# Web-Site Boundary Detection

Ayesh Alshukri, Frans Coenen, and Michele Zito

Dept. of Computer Science, The University of Liverpool, Liverpool L69 3BX, UK
{a.alshukri,coenen,michele}@liverpool.ac.uk

**Abstract.** Defining the boundaries of a web-site, for (say) archiving or information retrieval purposes, is an important but complicated task. In this paper a web-page clustering approach to boundary detection is suggested. The principal issue is feature selection, hampered by the observation that there is no clear understanding of what a web-site is. This paper proposes a definition of a web-site, founded on the principle of *user intention*, directed at the boundary detection problem; and then reports on a sequence of experiments, using a number of clustering techniques, and a wide range of features and combinations of features to identify web-site boundaries. The preliminary results reported seem to indicate that, in general, a combination of features produces the most appropriate result.

**Keywords:** Web-site definition, Web-page Clustering, Web Archiving.

## 1 Introduction

As the World Wide Web has grown in size and importance as a medium for information storage and interchange, the problem of managing the information within it has assumed great significance. In particular, there has been a lot of interest, recently, in working with whole web-sites, and other compound web-objects rather than single web-pages [5,17,18]. The detection of web-site boundaries is an important aspect with respect to many applications such as web archiving, WWW information retrieval and web spam detection. The process of archiving web content is a non trivial task [6, page 82]. The target information may be contained in just a few HTML files, or a very complex web application [1]. Identifying the boundary of a web-site can automate the choice of pages to archive. Studying the world-wide web at web-site level rather than web-page level may also have useful applications [3].Documents can be represented by multiple pages on the web [5]. Thus, sometimes, it is not reasonable to study attributes like authorship at page level. A web-site entity may be reorganised at the site owners control, as pages and links appear/disappear on an infinite basis [14]. This characteristic implies the separate study of inter and intra site links. The accessibility of content on the web [4], assuming content is fully accessible from within a site (navigation between pages all of the site) can focus on connectivity between sites. Finally, the study of the web using statistical analysis of web-pages maybe skewed due to the simplicity of rapid and dynamic generation.

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 529–543, 2010.

The identification of the boundaries of a web-site can be a relatively simple task for a human to achieve. When traversing the web, navigating from one web-page to another, the detection of a particular web-sites boundaries is done by a human recognising certain attributes from these pages or closely related content. The set of attributes from a page is usually common to each of the pages within a web-site, this is also true of the topics which are closely related or about one theme. The features that a user can recognise to determine similarity between pages can be the style and layout of the page, including; colours, borders, fonts, images and the positioning of these items. Also the content covered, including topic or topics displayed in various sections or sub section within the same page or spread across pages. Although this is all fairly obvious to humans, the boundary detection task is far from trivial for a machine. This paper tries to overcome such difficulties by proposing a data mining approach to the web-site boundary identification problem.

The identification process is hampered by the lack of a clear, general, and useful definition of what a web-site is [2,3]. The term is often used either informally (for instance when investigating the sociological impact of the web [10]), or in rather specific ways. The simplest option is to state that a web-site is defined by the machine it is hosted on. However, several web-sites may be hosted on the same machine (e.g. `http://www.member.webspace.virginmedia.com` has content by many authors), alternatively a single web-site may span several machines (for example the INRIA's web-site has content on domains `www.inria.fr`, `www-rocq.inria.fr`, `osage.inria.fr`, etc). A web-site may also comprise several sub web-sites. To apply data mining techniques to the web-site boundary detection problem, in the context of applications such as web archiving, requires some definition of a web-site. This is one of the issues addressed in this paper. The second issue has to do with the nature of the web-page features that should be included in a feature vector representation that permits the application of data mining techniques to identify web-site boundaries. From the above it is clear that URL alone is not sufficient. Intuitively content alone would also not be sufficient given that any web-site can be expected to link to other sites with similar content. In this paper we present a number of experiments investigating which features are the most appropriate to aid the identification of web-site boundaries.

Given a collection of web-pages, represented in terms of a set of features, we can attempt to identify boundaries either by processing the collection in a static manner or a dynamic manner (by "crawling" through it). The first option is considered in this paper. In the static context clustering techniques may be applied so as to distinguish between web-pages that belong to a given web-site and web-pages that do not belong to the site.

The contributions of this paper may thus be summarised as follows;

1. A definition of what constitutes a web-site in the context of web-site boundary identification.
2. A report on a sequence of preliminary experiments, conducted using a number of different web-page features, and a combination of features, to determine the most appropriate features for boundary identification.

