A Two-Phase Sampling Technique for Information Extraction from Hidden Web Databases

Y.L. Hedley, M. Younas, A. James
School of Mathematical and Information Sciences
Coventry University, Coventry CV1 5FB, UK
{y.hedley, m.younas, a.james}@coventry.ac.uk

M. Sanderson
Department of Information Studies
University of Sheffield, Sheffield, S1 4DP, UK
m.sanderson@sheffield.ac.uk

ABSTRACT
Hidden Web databases maintain a collection of specialised documents, which are dynamically generated in response to users' queries. However, the documents are generated by Web page templates, which contain information that is irrelevant to queries. This paper presents a Two-Phase Sampling (2PS) technique that detects templates and extracts query-related information from the sampled documents of a database. In the first phase, 2PS queries databases with terms contained in their search interface pages and the subsequently sampled documents. This process retrieves a required number of documents. In the second phase, 2PS detects Web page templates in the sampled documents in order to extract information relevant to queries. We test 2PS on a number of real-world Hidden Web databases. Experimental results demonstrate that 2PS effectively eliminates irrelevant information contained in Web page templates and generates terms and frequencies with improved accuracy.

Categories and Subject Descriptors
H.3.5 [**Information Storage and Retrieval**]: Online Information Services – *Web-based services*.

General Terms
Algorithms, Experimentation.

Keywords
Hidden Web Databases, Document Sampling, Information Extraction.

1. INTRODUCTION
An increasing number of databases on the Web maintain a collection of documents such as archives, user manuals or news articles. These databases dynamically generate documents in response to users' queries and are referred to as Hidden Web databases [5]. As the number of databases proliferates, it has become prohibitive for specialised search services (such as search.com) to evaluate databases individually in order to answer users' queries.

Current techniques such as database selection and categorisation have been employed to enhance the effectiveness of information retrieval from databases [2, 5, 10, 11, 15]. In the domain of the Hidden Web, knowledge about the contents of databases is often unavailable. Existing approaches such as in [2, 10, 15] acquire knowledge through sampling documents from databases. For instance, query-based sampling [2] queries databases with terms that are randomly selected from those contained in the sampled documents. The techniques in [10, 15] sample databases with terms obtained from Web logs to retrieve additional topic terms. A major issue associated with existing techniques is that they also extract information irrelevant to queries. That is, information extracted is often found in Web page templates, which contain navigation panels, search interfaces and advertisements. Consequently, the accuracy of terms and frequencies generated from sampled documents has been reduced.

In addition, approximate string matching techniques are adopted by [13] to extract information from Web pages, but this approach is limited to textual contents only. Alternatively, the approaches proposed in [3, 4] analyse Web pages in tree-like structures. However, such an approach requires Web pages with well-conformed HTML tag trees. Furthermore, [3] discovers dynamically generated objects from Web pages, which are clustered into groups of similar structured pages based on a set of pre-defined templates, such as exception page templates and result page templates.

In this paper, we propose a sampling and extraction technique, which is referred to as *Two-Phase Sampling* (*2PS*). 2PS aims to extract information relevant to queries in order to acquire information contents of underlying databases. Our technique is applied in two phases. First, it randomly selects a term from those found in the search interface pages of a database to initiate the process of sampling documents. Subsequently, 2PS queries the database with terms randomly selected from those contained in the sampled documents. Second, 2PS detects Web page templates and extracts query-related information from which terms and frequencies are generated to summarise the database contents.

Our approach utilises information contained in search interface pages of a database to initiate the sampling process. This differs from current sampling techniques such as query-based sampling, which performs an initial query with a frequently used term. Furthermore, 2PS extracts terms that are relevant to queries thus generating statistics (i.e., terms and frequencies) that represent database contents with improved accuracy. By contrast, the approaches in [2, 10, 15] extract all terms from sampled documents, including those contained in Web page templates. Consequently, information that is irrelevant to queries is also extracted.

WIDM'04, November 12–13, 2004, Washington, DC, USA.

