimum cut" is a cut that splits a network into two partitions such that a "minimum" number of edges are cut, thus forming two clusters comprising dense (or at least denser) connections [46]. The minimum cut of a network can be found effectively by applying the Max-Flow Min-Cut theorem [15,23].

The concept of a flow network model is formally described as follows (the description and examples are based on [15]). A flow network is a directed graph $G = (V, E)$, of the form defined previously. An edge between two vertices u and v ($u, v \subseteq V$) is indicated using the notation (u, v). Each edge (u, v) has an associated non-negative capacity c defined as $c(u, v) \geq 0$, and can also have an associated flow f of "material" from u to v ($f(u, v) \geq 0$). The value of the flow cannot exceed the capacity of an edge (see below). This model can be conceptualised as a network of water pipes, whereby the *capacity* is constrained by the diameters of the pipes, thus restricting the amount of water that can flow between any two points. One vertex of the flow network is considered to be the source $s \in V$ and another vertex is considered to be the sink $t \in V$; "material" is then considered to "flow" through the network from the source to the sink. For a graph G to be (typically) considered to be a flow network, some constraints about its structure are imposed: (i) reverse edges are disallowed, if an edge (u, v) exists then there must be no edge (v, u) in the reverse direction; and (ii) self loops are disallowed, vertex v cannot have an edge of the form (v, v) (an edge which points back to its self). A path p in a flow network is simply an ordered set of vertices connected by edges. A path from s to t through (say) two intermediate nodes u and v is represented as $p = \{s \rightarrow u \rightarrow v \rightarrow t\}$. This path thus comprises three edges (s, u), (u, v) and (v, t). The assumption is made that each vertex in V is on some path p connecting s to t. The notation $\rightsquigarrow$ is used to indicate a path from one vertex to another via some unspecified subset of vertices in V, for example $p = \{s \rightsquigarrow t\}$.

Using the above definition of a flow network, a flow f inside a network can be formally defined as follows. The flow f, in a flow network G, is considered to be a value which can be measured between vertices u and v. A flow $f(u, v)$ equals a non negative value such that the following constraints are met:

1. **Capacity Constraint**: The flow from one vertex to another must be non-negative, and must not exceed a given capacity. More formally, for all edges $(u, v) \in E$, a flow f must be ≥ 0 and cannot exceed the capacity $c(u, v)$ of the edge along which it travels.
2. **Flow Conservation**: The total flow into each vertex v, must equal the total flow out of a vertex v, excluding vertex s and t. Formally:

$$\sum_{v \in V} f(v, u) = \sum_{v \in V} f(u, v) \qquad (5)$$

A residual network G_f consists of a network with edge capacities that show how the flow can be changed in a given flow network G. Given a flow network $G = (V, E)$ and a corresponding flow f, the residual network of G induced by f is $G_f = (V, E_f)$. E_f is a set of edges representing the "residual capacity" c_f of G given the flow f. E_f may also contain edges that are not contained in G. This occurs when back flow is admitted in an opposite direction to some existing flow. An edge (u, v) has a positive value in E_f if there is a corresponding available residual capacity $c_f \geq 0$. An augmenting path is a path from vertex source s to sink t in a residual network G_f. The (augmenting) path from s to t must intuitively have a residual capacity along its edges on which an amount of flow can be admitted, other wise the path would not exist in G_f.

A $cut(S, T)$ of a flow network $G = (V, E)$ is a partition of the set of vertices V into two sets S and T, such that $s \in S$ and $t \in T$. The *capacity* of a $cut(S, T)$, the number of edges cut, is calculated using:

$$c(S, T) = \sum_{u \in S} \sum_{v \in T} c(u, v) \qquad (6)$$

Thus the sum of the capacities of edges cut that connect S to T. The net flow of a cut is the sum of flows from S to T minus the flows from T to S. The net flow is calculated using:

$$f(S, T) = \sum_{u \in S} \sum_{v \in T} f(u, v) - \sum_{u \in S} \sum_{v \in T} f(v, u) \qquad (7)$$

Given a flow network $G = (V, E)$ with vertex source s and sink t with a flow f in G, the max-flow/min-cut theorem states that *the maximum flow f in a flow network G between s and t is equal to the capacity of the minimum $cut(S, T)$ of G*. Given a flow f in a flow network G the max-flow/min-cut is found when the following (equivalent) conditions are met:

1. A flow f is a maximum flow in G
2. The residual network G_f contains no augmenting paths
3. The flow in the network $f(G) = c(S, T)$ for some $cut(S, T)$ of G.

(1) states that a flow f needs to be a maximum flow in G. This can be proven using (2). If f is said to be a max flow, while there is an augmenting path in G_f, then there will be a possible increase in flow such that f can be increased and is not a max flow. (3) states that the total flow out of the source minus the total flow into the source equals the capacity of some $cut(S, T)$.

The Ford Fulkerson Algorithm [23] is used to compute the max-flow in a given flow network $G =$

(V, E). The operation of the algorithm is straightforward. For a flow network G, if there exists an augmenting path in the corresponding residual network G_f from s to t, with available residual capacity c_f, flow is admitted along this path. This process is performed iteratively until there are no more augmenting paths in G_f, thus f is a max-flow in G. On commencement of the algorithm the edges have a flow of zero ($f(u, v) = 0$). The algorithm then proceeds to augment flow along available augmenting paths, in a breadth first manner, until no more augmenting paths exist in G_f. The resulting residual network can then be used to make a cut, such that the edges with a maximum residual capacity (max-flow) are cut.

To produce a WBD solution from a graph partitioned into two clusters using the min cut of the graph a similar method was used to that used with respect to the static WBD approaches described above. Namely that the cluster containing the seed node w_s is chosen as the target cluster K_T, and the remaining cluster is identified as the noise cluster K_N. As noted previously the objective is to produce a target cluster that contains a maximum number of target pages for the website of interest, and a minimum number of noise pages.

The WBD approaches presented in this section focused on exploiting only the hyper link structure of W when producing WBD solutions, whilst the WBD approach presented in the previous section (Section 4) exploited only the content of the web pages in W with respect to the WBD problem. The approaches presented in the following section (Section 6) are directed at both the content and hyper link structure of web pages when producing WBD solutions.

6. Random Walk Dynamic WBD

This section presents the third and final WBD approach considered in this paper, the dynamic approach. Recall that in the dynamic context the web data is not available prior to the start of the analysis, the web data is "fetched" as the approach proceeds. The three approaches in this section are based on the Random Walk concept, a method of traversing a graph structure in the form of a succession of random steps from node to node. The idea is to include an element of serendipity (chance discovery) in the process. A random walk is used in this work as a probabilistic method of visiting nodes (web pages) of a web graph in a non-deterministic order. It is argued in this work that such a probabilistic method provides advantages with respect to WBD. Thus in the context of WBD we again conceive of a collection of web pages, in which we wish to find a web boundary centred on some seed page w_s, in terms of a graph $G = (V, E)$ to which a random walk can be applied. Three variations of the proposed random walk dynamic WBD approach are proposed: (i) standard Random Walk (RW), (ii) Self Avoiding Random Walk (SARW) and (iii) the Metropolis Hastings Random Walk (MHRW).