3. A report on the use of a number of different clustering techniques to identify the most appropriate for web-site boundary identification.

Note that the most appropriate clustering technique and web-page model combination, as will be demonstrated, is that which most accurately generates the known clusters present in a number of "test" input data sets.

The rest of this paper is organised as follows. In Section 2 we present our definition of what a web-site is, and compares this to previous proposals. Section 3 then presents a discussion of the web-site boundary identification process, and discussion of the potential features that may be most appropriately used to identify such boundaries. The results of the evaluation of the different potential features, using a number of clustering techniques, is presented in Section 4. some conclusions are presented in Section 5.

## 2 Web-Site Definition

A number of proposals have been put forward over the years, to characterize the idea of a collection of strongly related web-pages. In the work by Senellart [19,20], the aim is to find web-pages that are contained in logically related groups using the link structure. Senellart emphasises the fact that there is no clear definition of what a web-site is, and defines a "logical" web-site as a collection of nodes that are significantly more connected than other "nodes". This definition abstracts from the physical notions described in the traditional definition (single server, single site) and makes a more subjective claim that concentrates on the similarity between pages.

Work has also been done in the area of detecting web *subsites* by, for example, Rodrigues et al. [16,17] and Neilsen [15]. The authors use the word subsite to refer to a collection of pages, contained within a main web-site, that fills the criteria of having a home page, and having distinct navigation and styling from the main pages of the site.

Research by Dmitriev [5,7] brings about the notion of *compound documents*. This is a set of web-pages that aggregate to a single coherent information entity. An example of a compound document is a news article that will be displayed over several pages within the news web-site, each with a unique URL. The authors intention is for the reader to absorb the article as a single piece of information [7]. Some points to note about compound documents is that they have an entry point (which can be non trivial to find) which is similar to the definition of a subsite above. Using the definition it challenges the synonymous notion of web node equals web-page.

It is suggested, in the context of boundary detection, that an appropriate definition must encompass several of the above concepts. The following definition is therefore proposed:

**Definition 1.** *A* web-site *is a collection of web-pages that:*

**WS1** *have a common entry point, referred to as the web-site* home-page*, such that every page in the collection is reachable from this home-page through a sequence of directed hyperlinks;*

**WS2** *have distinct navigation or styling features, and*

**WS3** *have a focused content and intention.*

The first two elements in the statement above are syntactic in nature. They refer to clearly recognizable features of the given collection of web-pages. The third one is intended to capture the purposes of the creators of the given collection.

Considering the above definition in further detail it should be noted that the definition is couched in terms of the expected structure of a web-site, and that some of the elements of the definition build upon existing ideas found in the literature. Constraint **WS1** is probably the most obvious one, and its importance has been recognized previously (see for instance [12,13,15,21]). It is also natural to add a constraint like **WS2**; similar styling is a clear sign of authorship. Collections of web-pages that have the same styling tend to have been created by the same people. Minor differences may arise between pages in the same collection, however common themes will often be shared by all pages that are part of a single conceptual unit. Constraint **WS2** also refers to the possibility that many pages in the same site may have similar link patterns. The styling may be completely different, but the navigation of the pages may share some common links (for instance a back link to the web-site home-page). As to **WS3**, the idea of focused content and intention has never been explicitly included in a web-site definition, although it is implicitly present in other proposals (e.g. [2]). The idea reflects the situation where an author has control over a collection of pages so that the pages can thus be said to be related by the author's intention.

It is perhaps also important to stress that we move away from the popular graphical vision associated to the web (see e.g. [4]). Web-sites are collections of related web-pages, but their hyper-link structure is only one of the many possible features that one should consider when grouping related pages. It will become apparent that hyper-links (directed out-going links) from a page are important, but, for instance, "popular-pages" [11] (a notion derived from the analysis of in-going links) seem to be less relevant with respect to web-site boundary definition.

The above definition (in the context of boundary detection) offers a number of advantages:

**Generality:** This is more general than previous proposals. Constraint **WS1** clearly relates to the notion of *seed pages* that has been used in the past as a means of clustering content-related web-pages.**WS2** encompasses the approaches based on the study of the URL's and the link structure of the given set of pages.

**Flexibility:** The definition is flexible. It is argued that any sensible definition must contain a semantic element referring to the authors' intentions. Such an element cannot be defined prescriptively, and is application dependent. Adding such element to the definition (constraint **WS3**) makes it suitable to describe a wide range of boundary detection scenarios.