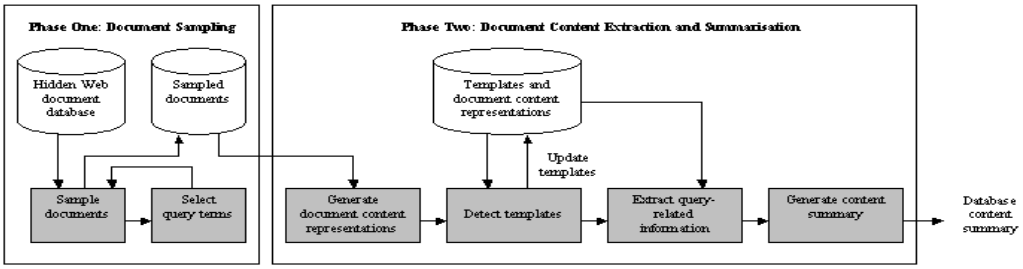

Figure 1. The Two-Phase Sampling (2PS) technique.

2PS is implemented as a prototype system and tested on a number of real-world Hidden Web databases, which contain computer manuals, healthcare archives and news articles. Experimental results show that our technique effectively detects Web page templates and generates terms and frequencies (from sampled documents) that are relevant to the queries.

The remainder of the paper is organised as follows. Section 2 introduces current approaches to the discovery of information contents of Hidden Web databases. Related work on the information extraction from Web pages or dynamically generated documents is also discussed. Section 3 describes the proposed 2PS technique. Section 4 presents experimental results. Section 5 concludes the paper.

2. RELATED WORK

A major area of current research into the information retrieval of Hidden Web databases focuses on the automatic discovery of information contents of databases, in order to facilitate their selection or categorisation. For instance, the technique proposed in [6] analyses the hyperlink structures of databases in order to facilitate the search for databases that are similar in content. The approach adopted by [10, 15] examines the textual contents of search interface pages maintained by data sources to gather information about database contents.

A different approach is to retrieve actual documents to acquire such information. However, in the domain of Hidden Web databases, it is difficult to obtain all documents from a database. Therefore, a number of research studies [2, 10, 15] obtain information by retrieving a set of documents through sampling. For instance, query-based sampling [2] queries databases with terms that are randomly selected from those contained in the sampled documents. The techniques in [10, 15] sample databases with terms extracted from Web logs to obtain additional topic terms. These techniques generate terms and frequencies from sampled documents, which are referred to as Language Models [2], Textual Models [10, 15] or Centroids [11].

A key issue associated with the aforementioned sampling techniques is that they extract information that is often irrelevant to queries, since information contained in Web page templates such as navigation panels, search interfaces and advertisements is also extracted. For example, a language model generated from the sampled documents of the Combined Health Information Database (CHID) contains terms (such as 'author' and 'format') with high frequencies. These terms are not relevant to queries but are used for descriptive purposes. Consequently, the accuracy of terms and frequencies generated from sampled documents has been reduced. The use of additional stop-word lists has been considered in [2] to eliminate irrelevant terms - but it is maintained that such a technique can be difficult to apply in practice.

Existing techniques in information extraction from Web pages are of varying degrees of complexity. For instance, approximate string matching techniques are adopted by [13] to extract texts that are different. This approach is limited to finding textual similarities and differences. The approaches proposed in [3, 4] analyse textual contents and tag structures in order to extract data from Web pages. However, such an approach requires Web pages that are produced with well-conformed HTML tag-trees. Computation is also needed to convert and analyse Web pages in a tree-like structure. Moreover, [3] identifies Web page templates based on a number of pre-defined templates, such as exception page templates and result page templates.

Our technique examines Web documents based on textual contents and the neighbouring tag structures rather than analysing their contents in a tree-like structure. We also detect information contained in different templates through which documents are generated. Therefore, it is not restricted to a pre-defined set of page templates.

Furthermore, we focus on databases that contain documents such as archives and new articles. A distinct characteristic of documents found in such a domain is that the content of a document is often accompanied by other information for supplementary or navigation purposes. The proposed 2PS technique detects and eliminates information contained in templates in order to extract the content of a document. This differs from the approaches in [1, 4], which attempt to extract a set of data from Web pages presented in a particular pattern. For example, the Web pages of a bookstore Web site contain information about authors followed by their associated list of publications. However, in the domain of document databases, information contained in dynamically generated Web pages is often presented in a structured fashion but irrelevant to queries.

Other research studies [9, 8, 12] are specifically associated with the extraction of data from query forms in order to further the retrieval of information from the underlying databases.