The remainder of this section is organised as follows. Sub-section 6.1 provides an overview of the dynamic WBD approach, whilst Sub-section 6.2 provides details on the graph traversal methods used. The remaining three sub-sections (Sub-sections 6.3, 6.4 and 6.5) present the three variations of the dynamic approach respectively.

6.1. Overview of the Dynamic WBD Approach

The generic dynamic WBD process is described by the template given in Table 3. At the start (line 1) the set of target pages K_T 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 parameters that dictate 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 technique is presented in Section 6.2 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. A key feature of using an incremental clustering algorithm, 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 be added to 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 3), until the system state (cluster centroids) does not change, or some other termination criteria is reached. Due to the differences in 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 3, a step is considered to be a single iteration of the loop from lines 3 to 7. The number of steps required for a dynamic WBD 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 $P$ to visit next;
5:     add $P$ to $K_T$ or $K_N$;
6:     update the process state;
7: until convergence;
8: return $K_T$;
```

Table 3
Template for the dynamic approach to WBD.

6.2. 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. Thus using a probabilistic approach there is an element of choice and unpredictability involved (serendipity). Each of the proposed probabilistic graph traversal techniques is considered in detail in the following three sub-sections in the context of the template presented previously in Table 3. For the evaluation presented later in this paper the operation of the proposed probabilistic traversal techniques are compared with two deterministic techniques: (i) breadth-first Search (BFS) and (ii) depth-first Search (DFS).

6.3. Random Walk (RW)

The Random Walk (RW) method of graph traversal is described by the pseudo code presented in Table 4. As before the input to the algorithm is a start seed page w_s. Given a current page w_c (which at start up will be the seed page) the page to be visited next is selected at random from the immediate neighbours of w_c in a conceptual web graph 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 visit 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.

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

Table 4
Pseudo code for Random Walk (RW)

6.4. 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 of nodes R visited so far. The pseudo code is shown in Table 5. The input to the algorithm is a starting seed page w_s (as in the case of the RW technique described above) and a reset value r. The process commences traversing nodes of the graph in a random fashion, with the exclusion of the set of visited pages contained in R. The process continues until some randomly generated number (between 0 and 1) is greater than $(reset)^j$ (were $reset$ is a user supplied number 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. The process continues until a predefined number of total steps is reached.