**Effectiveness:** The proposed definition is effective because, as will be demonstrated, it can be used to identify web-site boundaries using data mining techniques.

## 3 The Web-Site Boundary Identification Process and Feature Selection

We now turn to the description of the proposed approach to the problem of web-site boundary identification.

As noted above, the process of identifying web-site boundaries adopted in this paper is a static one (as opposed to a dynamic one). The process commences with a crawl whose aim is to collect a set of web-pages that will represent the domain of investigation in the subsequent boundary detection process. The start point for the crawl is the home page of the target site. The search then proceeds in a breadth-first fashion with a crawling that is not limited to URL domain or file size. Thus, for example, if an external link (e.g. `google.co.uk`) was found, it would be followed and included in the dataset. Once a sufficiently large collection of pages has been gathered, feature vectors are constructed, one for each page, and a clustering algorithm applied to distinguish the target site from the "noise" pages. To complete the description of our approach we need to specify what features (attributes) to include in the feature vector and what clustering technique is the most appropriate. A tentative answer to the latter is provided in Section 4.4, here we address the former.

The space of features that may be used to describe a given web-page is massive. The features selected in the study described here include: hyper links, image links, *Mailto* links, page anchor links, resource links, script links, title tags and URLs. We contend that web-pages grouped based on such features and arbitrary combinations therein can be considered part of the same web-site, based on the definition given in Section 2.

The hyper-link based features were constructed by extracting all of the hyperlinks from each of the pages. Each textual hyperlink, representing a pointer to another web-page was stored as a single string (the only processing that was done was that the text was cast into lower-case, to facilitate comparison). The values associated with each of the features in the hyper-link group was the number of potential occurrences (frequency count) of each identified hyper link. The theory behind the use of hyper-links is that pages that are related may share many of the same hyper-links. The shared links may be other pages in the same website (e.g. the web-site home page) or significant external pages (e.g. most pages from a Department with a University web-site may point to the main University portal).

The image links feature sub-vector was built by extracting all of the links to images ($< img >$) from each of the given pages in $W$. The image links were processed in a similar fashion to the hyper-links, as described above. Pages that link to the same images were deemed to be related; for example the same set of logos or navigation images.

Another feature sub-vector was constructed by extracting *Mailto* links from the pages in $W$. The idea is that a group of related pages may contain a *Mailto* link to a single email contact (for example the *web master*). The links are extracted from the HTML code using the same method as described above, but looking for the *Mailto* tags.

The page anchor links sub-vector was constructed by extracting all of the page anchors from each of the pages. Page anchors are used to navigate to certain places on the same page, these can be helpful for a user and can very often have meaningful names. It is conjectured that if the same or related names are used on a set of pages it could imply related content. The Page Anchor Links group of attributes were extracted by parsing the HTML code as above and identifying the number of possible occurrences (the values for the individual attributes).

The Resource Links feature sub-vector was constructed by extracting all of the resource links from the given pages. This commonly includes CSS (Cascading Style Sheet) links. The motivation is that the styling of a page is often controlled by a common CSS which could imply that a collection of pages that use the same style sheet are related. In this case the feature space is built by extracting only the resource links from the HTML.

The script links sub-vector was constructed by extracting all of the script links from each of the pages in $W$. This commonly included Java script links. The observation here is that some functions that are used on web-pages can be written and used from a common script file; if pages have common script links then they could be related. This feature sub-vector was built by extracting this information.

It is conjectured that the titles used in a collection of web-pages belonging to a common web-site are a good indicator of relatedness. The title group of features was constructed by extracting the title from each of the given pages. The individual words in each title were then processed to produce a "bag of words" (a common representation used in text mining). Note that when the textual information was extracted from the *title* tag non-textual characters were removed, along with words contained in a standard "stop list". This produced a group of feature's comprising only what were deemed to be the most significant title words.

The textual content was extracted from each page in the dataset by using a html text parser/extractor (`http://htmlparser.sourceforge.net/`). This gave only the text that would be rendered by a web browser. This is deemed to be the same text that a user would use to judge a pages topic/subject. Stop words (same list as used to identify title text above) were then removed and a bag of words model produced as in the case if the title sub-vector.

Finally the URL feature sub-vector was constructed by collating the URL's from each of the pages. The motivation is that the URL is likely to be an important factor in established whether subsets of web-pages are related or not. As noted above URL should not be considered to be a unique identifier for every web-page in the given collection. The URL of each page was split into "words" using the delimiters found in URL's. For example the URL `http://news.bbc.co.uk` would produce the attributes `news`, `bbc`, `co` and `uk`. Non textual characters were removed (no stop word removal was undertaken). The process constructed an attribute group that would have a high frequency count for common URL elements (words).