3. TWO-PHASE SAMPLING

This section presents the proposed technique for extracting information from Hidden Web document databases in two phases, which we refer to as *Two-Phase Sampling* (*2PS*). Figure 1 depicts the process of sampling a database and extracting query-related

information from the sampled documents. In phase one, 2PS obtains randomly sampled documents. In phase two, it detects Web page templates. This extracts information relevant to the queries and then generates terms and frequencies to summarise the database content. The two phases are detailed in section 3.1 and 3.2.

3.1 Phase One: Document Sampling

In the first phase we initiate the process of sampling documents from a database with a randomly selected term from those contained in the search interface pages of the database. This retrieves top N documents where N represents the number of documents that are the most relevant to the query. A subsequent query term is then randomly selected from terms extracted from the sampled documents. This process is repeated until a required number of documents are sampled. The sampled documents are stored locally for further analysis.

Figure 2 illustrates the algorithm that obtains a number of randomly sampled documents. t_q denotes a term extracted from the search interface pages of a database, D. qt_p represents a query term selected from a collection of terms, Q, $qt_p \in Q$, $1 \le p \le m$; where m is the distinct number of terms extracted from the search interface pages and the documents that have been sampled. R represents the set of documents randomly sampled from D. t_r is a term extracted from d_i. d_i represents a sampled document from D, $d_i \in D$, $1 \le i \le n$, where n is the number of document to sample.

```
Algorithm SampleDocument
Extract t_q from search interface pages of D, Q = t_q
For i = 1 to n
    Randomly select qt_p from Q
    If (qt_p has not been selected previously)
        Execute the query with qt_p on D
        j = 0
        While j <= N
            If (d_i ∉ R)
                Retrieve d_i from D
                Extract t_r from d_i,
                R = d_i
                Q = t_r
                Increase j by 1
            End if
        End while
    End if
End for
```

Figure 2. The algorithm for sampling documents from a database.

2PS differs from query-based sampling in terms of selecting an initial query. The latter selects an initial term from a list of frequently used terms. 2PS initiates the sampling process with a term randomly selected from those contained in the search interface pages of the database. This utilises a source of information that is closely related to its content. Moreover, 2PS analyses the sampled documents in the second phase in order to extract query-related information. By contrast, query-based sampling does not analyse their contents to determine whether terms are relevant to queries.

3.2 Phase Two: Document Content Extraction and Summarisation

The documents sampled from the first phase are further analysed in order to extract information relevant to the queries. This is then followed by the generation of terms and frequencies to represent the content of the underlying database. This phase is carried out through the following processes.

3.2.1 Generate Document Content Representations

The content of each sampled document is converted into a list of text and tag segments. Tag segments include start tags, end tags and single tags specified in HyperText Markup Language (HTML). Text segments are text that resides between two tag segments. The document content is then represented by text segments and their neighbouring tag segments, which we refer to as *Text with Neighbouring Adjacent Tag Segments* (*TNATS*). The neighbouring adjacent tag segments of a text segment are defined as the list of tag segments that are located immediately before and after the text segment until another text segment is reached. The neighbouring tag segments of a text segment describe how the text segment is structured and its relation to the nearest text segments. Assume that a document contains n segments, a text segment, txs, is defined as: $txs = (tx_i, tg\text{-}lst_j, tg\text{-}lst_k)$, where tx_i is the textual content of the i^{th} text segment, $1 \le i \le n$; $tg\text{-}lst_j$ represents p tag segments located before tx_i and $tg\text{-}lst_k$ represents q tag segments located after tx_i until another text segment is reached. $tg\text{-}lst_j = (tg_1, \ldots, tg_p)$, $1 \le j \le p$ and $tg\text{-}lst_k = (tg_1, \ldots, tg_q)$, $1 \le k \le q$.

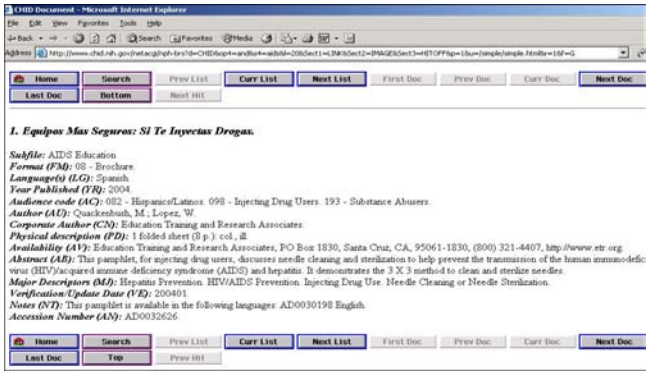

Figure 3. A template-generated document from CHID.