```
Algorithm SARW ($w_s$, $r$)
  $K_T = \{w_s\}$; $K_N = \{\}$;
  $reset = r$;
  set $P$ to $w_s$; set counter to one;
  set up the process internal state;
  loop
    increment 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 $R$;
      $P = v_i$;
      add $P$ to $R$;
      add $P$ to $K_T$ or $K_N$;
      update the process state;
      if $P$ added to $K_T$ then
        decrease $j$ (if positive);
      else
        increase $j$;
      end if
      increase $i$;
    until random number $> (reset)^j$;
  end loop counter > MAXITERATIONS
  return $K_T$;
```

Table 5

Pseudo code for Self Avoiding Random Walk (SARW)

6.5. Metropolis Hastings Random Walk (MHRW)

The random traversal of the RW and SARW methods is greatly effected 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 the MHRW is given in Table 6. As in the case of the previous two random walk algorithms the input is a seed page w_s. The process starts by randomly selecting neighbouring nodes to visit that have a degree below a given threshold value calculated using the function $Deg(w)$ which simply returns the degree of the given node. Thus as the value of $Deg(w_c)/Deg(w_{new})$ increases (current page degree and neighbouring page degree), the likelihood of $random\ number > Deg(w_c)/Deg(w_{new})$ decreases, and thus there is a decreasing chance of not moving (staying at the current node).

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

Table 6

Pseudo code for Metropolis Hastings Random Walk (MHRW)

7. Evaluation

From the foregoing seven distinct technique to address the WBD problem have been presented. Four static techniques, where the input data is collected in advance, and three dynamic techniques where this is not the case. The four static techniques were characterised as being either content based or structure based (two variations of each). The two variations of the content based static WBD approach were: (i) Composite Feature (CF) and (i) Discriminative Feature (DF). The two variations of the structure based static WBD approach were: (i) Hierarchical Graph Partitioning (HGP) and (ii) Mincut Graph Partitioning (MGP). The three random walk dynamic WBD variations were: (i) standard Random Walk (RW), (ii) Self Avoiding Random Walk (SARW) and (iii) the Metropolis Hastings Random Walk (MHRW). This section provides an evaluation of these seven WBD approaches.

The standard mechanism for evaluating solutions to the WBD problem is to use hand annotated collections of data extracted from the WWW [26,29,43]. This method was thus also adopted with respect to the work presented in this paper. The four data sets used for the evaluation were 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 between 450 to 500 web pages. The four target web sites were: (i) Chemistry (LivChem), (ii) History (LivHistory), (iii) Mathematics (LivMaths) and (iv) Archaeology, Classics and Egyptology (LivAce). This collection of data sets will be referred to as the Real Web Graph (RWG) collection.

The remainder of this section is organised as follows. Sub-section 7.1 presents the metrics used for the evaluation. The next three sub-sections, Sub-sections 7.2, 7.3 and 7.4, present the results of the evaluation of the three WBD approaches considered in isolation. Sub-section 7.5 then presents a comparative evaluation of the best performing WBD solutions from the foregoing evaluation. The final Sub-section presents a summary of the conducted evaluation.

7.1. Evaluation Metrics

To measure the quality of the proposed WBD approaches performance score was used; the average of the measured accuracy, precision, recall and F-measure (all standard evaluation metrics taken from the domain of data mining). In the case of the dynamic approaches *coverage* and number of steps was used. Coverage is a measurement of the fraction of total target pages C_T or total noise pages C_N that are retrieved at any point in a given clustering configuration K. More formally, for a particular clustering K, the target page coverage is calculated as: $coverage(K) = \frac{M_{TT}+M_{NT}}{C_T}$, where: (i) M_{TT} is the number of target pages in the target cluster and (ii) M_{NT} is the number of target pages in the noise clusters. Similarly for noise pages in K is calculated as: $coverage(K) = \frac{M_{TN}+M_{NN}}{C_N}$, where: (i) M_{TN} is the number of noise pages in the target cluster and (ii) M_{NN} is the number of noise pages in the noise cluster. The number of steps is the number of steps required for a dynamic WBD algorithm to arrive at an appropriate WBD solution as this is a good indicator of "time complexity".

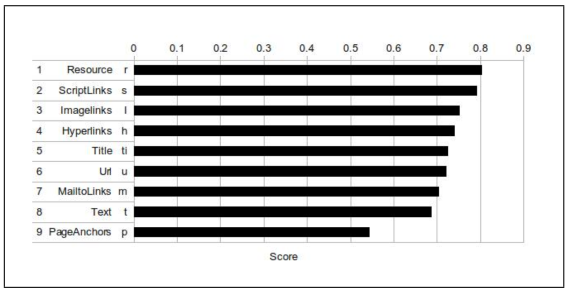

Fig. 1. Performance of the top nine single features, ordered from best performing 1, to worst performing 9.

7.2. Content Based Static WBD Evaluation

The subsection presents the evaluation of the content based static approach. The evaluation was broadly directed at evaluating the most appropriate features, and combinations of features, that could best be used to represent web pages, with respect to WBD. More specifically the objectives were:

1. To analyse the performance of the CF technique in the context of the best features to be used and the WBD solutions generated.
2. To analyse the performance of the DF technique in the context of the WBD solutions generated.
3. To identify the most appropriate clustering algorithm to be used in the context of content based static WBD.

Each is discussed in further detail in the following three sub-sections.

7.2.1. Analysis of Composite Feature (CF) Technique

This subsection reports on the evaluation directed at determining the most appropriate set of features to be used in conjunction with the CF technique. Recall, that this technique works by selecting a feature or combination of features for WBD as presented previously in Sub-section 4.2. A sequence of experiments was undertaken using single features, pairs of features and triples of features. The results are presented in Figures 1 and 2. Figure 1 lists the nine top performing single features ordered according to their performance score. From the Figure it can be seen that the top performing features are resource, script and image links. These features are intuitively indicative of the authors intent to express relationships in terms of a website. Resource and script links are used to express an underlying similar "look and feel" of pages in the same collection. It is very common for shared scripts, or blocks of code providing functionality, to be stored once, in a com-

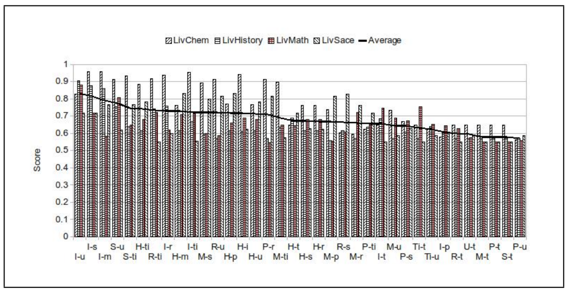

Fig. 2. The performance of feature pairings, ordered from best performing 1, to worst performing 36.

mon location, and referred to many times by different resources. A similar practice is used in the styling of pages, which may link to common CSS or shared design elements. From Figure 1 the features that do not seem to provide a good WBD solution are: page anchors, text and "mailto" links.