## 4 Evaluation

This section describes the results of the sequence of experiments conducted to identify the most appropriate set of features, considering a number of clustering algorithms, in the context of web-site boundary identification and with respect to the web-site definition given in Section 2. The clustering algorithms used are briefly reviewed in Sub-section 4.1. The test data is described in Sub-section 4.2. The evaluation strategy adopted is introduced in Sub-section 4.3. The results are presented and discussed in Sub-section 4.4.

### 4.1 Clustering Algorithms

Four different clustering algorithms were selected to evaluate the proposed web-site boundary identification process: two variants of the well-known $k$-means process ($k$-means and Bisecting $k$-means), $k$-nearest neighbour, and the DBSCAN. A brief overview of each is given below:

**$k$-means:** The *k-means* algorithm is an example of an iterative partitional algorithm [8]. It operates on the actual feature space of the items. Items are allocated to a user specified number of $k$ clusters. Only "spherical" shaped cluster are found, and the process has the disadvantage that results can be influenced by outliers.

**Bisecting $k$-means:** The *bisecting k-means* clustering algorithm is a partitional clustering algorithm that works by computing a user specified $k$ number of clusters as a sequence of repeated bisections of the feature space. A $k$-way partitioning via repeated bisections is obtained by recursively computing 2-way clusterings. At each stage one cluster is selected and a bisection is made[22].

**$k$-nearest neighbour:** The *k-nearest neighbour algorithm* is an iterative agglomerate clustering algorithm [8]. Items are iteratively merged into existing clusters that are "closest", within a user specified threshold value. If items exceed the threshold, they start a new cluster. The algorithm has the ability to find arbitrary shaped clusters in the feature space.

**DBSCAN:** The *DBSCAN (Density-Based Spatial Clustering of Applications with Noise)* algorithm creates clusters that have small size and density [9]. Density is defined as the number of points within a certain distance of one another. Note that the number of clusters, $k$ is not input, but it is determined by the algorithm.

The selection of candidate clustering algorithms was made according to the distinctiveness of their operation. To remove the dependence on the number of clusters of some of the algorithms, in each case the cluster containing the start page of the crawl in each data-set was designated as the *target* cluster $K_T$ (ideally, such cluster would include all pages belonging to the web-site to be archived). All other clusters were then identified as *noise* cluster $K_N$.

### 4.2 Test Data

For the purposes of the experiments a collections of web-pages was obtained by crawling the University of Liverpool's WWW site. For the evaluation four sets of web-pages were obtained, comprising 500 pages each, and describing the activity of a number of University Departments, namely: (i) Chemistry, (ii) Mathematics, (iii) History, and (iv) Archaeology, Classics and Egyptology. The data sets were identified as: $LivChem500$, $LivMaths500$, $LivHistory500$ and $LiveSace500$.

### 4.3 Evaluation Strategy

For evaluation purposes the four clustering algorithms identified above were applied to the data collection several times to identify each of the four University Departments web-sites, each time using different groups of features to characterize the given web-pages. The objective on each occasion was to correctly identify all pages describing a particular department, and group in a generic "noise" cluster all other pages. For each experiment one of the data sets was identified as the target class, $C_T$, and the remainder as noise. The results are presented in the following section.

Two measures were used to evaluate the quality of the resulting cluster configuration: (i) accuracy and (ii) entropy. The accuracy was calculated as the sum of the correctly classified target class web-pages within $K_T$ plus the sum of the number of "noise" web-pages correctly allotted outside $K_T$ divided by the total number of web-pages. Thus:

$$accuracy = \frac{correctClass(K_T) + \sum_{i=1}^{i=n} correctClass(K_i)}{|W|} \tag{1}$$

Where the function *correctClass* returns the number of correctly classified items in its argument, which must be a cluster, $K_T$ is the target cluster, $K_1$ to $K_n$ are the remaining clusters and $W$ is the input set.