Figure 3 shows a template-generated document retrieved from the CHID database. The source code for this document is given in Figure 4. For example, text segment, '1. Equipos Mas Seguros: Si Te Inyectas Drogas.', can be identified by the text (i.e., '1. Equipos Mas Seguros: Si Te Inyectas Drogas.') and its neighbouring tag segments. These include the list of tags located before the text (i.e., </TITLE>, </HEAD>, <BODY>, <HR>, <H3>, <B> and <I>) and the neighbouring tags located after the text (i.e., </I>, </B>, </H3>, <I> and <B>). Thus, this segment is then represented as ('1. Equipos Mas Seguros: Si Te Inyectas Drogas.', (</TITLE>, </HEAD>, <BODY>, <HR>, <H3>, <B> ,<I>), (</I>, </B>, </H3>, <I>, <B>)). Figure 5 shows the content

representation of the CHID document (given in Figure 3) generated based on TNATS. Given a sampled document, d, with n text segments, the content of d is then represented as: $Content(d) = \{txs_1, \ldots, txs_n\}$, where txs_i represents a text segment, $1 \le i \le n$.

```
…
<HTML><HEAD><TITLE>CHID Document
</TITLE></HEAD>
<BODY>
<HR><H3><B><I> 1.  Equipos Mas Seguros: Si Te Inyectas
Drogas.
</I></B></H3>
<I><B>Subfile: </B></I>
AIDS Education<BR>
<I><B>Format (FM): </B></I>
08 - Brochure.
<BR>
…
```

Figure 4. The source code for the CHID document.

```
…
'CHID Document', (<HTML>, <HEAD>, <TITLE>),
(</TITLE>, </HEAD>, <BODY>, <HR>, <H3>, <B>,
<I>);
'1.  Equipos Mas Seguros: Si Te Inyectas Drogas.',
(</TITLE>, </HEAD>, <BODY>, <HR>, <H3>, <B>,
<I>), (</I>, </B>, </H3>, <I>, <B>);
'Subfile:', (</I>, </B>, </H3>, <I>, <B>), (</B>, </I>);
'AIDS Education', (</B>, </I>), (<BR>, <I>, <B>);
'Format (FM):', (<BR>, <I>, <B>), (</B>, </I>);
…
```

Figure 5. The content representation of the CHID document using TNATS.

3.2.2 Detect Templates

In the domain of Hidden Web databases, documents are often presented to users through one or more templates. Templates are typically employed in order to describe document contents or to assist users in navigation. For example, information contained in the document (as shown in Figure 3) can be classified into the two following categories:

(i) Template-Generated Information. This includes information such as navigation panels, search interfaces and advertisements. In addition, information may be given to describe the content of a document. Such information is irrelevant to a user's query. For example, navigation links (such as 'Next Doc' and 'Last Doc') and headings (such 'Subfile' and 'Format') are found in the document.

(ii) Query-Related Information. This information is retrieved in response to a user's query, i.e., '1. Equipos Mas Seguros: Si Te Inyectas Drogas. …'.

The 2PS technique detects Web page templates employed by databases to generate documents in order to extract information that is relevant to queries. Figure 6 describes the algorithm that detects information contained in Web page templates from n sampled documents. d_i represents a sampled document from the database D, $d_i, \in D$, $1 \le i \le n$. $Content(d_i)$ denotes the content representation of d_i.

Algorithm DetectTemplate
For $i = 1$ to n
  If $T = \varnothing$
    If $S = \varnothing$
      $S = d_i$
    Else if $S \neq \varnothing$
      While $l <=$ s AND $T = \varnothing$
        Compare ($Content(d_i)$,$Content(d_l)$)
        If $Content(d_i) \approx Content(d_l)$
          $wpt_k = Content(d_i) \cap Content(d_l)$,
          Store wpt_k, $T = wpt_k$
          Delete ($Content(d_i) \cap Content(d_l)$) from
          $Content(d_i)$, $Content(d_l)$
          $G_k = d_i$, $G_k = d_l$
          Delete d_l from S
        End if
      End while
      If $T = \varnothing$
        $S = d_i$
      End if
    End if
  Else if $T \neq \varnothing$
    While $k <= r$ AND $d_i \notin G_k$
      Compare ($Content(wpt_k)$, $Content(d_i)$)
      If $Content(wpt_k) \approx Content(d_i)$
        Delete ($Content(wpt_k) \cap Content(d_i)$) from
        $Content(d_i)$
        $G_k = d_i$
      End if
    End while
    If $S \neq \varnothing$ AND $d_i \notin G_k$
      While $l <=$ s AND $d_i \notin G_k$
        Compare ($Content(d_i)$,$Content(d_l)$)
        If $Content(d_i) \approx Content(d_l)$
          $wpt_k = Content(d_i) \cap Content(d_l)$
          Store wpt_k, $T = wpt_k$
          Delete ($Content(d_i) \cap Content(d_l)$) from
          $Content(d_i)$, $Content(d_l)$
          $G_k = d_i$, $G_k = d_l$
          Delete d_l from S
        End if
      End while
    End if
    If $d_i \notin G_k$
      $S = d_i$
    End if
  End if
End for