Figure 2 shows the best results obtained when considering pairs of features. In the figure the x-axis lists the 36 possible pairings ordered according to their performance score. It is interesting to note that the best performing pairs were found to be made up of combinations of a strong and weaker single features. The results demonstrate that when using pairings better results are obtained than when using single features. With respect to the double combinations, script, image and resource links appear in the top performing combinations. In contrast to what might have been predicted, combining the top performing single features does not necessarily yield top performing combinations. The features that performed the best with respect to the single feature experiments, do however, seem to perform much better when combined with features that appear lower in the performance list generated with respect to the single feature analysis. The most effective lower performing single features, when used in combination with higher performing single features, appear to be: url, mailtolinks and title features.

The results from the triple feature combinations, not shown here, were also evaluated. It was observed that the results obtained were consistent with the results obtained using pairs of features; under performing features appear to increase the performance of the high performing features when used in combination. In conclusion the top performing features are resource, script and image links.

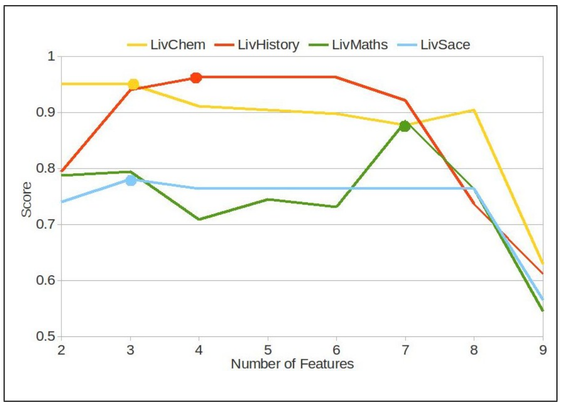

Fig. 3. The best WBD performance score for each feature combination size, the highest scoring combination for each data set is highlighted by a dot.

7.2.2. *Analysis of Discriminant Feature (DF) Technique*

This sub-section presents the results obtained from the evaluation conducted to determine WBD solutions using the DF technique. Recall that the CF technique uses a predefined set of features and, as demonstrated in the foregoing sub-section, there is no best set of features guaranteed to produce a best WBD solution. The DF technique addresses this issue by using a small amount of training data whereby a best set of features can be learnt. The DF technique iteratively combines features, starting with a single feature, until a best set of features is arrived at which can then be used to derive a WBD solution in the same manner as in the case of the CF technique. On each iteration, the combination that provides the biggest increase in performance score with respect to the WBD solution is chosen to be the "feature set so far".

The training data that is used for the DF technique was a labelling of pages within the website boundary solution for each of the input data sets. Recall that as the DF technique progresses this labelled data was used to evaluate the best combination of features, such that a best final feature combination was derived which could then be used to generate a WBD solution. On each iteration the resulting WBD solutions are compared to the known training data solution and the best performing feature combination selected for use in the next iteration. This iterative process builds combinations of features. The results obtained are given in Table 4 which shows the best feature combination, and corresponding performance score, from each iteration as the DF algorithm progresses (with respect to the RWG evaluation data sets). From the table it can be

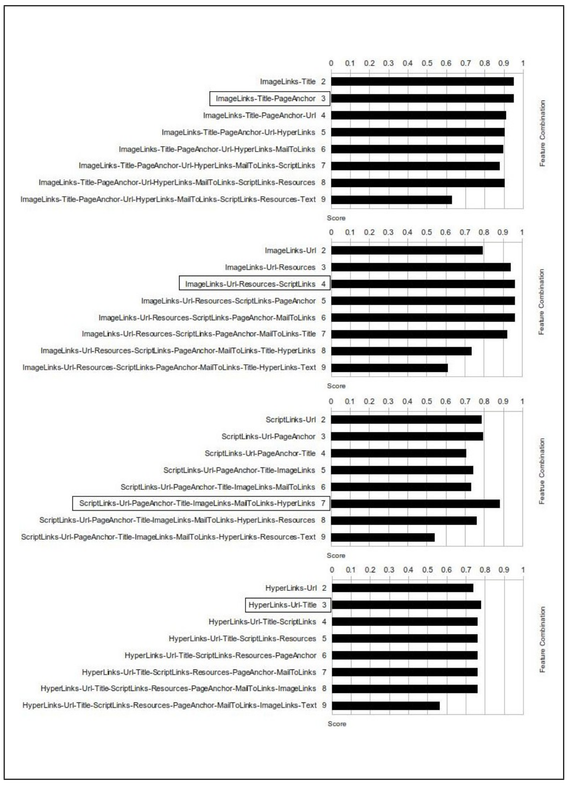

Fig. 4. The best performing feature combinations and performance scores for each RWG data set. The highest scoring feature combination in each case has been highlighted.

seen that, as expected, the highest performing feature combinations contain features that were identified as high performing using the CF approach. Namely resource, script and image links.

Figure 3 shows the same results as presented in Figure 4 but in the form of line graphs. In the figure 4 performance score is listed on the y-axis and size of feature combination, from 2 to all 9, along the x-axis. From the figure it can be seen that by using all features in combination produced the worst performing WBD solution with respect to all four test data sets (probably because the representation got to convoluted).

The results presented in Table 4 and Figure 3 corroborate the results presented in Sub-section 7.2, namely that the best feature combinations are those that comprise both at least one high and at least one mid performing single feature. It can also be concluded that there always exists a combination of features that gives a highest WBD performance score, and that the size of this best combination is somewhere between three and seven.

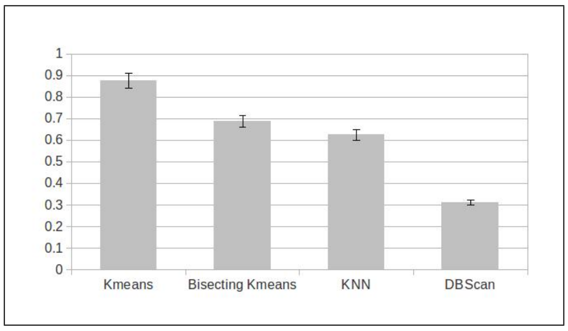

Fig. 5. Clustering algorithms ranked according to average WBD performance score across the RWG data set using single, double and triple feature combinations.

7.2.3. Clustering Algorithms

In this sub-section the results from experiments conducted to identify the most appropriate clustering algorithm to be used are presented. Four clustering algorithms were considered as follows:

1. Kmeans (Centroid based) [36]
2. Bisecting Kmeans (Hierarchical based) [54]
3. DBScan (Density based) [22]
4. KNN (Centroid based) [19]

Figure 5 shows the ranked average WBD performance score, with respect to each clustering algorithm, with respect to the four datasets considered and using single, double and triple feature combinations. Both the Kmeans clustering algorithm and the bisecting Kmeans algorithm require the number of clusters to be pre-specified. In contrast, KNN and DBScan do not use such a value, instead they rely on a user provided similarity threshold value to determine a cluster configuration. In the evaluation presented in this section the parameters for each algorithm were systematically varied and the best performing output reported. The best performing parameter in each case was as follows: (i) for Kmeans the best number of clusters was $k = 3$, (ii) for bisecting Kmeans $k = 4$ produced the best results, (iii) for KNN the most appropriate user defined threshold was found to be $c = 5$, and (iv) for DBScan the most appropriate distance value was found to be $d = 0.3$ and the minimum number of points $m = 5$. Inspection of Figure 5 shows that the Kmeans algorithm produced the best overall performance, arguably because of its robust operation, allowing it to best adapt to the various feature spaces.