Similarly, denoting by $m_{TT}$ (resp. $m_{TN}$) the number of pages from (resp. not in) the given web-site (according to the human classification) that land in $K_T$, the entropy for $K_T$ is defined as:

$$e_T = -\frac{m_{TT}}{|K_T|}\log\frac{m_{TT}}{|K_T|} - \frac{m_{TN}}{|K_T|}\log\frac{m_{TN}}{|K_T|} \tag{2}$$

(here, clearly, the size of cluster $K_T$ satisfies $|K_T| = m_{TT} + m_{TN}$). Therefore, the total entropy of the resulting set of clusters, is calculated as:

$$\frac{|K_T|e_T + (500 - |K_T|)e_N}{500} \tag{3}$$

(where $e_N$ is defined in a similar way to $e_T$ with respect to $K_N$).

### 4.4 Results and Discussion

The results of the experiments are presented in this section. Table 1 describes results using the Chemistry data-set. The table presents a comparison of the effectiveness of the proposed web-site boundary identification process using: (i) different web-page features and (ii) different clustering algorithms. The first column lists the feature of interest. The *Composite* feature (row 1) combines all features except the textual content feature (row 2). The second column gives the clustering algorithm used, and the third the value of any required parameters. The clustering algorithm, with respect to each feature, are ordered according to the accuracy value (column 5) obtained in each case. The fourth column gives the entropy value obtained using each of the 10 identified features with respect to each of the clustering algorithms.

Similar experiments were conducted with respect to the other data sets. Table 2 summarises the entire set of experiments. The column headings are the same as for Table 1. For each target class the best two performing features (according to the accuracy measure) were selected and reported in Table 2.

**Discussion of Feature Selection.** The first observation that can be made from Table 1 is that the entropy and accuracy measure corroborate each other. The second observation is that *Resource Links*, *Image Links*, *Mailto Links* and *Page Anchors*, when used in isolation, are poor discriminators. With respect to accuracy the best discriminators are (in order): *Composite*, *URL*, *Hyper Links* and *ScriptLinks*. In terms of maximising the entropy the best features are (in order): *ScriptLinks*, *URL*, Hyperlinks and Composite. Putting these results together we can observe that there are clear candidates for the most appropriate features to use for boundary identification. There is two possible reasons for the poor performance of the trailing features. One reason could relate to the absence of a feature, this could be a consequence of a specific design choice or function of a web-page. In terms of page anchors and mailto links, these feature will only be present if the specific function is needed/used for a certain page, mailto link may not be provided, or page anchors might not be used. The second reason might be because of the common presence of the feature amongst all pages in the dataset. The pages collected in the dataset that are classed as irrelevant (i.e not in the target class C*T*) still come from various divisions of the Liverpool University. If many pages use many common images, scripts or resource links, distinguishing between pages may prove quite difficult if the pages only vary by a small degree. Finally, it is perhaps worth noticing that the composite feature acts as a boost in terms of dissimilarity between pages. As described above, if the difference in the pages using a single feature are very small, then combining features will increase this small distinction, to provide a more detectable difference in the inter page dissimilarity between groups. It also copes well with missing features, as the composite feature provides other items that can be present to correctly classify data items.

Inspection of Table 2 indicates that the best discriminators, across the data sets, are: *Composite*, *URL*, *Hyperlinks* and *Textual*. The composite feature

**Table 1.** Clustering accuracy and entropy results obtained using *LivChem*500, different features and using different clustering algorithms. (Results ordered by clustering algorithm with respect to best average performing feature, according to accuracy).