Figure 6. The algorithm for detecting and eliminating the information contained in Web page templates.

Similar to the representation for the contents of sampled documents, the content of a Web page template, *wpt*, is represented as $Content(wpt) = \{txs_1, \ldots, txs_q\}$, where q is the number of text segments, txs_j, $1 \le j \le q$. T represents a set of templates detected. $T = \{wpt_1, \ldots, wpt_r\}$, where r is the distinct number of templates, wpt_k, $1 \le k \le r$. G_k represents a group of documents generated from wpt_k. Furthermore, S represents the sampled documents from which no templates have yet been detected. Thus, $S = \{d_1, \ldots, d_s\}$, where s is the number of temporarily stored document, d_l, $1 \le l \le s$.

The process of detecting templates is executed until all sampled documents are analysed. This results in the identification of one or more templates. For each template, two or more documents are assigned to a group associated with the template from which the documents are generated. Each document contains text segments that are not found in their respective template. These text segments are partially related to their queries. In addition to a set of templates, the content representations of zero or more documents in which no matched patterns are found are stored.

3.2.3 Extract Query-Related Information

This process analyses a group of documents associated with each template from which documents are generated. It further identifies any repeated patterns from the remaining text segments of the documents in order to extract query-related information.

We compute cosine similarity [14] given in (1) to determine the similarities between the text segments of different documents that are associated the template where the documents are generated. The textual content of each text segment is represented as a vector of terms with weights. The weight of a term is obtained by its occurrence in the segment.

$$COSINE(txs_i, txs_j) = \sum_{k=1}^{t}(tw_{ik} * tw_{jk}) \;/\; \sqrt{\sum_{k=1}^{t}(tw_{ik})^2} * \sqrt{\sum_{k=1}^{t}(tw_{jk})^2} \quad \textbf{(1)}$$

where txs_i and txs_j represent two text segments in a document; tw_{ik} is the weight of term k in txs_i, and tw_{jk} is the weight of term k in txs_j . This is only applied to text segments with identical adjacent tag segments. Two segments are considered to be similar if their similarity exceeds a threshold value. The threshold value is determined experimentally.

The algorithm that extracts information relevant to queries is illustrated in Figure 7. d_a and d_b represent the sampled documents from the database, D, $d_a, d_b \in G_k$, where G_k denotes a group of documents associated with the template, wpt_k, from which the documents are generated. tx_m represents the textual content of a text segment, txs_m, contained in d_i, $d_i \in G_k$. tx_n represents the textual content of a text segment, txs_n, contained in d_l, $d_l \in S$. S represents the sampled documents from which no templates are detected.

The results of the above algorithm extract text segments with different tag structures. It also extracts text segments that have identical adjacent tag structures but are significantly different in their textual contents. Figure 8 shows the information extracted from the document content (given in Figure 4) as a result of eliminating information contained in the Web page template.

3.2.4 Generate Content Summary

Frequencies are computed for the terms extracted from randomly sampled documents. These summarise the information content of a database, which we refer to as *Content Summary*.

Algorithm ExtractQueryInfo
For each ($d_a \in G_k$)
  For each ($d_b \in G_k$), $d_a \neq d_b$
    Compare ($Content(d_a)$,$Content(d_b)$)
    If $Content(d_a) \approx Content(d_b)$
      Delete ($Content(d_a) \cap Content(d_b)$) from $Content(d_a)$, $Content(d_b)$
    End if
  End for
End for
For each ($d_i \in G_k$)
  Extract tx_m of txs_m from $Content(d_i)$
End for
For each ($d_l \in S$)
  Extract tx_n of txs_n from $Content(d_l)$
End for

Figure 7. The algorithm for extracting query-related information from template-generated documents.