The main problems with applying clustering algorithms to the different kinds of feature space used in this work is the so called "curse of dimensionality"; whereby, as the number of dimensions increases it becomes increasingly computational intensive to evaluate distance functions with respect to the feature space. The fact that the feature space becomes very sparse as the number of dimensions increases can be considered to be a contributing factor to the poor performance when all features are used.

7.3. Structure Based Static WBD Evaluation

The evaluation of the two variations of the structure based static WBD approach is presented in this sub-section: (i) Hierarchical Graph Partitioning (HGP) and (ii) Mincut Graph Partitioning (MGP). Recall that the first was based on Newman hierarchical graph partitioning (see Sub-section 7.3.1), while the second was based on the minimum cut algorithm which in turn was based on min-cut max flow theorem (see Sub-section 7.3.2). The objective of the evaluation presented in this section is to measure the performance of the two structure based approaches (HGP and MGP) with respect to the WBD solutions produced. The evaluation results for each technique are presented and discussed in further detail in the following two sub-sections. The results, in both cases, indicated that the structure based static WBD approach has potential in terms of producing high quality WBD solutions. The results obtained also demonstrated that the connectivity between vertices of a single website boundary are in fact encoded in the underlying web graph structure.

7.3.1. HGP evaluation

The evaluation of the HGP technique was conducted by measuring the performance of the Newman HGP algorithm using a sequence of web graph snapshots varying in size from 50 to 450 vertices increasing in steps of 50.The pages contained in each snapshot were collected using a breadth first crawl from a seed (home) page w_s, thus the ratio of noise to target pages increased as the number of vertices considered increased. The aim was to mimic the way that a web crawler might collect data.

The results are presented, in terms of performance score as before but also precision and recall, in Figure 6. From the figure it can be seen that as the snapshot size increased recall also increases while precision decreased. The increase in recall indicated a corresponding increase in the number of target web pages grouped with the seed page. This in turn demonstrated that, as the snap shot size increased, the Newman algorithm was increasingly partitioning the graph in such a way that target pages were not being allocated to too

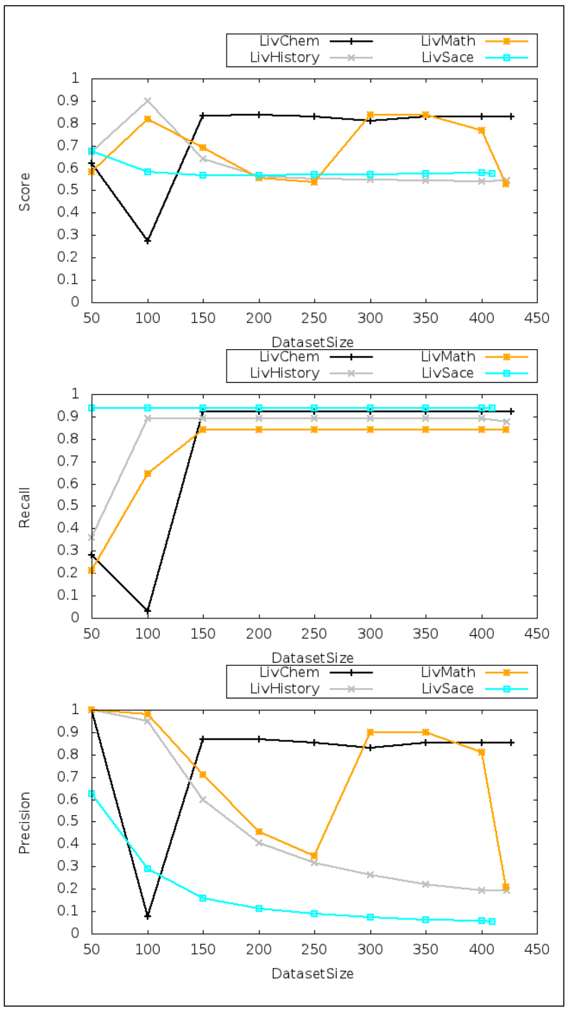

Fig. 6. The results obtained using the HGP technique on the RWG datasets (LivChem, LivHistory, LivMath and LivSace) with varying snapshot sizes.

many differing clusters, but were in fact being mostly grouped together (the desired result). The decrease in precision indicated that the number of noise pages grouped with the seed page increased as the snapshot size increased. In other words as the snapshot size increased the HGP technique was increasingly partitioning the graph in such a way that pages belonging to the website of interest (defined by w_s) were grouped together but with increasing amounts of noise pages.

The results presented in Figure 6 also indicate that the most appropriate WBD solution associated with each data set is not produced using a single particular snapshot size (number of vertices/pages). What is consistent is that the best WBD solution is arrived at using smaller sized snapshot than larger sized snapshots.

This intuitively indicates that reducing the amount of noise improves the WBD solution using the proposed HGP technique. What this suggests in terms of the deployment of the technique is that an improved crawling strategy, that serves to reduce the number of noise pages included in the snapshot, such as that employed by the random walk techniques also considered in this paper, will be beneficial in the context of deriving WBD solutions. In the figure it can also be observed that there is some volatility in the results. The precision values of LivChem and LivMaths drops for certain data set sizes, as does the score in some cases. This is due to the nature of the graph partitioning algorithm working on the hyperlink structure of varying sized snapshots. These snapshot sizes could contain a vastly different hyperlink structure if they contain or omit a high degree page (has many links to other pages). This then then cause the HGP technique to segment the graph in a very different set of partitions. Due to this there is volatility in the results, as shown in figure 6.

7.3.2. MGP Evaluation

The evaluation of the proposed MGP technique was directed at analysing the Mincut algorithm's ability to segment a graph using the concept of flow networks so as to generate a WBD solution. Recall that the Mincut algorithm uses a source s and a sink t vertex within the graph to conceptualise flow, and then makes a cut of the network at the "bottle necks". For the evaluation of the MGP technique the source s and sink t were iterated over all the possible vertices in the RWG data sets. Thus for a graph containing n vertices this involved $n \times n$ combinations (s and t can be at the same vertex). The evaluation was therefore exhaustive.

The results are presented in Figure 7. The figure shows the performance scores for each of the WBD solutions along the y-axis and the identifier for each possible vertex combinations of s and t along the x-axis. The x-axis is ordered according to the source/sink combination, in BFS order from the seed node, which is displayed at point 0 on the x-axis. From the figure it can be observed that for each of the datasets there exists at least one WBD solution that has a performance score of > 0.9. Analysis of the top performing results (after WBD solutions have been generated) reveal that best results were obtained when the sink vertex t was selected from amongst the target website vertices (pages within the website boundary). There was no significant difference in performance regarding whether the source vertex s was selected from amongst the target or noise vertices. Thus, in practice, when deploying the MGP technique w_s should be selected as the sink vertex t; this is contrary to intuitive thinking where the source would be identified as the seed page.

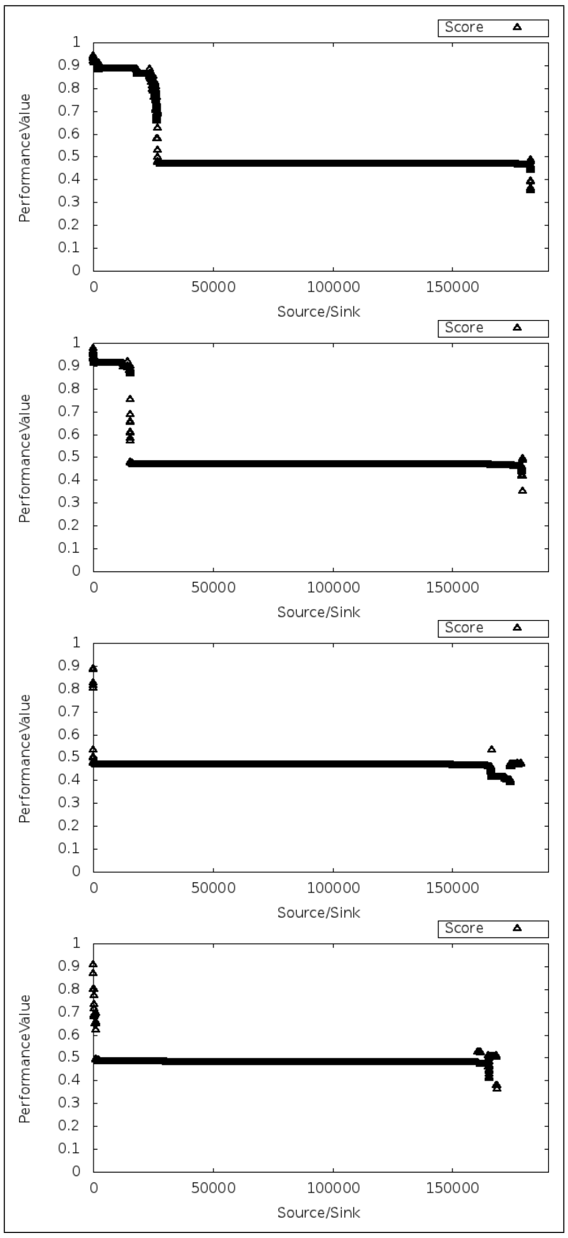

Fig. 7. WBD performance scores using the MGP technique with iterating source s and sink t vertices. (performance score on y-axis combination identifier on x- axis).

7.4. Random Walk Evaluation

This section presents a comparative evaluation of the three proposed dynamic probabilistic random walk techniques for WBD: (i) RW, (ii) SARW and (iii) MHRW. The reported evaluation was conducted by comparing the proposed techniques with two deterministic techniques, namely Breadth First Search (BFS) and Depth First Search (DFS). Recall that the proposed dynamic approaches are based on random walk traversals of the web graph, combined with incremental clustering of web pages to produce a WBD solution. The objectives of the evaluation were:

1. To compare the operation of the considered techniques in the context of the most appropriate features with which to represent web pages. For this purpose five single features were considered: body text, title text, script links, resource links and image links. These features were chosen because they were the top performing features identified with respect to the analysis presented in Section 7.2 above. Note that, for simplicity, only single features (in contrast to combinations of doubles or triples) were used with respect to the evaluation results presented here.
2. To compare the performance of the five techniques in terms of runtime and coverage.
3. To compare the operation of two different clustering algorithms, the Incremental Kmeans (IKM) algorithm[41,47] and the Incremental Clustering Algorithm (ICA) [35], in the context of the techniques under consideration.

With respect to the first objective the results are presented in Figure 8. The results demonstrate that the title feature exhibits the best performance in terms of the proposed dynamic WBD solutions. The performance of the title feature in the dynamic context contrasts to that of the static context (see Sub-section 7.2). The reason why the title feature performed well in the context of the dynamic WBD approaches, whereas it did not perform so well in the context of the static CF approaches, was found to be due to the nature of the incremental clustering and web crawling used. The dynamic approaches were found to be significantly influenced by the order in which web pages were received which in turn was a consequence of the adopted web crawl strategy used.

The advantages offered by usage of the title feature were: (i) it is written by the author/publisher to convey some summary of the page that is quickly digestible by users, (ii) it is focussed on a particular subject, (iii) it is written in a concise manner designed to "round-up" the subject of the www page, (iv) it contains much less noise than (say) the body text of a www page, (v)

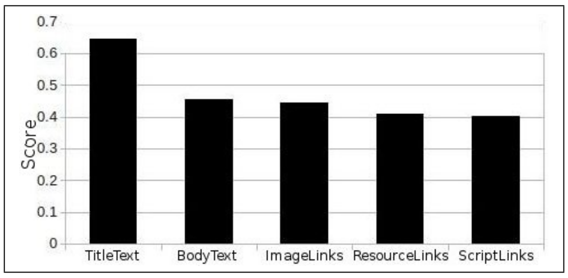

Fig. 8. Average WBD Performance score for dynamic techniques using Body text, Title text, Script links, Resource links and Image links as the feature feature representation.

it is short (few words are typically used) and (vi) it is focussed on a particular subject. As a result usage of www page titles as a feature with which to represent www pages offers good similarity measurement between related pages. The image, script and resource links performed the worst in the dynamic context because they did not provide for effective comparison to allow target and noise pages to be distinguish (at least in the dynamic context). It is suggested that the nature of the dynamically changing feature space is the reason for the difference in performance between the static and dynamic approaches founded on the same features.

In the context of the evaluation with respect to the second objective for the dynamic approach to WBD only the title feature was considered (because the previous experiments had indicated this provided the best performance). The runtime results are presented in Table 7 where the rows represent the five dynamic techniques considered. The first three columns give the K_T, K_N and total coverage results obtained on termination. The fourth column lists the best performance score (average are given in Figure 8), whilst the fifth and six columns give the average runtime per step in seconds and the overall runtime in terms of the number of steps required to obtain a best performance. From the table it can be seen that the best performing technique in terms of performance score was the proposed MHRW technique (performance score of 0.791). The BFS and DFS techniques produced the best run times they traverse the graph structure in linear time.

Figure 9 gives plots of the average coverage, target coverage and noise coverage as the dynamic techniques proceed (number of steps along x-axis). The figure shows how the coverage values change over time. From the figure it can be seen that the graph coverage using the RW reaches a high level earlier on in the process in comparison with SARW and MHRW.

MHRW is the slowest at covering the total graph although the MHRW technique covers noise pages at a lower rate whilst maintained a high coverage of target pages. Note that the randomised traversal using RW, SARW and MHRW are all subject to the underlying structure of the graph, the hyperlink structure can be complex, and unpredictable. The MHRW approach deals well with the complex structure and produces the best WBD score *** NOT ACCORDING TO FIGURE 8 ***.

Table 7

WBD performance for the dynamic techniques considered, using the RWG data sets, ordered according to performance score.

	Coverage				Time(ms)	Total
	Target	Noise	Total	Score	/Steps	Steps
MHRW	0.941	0.768	0.788	0.791	4.740	25000
BFS	1.000	1.000	1.000	0.717	314.801	842
DFS	1.000	1.000	1.000	0.714	320.561	842
RW	0.960	0.992	0.988	0.632	5.957	25000
SARW	0.890	0.869	0.871	0.506	5.611	25000

From the foregoing reported evaluation It was concluded that the MHRW approach was the best performing in terms of performance score. The approach produced an acceptable WBD solution, while at the same time reducing the amount of noise pages covered. This effectively reduces the resources required to produce a WBD solution.

The third and final objective of the evaluation directed at the dynamic approaches was to compare the usage of IKM and ICA clustering algorithms in the context of dynamic WBD. The average WBD performance score of the BFS and Random Walk (RW) approaches, using ICA and IKM, are shown in Figure 10. The BFS and RW techniques was chosen as a comparison of a probabilistic (RW) and deterministic (BFS) technique. The Figure 10 show the "history" of the graph traversal using both clustering algorithms as the pages of the graph were visited step by step. The history of the walks show how the performance of the WBD solutions changes as the walk proceeds. The BFS technique of traversing the graph produced an ordering of pages such that both the ICA and IKM algorithms effectively converge in under 2k steps. The RW method of traversal, however, produces an ordering of pages such that the pages were constantly being randomised therefore the algorithms do not converge within 10k steps. From the figure it can be seen that the score of the WBD solution produced by the BFS traversal using IKM algorithm outperform that of the ICA (Figure 10), this indicated that the IKM algorithm groups pages in a more beneficial way compared to ICA in terms of WBD.

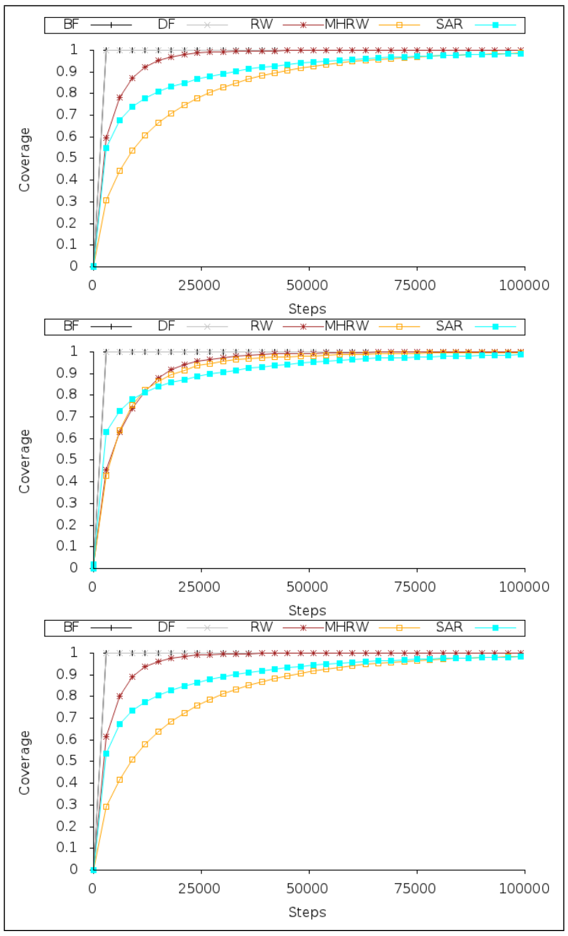