| **Chemistry Department 500** (LivChem500) | | | | |
|---|---|---|---|---|
| **Feature** | **Algorithm** | **Params (optimal)** | **Entropy (%)** | **Accuracy (%)** |
| Composite | Bisecting Kmeans | k=4 | 87.09% | 98.2% |
| | Kmeans | k=4 | 86.99% | 98% |
| | DBSCAN | minPoints=1, eps=250 | 62.82% | 91.8% |
| | KNN | Threshold=20 | 46.07% | 13.2% |
| Textual | Kmeans | k=5 | 81.09% | 96.8% |
| | Bisecting Kmeans | k=4 | 66.18% | 91.6% |
| | DBSCAN | minPoints=1, eps=999 | 48.99% | 88.6% |
| | KNN | Threshold=25 | 64.02% | 23.4% |
| URL | Bisecting Kmeans | k=4 | 87.42% | 98.2% |
| | Kmeans | k=4 | 85.86% | 98.1% |
| | DBSCAN | minPoints=1, eps=5 | 58.72% | 91.6% |
| | KNN | Threshold=5 | 45.92% | 12.4% |
| Hyperlinks | Kmeans | k=5 | 87.28% | 98.2% |
| | Bisecting Kmeans | k=5 | 65.78% | 93% |
| | DBSCAN | minPoints=1, eps=250 | 55.87% | 90.6% |
| | KNN | Threshold=30 | 45.96% | 12.6% |
| Title | Kmeans | k=6 | 83.98% | 97% |
| | DBSCAN | minPoints=3, eps=5 | 54.41% | 90.4% |
| | Bisecting Kmeans | k=4 | 60.14% | 84.6% |
| | KNN | Threshold=5 | 45.92% | 16.8% |
| ScriptLinks | Kmeans | k=4 | 88.25% | 97.8% |
| | Bisecting Kmeans | k=3 | 64.08% | 91.8% |
| | DBSCAN | minPoints=3, eps=5 | 47.73% | 46.6% |
| | KNN | Threshold=1 | 45.92% | 12.4% |
| ResourceLinks | DBSCAN | minPoints=3, eps=5 | 63.23% | 91.8% |
| | Bisecting Kmeans | k=5 | 52.85% | 63% |
| | Kmeans | k=5 | 55.29% | 61.6% |
| | KNN | Threshold=5 | 45.92% | 12.4% |
| MailtoLinks | Bisecting Kmeans | k=6 | 48.00% | 74.2% |
| | Kmeans | k=7 | 46.00% | 12.8% |
| | DBSCAN | minPoints=1, eps=200 | 45.92% | 12.4% |
| | KNN | Threshold=5 | 48.98% | 11.4% |
| ImagesLinks | Bisecting Kmeans | k=6 | 46.05% | 34.4% |
| | Kmeans | k=8 | 48.95% | 27% |
| | DBSCAN | minPoints=1, eps=250 | 46.19% | 13.8% |
| | KNN | Threshold=15 | 46.11% | 13.4% |
| PageAnchors | Bisecting Kmeans | k=9 | 45.92% | 12.4% |
| | Kmeans | k=9 | 45.92% | 12.4% |
| | KNN | Threshold=5 | 45.92% | 12.4% |
| | DBSCAN | minPoints=1, eps=1 | 45.92% | 12.4% |

**Table 2.** Best results for all four test set combinations (Results ordered by clustering algorithm with respect to best average performing feature, according to accuracy)

| **Departments from University Of Liverpool** | | | | |
|---|---|---|---|---|
| **Best performing feature** | **Best performing Algorithm** | **Params (optimal)** | **Entropy (%)** | **Accuracy (%)** |
| Chemistry Department (LivChem500) | | | | |
| Composite | Bisecting Kmeans | k=4 | 87.09% | 98.2% |
| | Kmeans | k=4 | 86.99% | 98% |
| URL | Bisecting Kmeans | k=4 | 87.42% | 98.2% |
| | Kmeans | k=4 | 85.86% | 98.1% |
| Mathematics Department (LivMaths500) | | | | |
| Textual | Bisecting Kmeans | k=8 | 76.3% | 96% |
| | Kmeans | K=7 | 75.35% | 95.8% |
| Hyperlink | DBSCAN | minPoints=3, eps=5 | 69.72% | 94.4% |
| | KNN | Threshold=90 | 44.16% | 86.8% |
| History Department (LivHistory500) | | | | |
| Composite | Bisecting Kmeans | k=3 | 77.28% | 96% |
| | Kmeans | k=6 | 72.83% | 95.2% |
| Hyperlinks | Bisecting Kmeans | k=3 | 75.06% | 92.6% |
| | Kmeans | k=5 | 72.25% | 95.2% |
| School of Archaeology, Classics and Egyptology (LivSace500) | | | | |
| Composite | Bisecting Kmeans | k=5 | 82.33% | 89.2% |
| | Kmeans | k=9 | 85.03% | 93.2% |
| Hyperlinks | Bisecting Kmeans | k=4 | 72.84% | 79% |
| | Kmeans | k=3 | 74.36% | 70% |

performs the best in three out the four cases and can thus be argued to have the best performance overall. It is conjectured that this is because it is the most robust comprehensive representation, and thus can operate better with respect to missing or irrelevant values in the vector space (compared to using features in isolation). For example, title seems to be a good indicator of pages in the same web-site, but if a title tag is missing then the page will be missed completely. Using a composite set of features boosts the performance, and helps find pages that span across multiple domains and services within the input data. There are some cases were the Textual (content) works well. However, content tends to be dynamic and is subject to change; it is suggested that the composite feature representation would be able to deal effectively with such changes.