1. Equipos Mas Seguros: Si Te Inyectas Drogas.
AIDS Education
…

Figure 8. The query-related information extracted from the CHID document.

Previous experiments in [2] demonstrate that a number of randomly sampled documents (i.e., 300 documents) sufficiently represent the information content of a database.

In the domain of Hidden Web databases, the inverse document frequency (*idf*), used in traditional information retrieval, is not applicable, since the total number of documents in a database is often unknown. Therefore, document frequency (*df*), collection term frequency (*ctf*) and average term frequency (*avg_tf*) initially used in [2] are applied in this paper. We consider the following frequencies to compute the content summary of a Hidden Web database.

- Document frequency (*df*): the number of documents in the collection of documents sampled that contain term *t*, where *d* is the document and *f* is the frequency
- Collection term frequency (*ctf*): the occurrence of a term in the collection of documents sampled, where *c* is the collection, *t* is the term and *f* is the frequency
- Average term frequency (*avg_tf*): the average frequency of a term obtained from dividing collection term frequency by document frequency (i.e., *avg_tf* = *ctf* / *df*)

Table 1. 3 Hidden Web databases used in the experiments

Database	URL	Subject	Content	Template
Help Site	www.help-site.com	Computer manuals	Homogeneous	Multiple templates
CHID	www.chid.nih.gov	Healthcare articles	Homogeneous	Single template
Wired News	www.wired.com	General news articles	Heterogeneous	Single template

The content summary of a document database is defined as follows. Assume that a Hidden Web database, D, is sampled with N documents. Each sampled document, d, is represented as a vector of terms and their associated weights [14]. Thus $d = (w_1, \ldots, w_m)$, where w_i is the weight of term t_i, and m is the number of distinct terms in $d \in D$, $1 \leq i \leq m$. Each w_i is computed using term frequency metric, *avg_tf* (i. e., $w_i = ctf_i/df_i$). The content summary is then denoted as $CS(D)$, which is generated from the vectors of sampled documents. Assume that n is the number of distinct terms in all sampled documents. $CS(D)$ is, therefore, expressed as a vector of terms: $CS(D)= \{w_1, \ldots, w_n\}$, where w_i is computed by adding the weights of t_i in the documents sampled from D and dividing the sum by the number of sampled documents that contain t_i, $1 \leq i \leq n$.

4. EXPERIMENTAL RESULTS

This section reports on a number of experiments conducted to assess the effectiveness of the 2PS technique in terms of: (i) detecting Web page templates, and (ii) extracting relevant information from the documents of a Hidden Web databases through sampling. The experimental results are compared with those from query-based sampling (abbreviated as QS). We compare 2PS with QS as it is a well-established technique and has also been widely adopted by other relevant studies [5, 10, 11, 15].

Experiments are carried out on three real-world Hidden Web document databases including Help Site, CHID and Wired News, which provide information about user manuals, healthcare archives and news articles, respectively. Table 1 summarises these databases in terms of their subjects, contents and templates employed. For instance, Help Site and CHID contain documents relating to subjects on computing and healthcare, respectively. Their information contents are homogeneous in nature. By contrast, Wired News contains articles that relate to different subjects of interest.

Where the number of templates is concerned, CHID and Wired News generate documents from one Web page template. Help Site maintains a collection of documents produced by other information sources. Subsequently, different Web page templates are found in Help Site sampled documents.

The experiment conducted using QS initiates the first query to a database with a frequently used term to obtain a set of sampled documents. Subsequent query terms are randomly selected from those contained in the sampled documents. It extracts terms (including terms contained in Web page templates) and updates the frequencies after each document is sampled. By contrast, 2PS initiates the sampling process with a term contained in the search interface pages of a database. In addition, 2PS analyses the sampled documents in the second phase in order to extract query-related information, from which terms and frequencies are generated.

Experimental results in [2] conclude that QS obtains approximately 80% of terms from a database, when 300 documents are sampled and top 4 documents are retrieved for each query. These two parameters are used to obtain results for our experiments in which terms and frequencies are generated for QS and 2PS after 300 documents have been sampled. The results generated from QS provide the baseline for the experiments.