Fig. 9. The average total coverage (top), target coverage (middle) and noise coverage (bottom) for the five dynamic WBD techniques using the RWG data sets and the title feature.

7.5. *Comparison of proposed approaches*

This section presents an overall comparison of the three WBD approaches presented in this paper. Recall that the above reported evaluations were undertaken in terms of WBD performance score and coverage and run time. The WBD performance score was used to measure how representative the website boundary solution was with respect to each approach. Graph coverage was used with respect to the dynamic approach and was used to measured the amount of data gathered (noise and target pages) to produce a WBD solution,

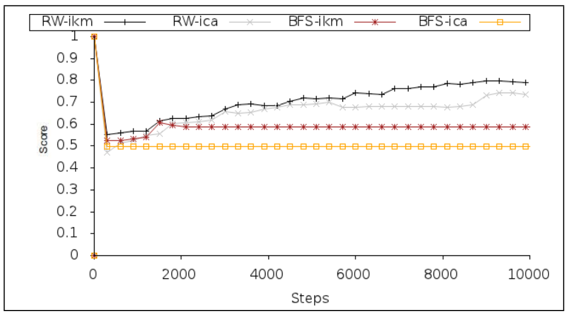

Fig. 10. The average WBD performance score of BFS and RW techniques using either IKM or ICA clustering.

which in turn provided an indicator of the potential resource cost associated with requesting, downloading and pre-processing data from the web. Run time was used to measure how long each technique took to produce a WBD solution. The amount of time taken when analysed using the number of steps was used to further comparatively evaluate the techniques presented in this work. The foregoing evaluation also considered the effect of using various parameters and variations. For the overall comparison reported on in this Sub-section the best performing variations of each of the proposed approaches was used as follows:

– Static Technique: DF
– Static Technique: MGP
– Dynamic Technique: MHRW using IKM

As before the evaluation was again conducted using the RWG data collection.

The results are presented in Figure 11. In the figure three bar graphs are presented, performance score, target coverage and noise coverage with respect to the RWG data sets. From the figure it can be seen that the static approaches performed much better than the dynamic approaches in the context of both performance score and coverage. The highest performing WBD solutions were produced using the DF and MGP techniques. This outcome would be expected as the static approaches have access to all the data apriori (assuming a large enough sample); the static approaches can thus use all the data to make relative decisions on what pages are included with respect to the WBD problem. In the dynamic context it is not the case that all data is known in advance, only partial (step-by-step) data is used to produce a WBD solution, but as the process proceeds we can expect the approach to get better. Therefore the decisions made using the dynamic approach are made relative to what is known about the partial data. It can be argued that the dynamic approach produces an adequate WBD solution while at the same time maximising the number of target pages gathered (Figure 11 Middle) and reducing the number of unwanted noise pages (Figure 11 Bottom). If a standard cost is to be associated with each web page, which is consistent with the requesting, downloading and pre-processing, then it can be seen from the comparative evaluation that the dynamic approach provides a much lower cost WBD solution, while still maintaining an adequate WBD performance. This is shown by the higher WBD score, and lower noise coverage (which is associated with a higher cost).

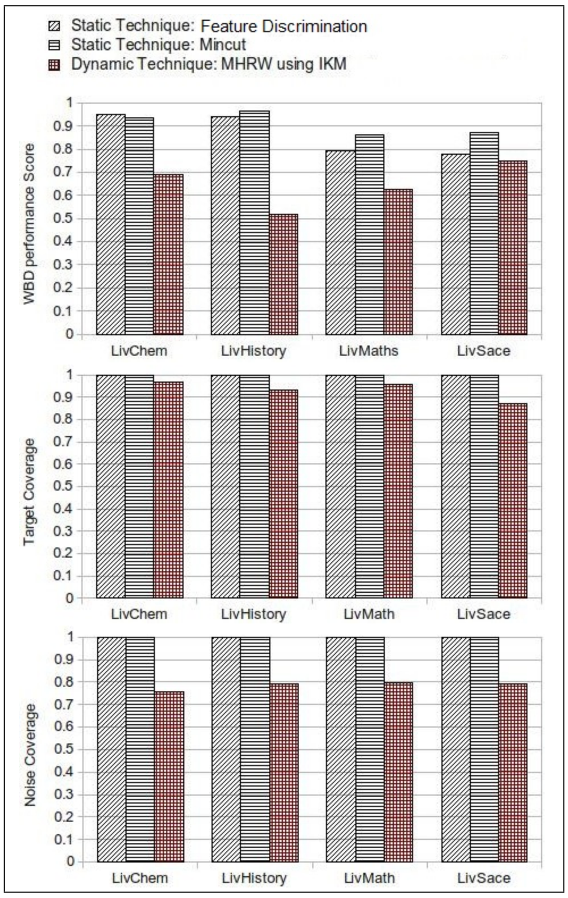

Fig. 11. Comparison of the highest performing approaches for the production of WBD solutions. Sub-Figure (Top) shows WBD performance score, (Middle) Target and (Bottom) Noise coverage.

8. Conclusion

In this paper a number of approaches/techniques for addressing the WBD problem have been proposed and evaluated. The significance of WBD is that it has application with respect to automated website archiving and digital preservation to give two examples (see Section 1). The work presented in this paper considered both static and dynamic approaches for resolving the WBD problem. The distinction is that in the static context all the www pages to be considered are first downloaded. This has the consequent advantage that decisions on what groupings to classify web pages into, either contained within a website (target pages) or outside of the website (noise pages), can be made based on all data apriori. In the dynamic context only portions of the Web are considered in turn, the analysis is thus conducted in a step by step manner. This has the consequent advantage that the number of irrelevant (noise) pages that are requested, stored and processed can potentially be reduced. The evaluation of the proposed approaches presented in this work concentrated on an important challenge with respect to WBD which is the resource cost required to request, store and process web content (see Section 1). This was conducted by evaluating the noise and target page coverage. The experimental analysis and evaluation of the static approaches was used to inform the dynamic approach to WBD. The static feature combination and graph partitioning analysis informed the dynamic approaches. The dynamic approach exploited the structural properties of the web graph while also using the content based attributes of web pages with respect to WBD.

The four static approaches, CF, DF, HGP and MGP were considered (and evaluated): content based (CF and DF) and structure based (HGP and MGP). The evaluation of the content based approach concentrated on the feature representation of web pages. The evaluation concluded that the most appropriate features to be used to represent a web page in the context of providing a good WBD solution were: image, script and resource links. The most appropriate clustering algorithm was shown to be Kmeans. The structure based approach concentrated on the structural properties of the hyperlinks in the web graph for the purpose of WBD. The evaluation concluded that hyperlinks could be successfully exploited with respect to WBD. The evaluation of the static approaches indicated that the MGP technique, which used the max-flow min-cut theorem, was the most effective structure based WBD approach.

The three dynamic probabilistic random walk techniques were considered: (i) RW, (ii) SARW and (iii) MHRW. In the evaluation the operation of these techniques was compared to two deterministic techniques, namely Breadth First Search (BFS) and Depth First Search (DFS). From the evaluation it was concluded that the MHRW approach was the best performing in terms of performance score. The approach produced an acceptable WBD solution, while at the same time reducing the amount of noise pages covered. This effectively reduces the resources required to produce a WBD solution.