In general, it can be said that the features considered in the composite feature include attributes of a web-page that are more representative of authors' overall intentions, rather than the authors means of conveying an idea. The composite feature representation will model a page using: the URL which can be considered as the place in the web structure it resides, the title provides a round-up of the overall message the page conveys, the hyperlinks consider the position it is in the website site structure (home page, leaf node etc); while the resource, script, mail, image and page anchors links provide a consistent representation of the skeleton structure of the page. All features in combination perform better than in isolation. The performance is better than only textual content, which can be thought of as a representation of the target information an author is trying to convey at a specific point. This is subject to change as events/schedules or activities change. The main skeleton structure will remain fairly consistent, and thus, in the experiments conducted in this paper, prove to be a better model for website boundary identification according to our definition.

**Discussion of Clustering Algorithms.** The best overall clustering algorithms tend to be Bisecting Kmeans and Kmeans. It is worth noting that the feature space that is created from each of the web-page models is quite dense, with low ranges of values with occasional outliers, and with very high frequency of certain features. Consequently the KNN and DBSCAN algorithms tend to produce clustering results that merge almost all items into a single cluster, or they overfit, and produce a single cluster for each data item (note that these clustering algorithms do not work with a predetermined number of clusters). The items in the feature space are densely packed so even using low threshold values cannot produce distinctions between related and non related items. This observation is also reflected with respect to the Kmeans and bisecting kmeans algorithms when a low initial cluster value ($k$) is used; in this case it can also be seen that the majority of items are grouped together, this is contrary to what we might expect to be produced, i.e. a cluster containing items from the ideal class and another cluster containing the remaining items.

In the early stages of the investigation it was thought that a cluster value of $K = 2$ for Bisecting Kmeans and Kmeans would be the most appropriate to distinguish between desired web-pages from the target class ($C_T$), and web-pages that are irrelevant (noise included in the crawl). However, from test results, it quickly became apparent that using $K = 2$ did not provide any useful distinction in the data sets. This was because the clusters produced by Bisecting Kmeans and Kmeans are Hyper spheres, i.e with equal radii in all $n$ dimensions. Any change in the cluster radius in any specific dimension impacted on all dimensions which meant that in some cases, given a low number of clusters (i.e. $K = 2$), some "short" dimensions was entirely encompassed by a single cluster. By increasing the value of $K$ much better results were produced as clusters were not able to grow in the same manner as with low values of $K$. Thus a high initial cluster value ($K$) was eventually used so as to distinguish between items in the densely packed feature space. The effect of this was to force the generation of many cluster centroids (in the case of kmeans) or many bisections (in the case of bisecting kmeans), This method of using high initial cluster values was re-enforced by the adverse results obtained using DBSCAN and KNN which do not operate with an initial number of cluster parameters, and instead tried to adapt to the feature space. DBSCAN and KNN either produced single clusters containing most items, or they "over-fitted" and generated a large number of clusters each containing very few items.

It can be argued, out of the clustering algorithms that were tested, that the Bisecting kmeans seemed to produce the overall best performance. The reason for this is that it suffered much less with initialisation issues; and that the feature space is bisected on each iteration which produced clusters that were not limited by *centroid distance*, as in the case of Kmeans (and others).

The method of using a high initial clustering value proved to have very good results when combined with the composite web-page representation. The features in isolation were out performed by the more robust composite feature, which is also true for the content (textual) representation. The composite feature representation using high initial cluster value for the Bisecting Kmeans algorithm produced a better more consistent performing result that fits our selective archiving application.

## 5 Conclusions

An approach to the clustering of web-pages for the purpose of web-site boundary detection has been described. The reported study focuses firstly on the identification of the most appropriate WWW features to be used for this purpose, and secondly on the nature of the clustering algorithm to be used. The evaluation indicated that web-page clustering can be used to group related pages for the purpose of web-site boundary detection. The most appropriate features, identified from the experimentation were *Composite*, *URL*, *Hyper Links* and *ScriptLinks*. These *Composite* features can be argued to be the most appropriate because it appears to be the least sensitive to noise because it provided a much more comprehensive representation (although it required more computation time to process). The most appropriate clustering algorithms, from the four evaluated, were found to be Bisecting Kmeans and Kmeans.

There are many applications that may benefit from the work. described Examples include: (i) WWW spam detection, (iii) creation of WWW directories, (iii) Search Engine Optimisation (SEO) and (iv) the generation of site maps. In future work the research team are interested in conducting experiments using much bigger data sets, including some currently popular web-sites.