Three sets of samples are obtained for each database and 300 documents are retrieved for each sample. First, we manually examine each set of sampled documents to obtain the number of Web page templates used to generate the documents. This is then compared with the number of templates detected by 2PS. The detection of Web page templates from the sampled documents is important as this determines whether irrelevant information is effectively eliminated.

Next, we compare the number of relevant terms (from top 50 terms) retrieved using 2PS with the number obtained by QS. Terms are ranked according to their *ctf* frequencies to determine their relevancy to the queries. This frequency represents the occurrences of a term contained in the sampled documents. *Ctf* frequencies are used to demonstrate the effectiveness of extracting query-related information from sampled documents since the terms extracted from Web page templates are often ranked with high *ctf* frequencies.

Table 2. The number of templates employed by databases and the number detected by 2PS

Databases		Number of templates	
		Employed	Detected
Help Site	Sample 1	17	15
	Sample 2	17	16
	Sample 3	19	17
CHID	Sample 1	1	1
	Sample 2	1	1
	Sample 3	1	1
Wired News	Sample 1	1	1
	Sample 2	1	1
	Sample 3	1	1

Experimental results for QS and 2PS are summarised as follows. Firstly, Table 2 gives the number of Web page templates employed by the databases and the number detected by 2PS. It shows that 2PS effectively identifies the number of templates found in the sampled documents. However, a small number of templates are not detected from Help Site. For instance, 2PS does not detect two of the templates from the first set of sampled documents, since the two templates are very similar in terms of content and structure.

Table 3 summarises the number of relevant terms (from top 50 terms ranked according to their *ctf* frequencies) obtained for the three databases. These terms are retrieved using 2PS and QS. We determine the relevancy of a term by examining whether the term is found in Web page templates. Table 3 gives the number of retrieved terms that do not appear in Web page templates. The results show that 2PS obtains more relevant terms. For instance, in the first set of documents sampled from CHID using 2PS, the number of relevant terms retrieved is 47. By comparison, the number of terms obtained for QS is 20.

The results generated from CHID and Wired News demonstrate that 2PS retrieves more relevant terms, as a large number of terms contained in the templates have been successfully eliminated from the top 50 terms. However, the elimination of template terms is less noticeable for Help Site. Our observation is that template terms attain high frequencies since the CHID and Wired News databases generate documents using a single Web page template. By comparison, a larger number of Web page templates are found in the documents sampled from Help Site. As a result, terms contained in the templates do not attain high frequencies as those found in the templates employed by CHID and Wired News.

Table 4 and 5 show the results of the top 50 terms ranked according to their *ctf* frequencies retrieved from the first set of sampled documents of the CHID database. Table 4 shows the top 50 terms retrieved for QS whereby terms contained in Web page templates are not excluded. As a result, a number of terms (such as 'author', 'language' and 'format') have attained much higher frequencies. By contrast, Table 5 lists the top 50 terms retrieved using 2PS. Our technique eliminates terms (such as 'author' and 'format') and obtains terms (such as 'treatment', 'disease' and 'immunodeficiency') in the higher rank.

Table 3. The number of relevant terms retrieved (from top 50 terms) according to *ctf* frequencies

Databases		Number of relevant terms	
		QS	2PS
Help Site	Sample 1	46	48
	Sample 2	47	48
	Sample 3	46	48
CHID	Sample 1	20	47
	Sample 2	19	47
	Sample 3	20	47
Wired News	Sample 1	14	42
	Sample 2	10	43
	Sample 3	11	39

5. CONCLUSION

This paper presents a sampling and extraction technique, 2PS, which utilises information that is contained in the search interface pages and documents of a database in the sampling process. This technique extracts information relevant to queries from the sampled documents in order to generate terms and frequencies with improved accuracy. Experimental results demonstrate that our technique effectively eliminates information contained in Web page templates, thus attaining terms and frequencies that are of a higher degree of relevancy. This can also enhance the effectiveness of categorisation in which such statistics are used to represent the information contents of underlying databases.

We obtain promising results by applying 2PS in the experiments on three databases that differ in nature. However, experiments on a larger number of Hidden Web databases are required in order to further assess the effectiveness of the proposed technique.