Overall this work argues that the dynamic approach using Random Walk graph traversal and incremental Kmeans clustering provides for both an effective WBD performance and a minimisation of the number of irrelevant noise pages considered, which is increasingly important if we wish to scale the proposed dynamic approach to much larger websites. In particular the Metropolis Hastings Random Walk (MHRW) graph traversal method, coupled with the title feature representation and the incremental Kmeans algorithm (IKM), was the best performing WBD method.,

References

[1] S. Abiteboul, G. Cobena, Julien Masanès, and G. Sedrati. A First Experience in Archiving the French Web. In *Proceedings of the 6th European Conference on Research and Advanced Technology for Digital Libraries*, volume 2458 of *Lecture Notes in Computer Science*, pages 1–15, London, UK, 2002. Springer-Verlag.

[2] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] L Becchetti, C Castillo, D Donato, S Leonardi, and R Baeza-Yates. Link-based characterization and detection of web spam. In *Proceedings of the 2nd International Workshop on Adversarial Information Retrieval on the Web*, pages 1–8, 2006.

[9] A. Benczúr, K. Csalogány, and T. Sarlós. Link-based similarity search to fight Web spam. In *Adversarial Information Retrieval on the Web*, pages 1–8, Seattle, Washington, USA, 2006.

[10] 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.

[11] A. Z Broder. Graph structure in the Web. *Computer Networks*, 33(1-6):309–320, jun 2000.

[12] A. Brown. *Archiving Websites: a practical guide for information management professionals*. Facet Publishing, London, England, 2006.

[13] K. Chellapilla and D. M Chickering. Improving Cloaking Detection using Search Query Popularity and Monetizability. In *Proceedings of the 2nd International Workshop on Adversarial Information Retrieval on the Web*, pages 17–24, Seattle, WA, aug 2006.

[14] 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, oct 2003. IEEE Computer Society.

[15] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, Cambridge, Massachusetts, third edition, sep 2009.

[16] M. Deegan and S. Tanner. *Digital Preservation*. Digital futures series, 2006.

[17] 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.

[18] P. Dmitriev and C. Lagoze. Automatically Constructing Descriptive Site Maps. In *Frontiers of WWW Research and Development, APWeb 2006*, pages 201 – 212. Springer Berlin, Heidelberg, 2006.

[19] M. H. Dunham. *Data Mining: Introductory and Advanced Topics*. Prentice Hall PTR Upper Saddle River, NJ, USA, 2002.

[20] 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.

[21] 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.

[22] M Ester and HP Kriegel. A density-based algorithm for discovering clusters in large spatial databases with noise. In *2nd International Conference on Knowledge Discovery and Data Mining*, pages 226–231, 1996.

[23] L.R. Ford and D.R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8(3):399—-404, 1956.

[24] L.R. Ford and D.R. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, NJ., 1962.

[25] A. Gibbons. *Algorithmic Graph Theory*. Cambridge University Press, 1985.

[26] K. Golub and A. Ardö. Importance of HTML structural elements and metadata in automated subject classification. In *In Proceedings of the 9th European Conference on Research and Advanced Technology for Digital Libraries*, pages 368–378. LNCS, 2005.

[27] 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.

[28] M. Keller and H Hartenstein. Mining Taxonomies from Web Menus: Rule-Based Concepts and Algorithms. *Web Engineering*, 2013.

[29] 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, apr 2012. ACM Press.

[30] O.-W. Kwon and J H Lee. Text categorization based on k-nearest neighbor approach for web site classification. *Information Processing and Management*, 39.(1):25–44, jan 2003.

[31] T. Lavergne, T. Urvoy, and F. Yvon. Detecting Fake Content with Relative Entropy Scoring. In *International Workshop on Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection*, volume 377 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

[32] B. Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications)*. Springer, 2nd edition, 2011.

[33] Nan Liu and C Yang. Extracting a website's content structure from its link structure. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 345–346, New York, NY, USA, 2005. ACM.

[34] Nan Liu and C. Yang. Mining web site's topic hierarchy. *Special interest tracks and posters of the 14th international conference on World Wide Web*, page 980, 2005.

[35] Y. Liu, Y. Ouyang, H. Sheng, and Z. Xiong. An Incremental Algorithm for Clustering Search Results. In *Proceedings of the 2008 IEEE International Conference on Signal Image Technology and Internet Based Systems*, pages 112–117, Washington, DC, USA, nov 2008. IEEE Computer Society.

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

[37] G. Mishne, D. Carmel, and R. Lempel. Blocking Blog Spam with Language Model Disagreement. In *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web*, Chiba, Japan, may 2005.

[38] M. Newman. Detecting community structure in networks. *The European Physical Journal B-Condensed Matter . . .*, 2004.

[39] M. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69(6):5, jun 2004.

[40] M. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2), feb 2004.

[41] D T Pham, S S Dimov, and C D Nguyen. An Incremental K-means algorithm. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 218(7):783–795, jan 2004.

[42] 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, feb 2010.

[43] E. M. Rodrigues, N. Milic-Frayling, and B. Fortuna. Detection of Web Subsites: Concepts, Algorithms, and Evaluation Issues. In *Web Intelligence*, pages 66–73. IEEE Computer Society, 2007.

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

[45] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, nov 1975.

[46] S.E. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

[47] D. Sculley. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, page 1177, New York, New York, USA, apr 2010. ACM Press.

[48] P. Senellart. *Website Identification*. Masters thesis dea internship report, Université Paris XI, Orsay, France., sep 2003.

[49] P. Senellart. Identifying Websites with Flow Simulation. Technical report, Gemo, INRIA Futurs., Orsay, France., 2005.

[50] 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.

[51] T. Urvoy, E. Chauveau, P. Filoche, and T. Lavergne. Web Spam Challenge 2007: France Telecom R&D Submission. may 2007.

[52] T. Urvoy, E. Chauveau, P. Filoche, and T. Lavergne. Tracking Web spam with HTML style similarities. *ACM Transactions on the Web (TWEB)*, 2(1):1–28, feb 2008.

[53] T. Urvoy, T. Lavergne, and P. Filoche. Tracking Web Spam with Hidden Style Similarity. In *2nd International workshop on Adversarial Information Retrieval on the Web*, pages 25–31, Seattle, Washington, USA, 2006.

[54] Ying Zhao and George Karypis. Clustering in Life Sciences. *Functional Genomics: Methods and Protocols, M. Brownstein, A. Khodursky and D. Conniffe (editors).*, 2003.