## References

1. Antoniol, G., et al.: Web site: files, programs or databases? In: Proceedings of WSE 1999: 1st International Workshop on Web Site Evolution (October 1999)
2. Asano, Y., Imai, H., Toyoda, M., Kitsuregawa, M.: Applying the site information to the information retrieval from the web. In: Ling, T.W., Dayal, U., Bertino, E., Ng, W.K., Goh, A. (eds.) WISE, pp. 83–92. IEEE Computer Society, Los Alamitos (2002)

3. Bharat, K., Chang, B.w., Henzinger, M., Ruhl, M.: Who links to whom: Mining linkage between web sites. In: Cercone, N., Lin, T.Y., Wu, X. (eds.) ICDM, pp. 51–58. IEEE, Los Alamitos (2001)
4. Broder, A.Z., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomikns, A., Wiener, J.: Graph structure in the Web. In: Proceedings of the Ninth Internation World Wide Web Conference (WWW9)/Computer Networks, vol. 33, pp. 1–6. Elsevier, Amsterdam (2000)
5. Dmitriev, P.: As we may perceive: finding the boundaries of compound documents on the web. In: Huai, J., Chen, R., Hon, H.-W., Liu, Y., Ma, W.-Y., Tomkins, A., Zhang, X. (eds.) WWW, pp. 1029–1030. ACM Press, New York (2008)
6. Deegan, M., Tanner, S. (eds.): Digital Preservation. Digital futures series (2006)
7. Dmitriev, P., Lagoze, C., Suchkov, B.: Finding the boundaries of information resources on the web. In: Ellis, A., Hagino, T. (eds.) WWW (Special interest tracks and posters), pp. 1124–1125. ACM, New York (2005)
8. Dunham, M.H.: Data Mining: Introductory and Advanced Topics. Prentice-Hall, PTR, Upper Saddle River (2002)
9. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Simoudis, E., Han, J., Fayyad, U.M. (eds.) Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD 1996), pp. 226–231. ACM, New York (1996)
10. Hine, C.: Virtual methods: issues in social research on the Internet. Berg (2005)
11. Kleinberg, J.M.: Authoratitive sources in a hyperlinked environment. Journal of the ACM 46(5), 604–632 (1999)
12. Kumar, R., Punera, K., Tomkins, A.: Hierarchical topic segmentation of websites. In: Eliassi-Rad, T., Ungar, L.H., Craven, M., Gunopulos, D. (eds.) Proceedings of the Twelfth International Conference on Knowledge Discovery and Data Mining (KDD 2006), pp. 257–266. ACM, New York (2006)
13. Li, W.-S., Kolak, O., Vu, Q., Takano, H.: Defining logical domains in a web site. In: Hypertext, pp. 123–132 (2000)
14. Liu, B.: Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data. Springer, Heidelberg (2007)
15. Nielsen, J.: The rise of the subsite. useit.com Alertbox for September 1996 (September 1996)
16. Rodrigues, E.M., Milic-Frayling, N., Hicks, M., Smyth, G.: Link structure graph for representing and analyzing web sites. Technical report, Microsoft Research. Technical Report MSR-TR-2006-94, June 26 (2006)
17. Rodrigues, E.M., Milic-Frayling, N., Fortuna, B.: Detection of web subsites: Concepts, algorithms, and evaluation issues. In: Web Intelligence, pp. 66–73. IEEE Computer Society, Los Alamitos (2007)
18. Schneider, S.M., Foot, K., Kimpton, M., Jones, G.: Building thematic web collections: challenges and experiences from the september 11 web archive and the election 2002 web archive. In: Masanès, J., Rauber, A., Cobena, G. (eds.) 3rd Workshop on Web Archives (In conjunction with the 7thEuropean Conference on Research and Advanced Technologies for Digital Libraries, ECDL 2003), pp. 77–94 (2003)
19. Senellart, P.: Website identification. Technical report, DEA Internship Report (September 2003)

20. Senellart, P.: Identifying websites with flow simulation. In: Lowe, D.G., Gaedke, M. (eds.) ICWE 2005. LNCS, vol. 3579, pp. 124–129. Springer, Heidelberg (2005)
21. Xi, W., Fox, E.A., Tan, R.P., Shu, J.: Machine learning approach for homepage finding task. In: Laender, A.H.F., Oliveira, A.L. (eds.) SPIRE 2002. LNCS, vol. 2476, pp. 145–159. Springer, Heidelberg (2002)
22. Zhao, Y., Karypis, G.: Clustering in life sciences. In: Brownstein, M., Khodursky, A., Conniffe, D. (eds.) Functional Genomics: Methods and Protocols (2003)