Table 4. Top 50 terms and frequencies ranked according to *ctf* generated from CHID when QS is applied

Rank	Term	Rank	Term	Rank	Term
1	hiv	18	document	35	lg
2	aids	19	disease	36	ve
3	information	20	published	37	yr
4	health	21	physical	38	ac
5	prevention	22	subfile	39	corporate
6	education	23	audience	40	mj
7	tb	24	update	41	description
8	accession	25	verification	42	www
9	number	26	major	43	cn
10	author	27	pamphlet	44	pd
11	persons	28	chid	45	english
12	language	29	human	46	national
13	sheet	30	date	47	public
14	format	31	abstract	48	immunodeficiency
15	treatment	32	code	49	virus
16	descriptors	33	ab	50	org
17	availability	34	fm		

Table 5. Top 50 terms and frequencies ranked according to *ctf* generated from CHID when 2PS is applied

Rank	Term	Rank	Term	Rank	Term
1	hiv	18	education	35	testing
2	aids	19	virus	36	programs
3	information	20	org	37	services
4	health	21	notes	38	clinical
5	prevention	22	nt	39	people
6	tb	23	cdc	40	hepatitis
7	persons	24	service	41	community
8	sheet	25	box	42	world
9	treatment	26	research	43	listed
10	disease	27	department	44	professionals
11	human	28	positive	45	training
12	pamphlet	29	tuberculosis	46	diseases
13	www	30	control	47	accession
14	http	31	drug	48	network
15	national	32	discusses	49	general
16	public	33	ill	50	std
17	immunodeficiency	34	organizations		

6. REFERENCES

[1] Arasu, A. and Garcia-Molina, H. *Extracting Structured Data from Web Pages*. In Proceedings of the 2003 ACM SIGMOD International Conference on Management, 2003, 337-348.

[2] Callan, J. and Connell, M. *Query-Based Sampling of Text Databases*. ACM Transactions on Information Systems (TOIS), Vol. 19, No. 2, 2001, 97-130.

[3] Caverlee, J., Buttler, D. and Liu, L. *Discovering Objects in Dynamically-Generated Web Pages*. Technical report, Georgia Institute of Technology, 2003.

[4] Crescenzi, V., Mecca, G. and Merialdo, P. *ROADRUNNER: Towards Automatic Data Extraction from Large Web Sites*, In Proceedings of the 27th International Conference on Very Large Data Bases (VLDB), 2001, 109-118.

[5] Gravano, L., Ipeirotis, P. G. and Sahami, M. *QProber: A System for Automatic Classification of Hidden-Web Databases*. ACM Transactions on Information Systems (TOIS), Vol. 21, No. 1, 2003.

[6] Heß, M. and Drobnik, O. *Clustering Specialised Web-databases by Exploiting Hyperlinks*. In Proceedings of the Second Asian Digital Library Conference, 1999.

[7] Hedley, Y.L., Younas, M., James, A. and Sanderson M. *Query-Related Data Extraction of Hidden Web Documents*. In Proceedings of the 27th Annual International ACM SIGIR Conference, 2004, 558-559.

[8] Lage, J. P., da Silva, A. S., Golgher, P. B. and Laender, A. H. F. *Automatic Generation of Agents for Collecting Hidden Web Pages for Data Extraction*. Data & Knowledge Engineering, Vol. 49, No. 2, 2004, 177-196.

[9] Liddle, S.W., Yau, S.H. and Embley, D. W. *On the Automatic Extraction of Data from the Hidden Web.* In Proceedings of the 20th International Conference on Conceptual Modeling, (ER) Workshops, 2001, 212-226.

[10] Lin, K.I. and Chen, H. *Automatic Information Discovery from the Invisible Web*. International Conference on Information Technology: Coding and Computing (ITCC), 2002, 332-337.

[11] Meng, W., Wang, W., Sun, H. and Yu, C. *Concept Hierarchy Based Text Database Categorization*. International Journal on Knowledge and Information Systems, Vol. 4, No. 2, 2002, 132-150.

[12] Raghavan, S. and Garcia-Molina, H. *Crawling the Hidden Web*. In Proceedings of the 27th International Conference on Very Large Databases (VLDB), 2001, 129-138.

[13] Rahardjo, B. and Yap, R. *Automatic Information Extraction from Web Pages*, In Proceedings of the 24th Annual International ACM SIGIR Conference, 2001, 430-431.

[14] Salton, G. and McGill, M. *Introduction to Modern Information Retrieval*. New York, McCraw-Hill, 1983.

[15] Sugiura, A. and Etzioni, O. *Query Routing for Web Search Engines: Architecture and Experiments*. In Proceedings of the 9th International World Wide Web Conference: The Web: The Next Generation, 2000, 417-430.