Automatic Identification of Best Entry Points for Focussed Structured Document Retrieval

Mounia Lalmas
Computer Science
Queen Mary, University of London
London, E1 4NS, UK
+44 (0)20 7882 5200
mounia@dcs.qmul.ac.uk

Jane Reid
Computer Science
Queen Mary, University of London
London, E1 4NS, UK
+44 (0)20 7882 5236
jane@dcs.qmul.ac.uk

ABSTRACT
Focussed structured document retrieval employs the concept of best entry points (BEPs), which are intended to provide optimal starting-points from which users can browse to relevant document components. This paper describes two small-scale studies, using experimental data from the Shakespeare user study, which developed and evaluated different approaches to the problem of automatic identification of BEPs.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *retrieval models.*

General Terms
Algorithms, Experimentation.

Keywords
Focussed structured document retrieval, Best entry points, Best entry point algorithms, Automatic identification of best entry points.

1. INTRODUCTION
Document collections often display structural characteristics, both within an individual document and between documents. Structured document retrieval (SDR) exploits structural information by retrieving documents based on combined structure and content information. *Focussed* SDR focusses retrieval by presenting only selected document components, rather than all relevant document components. This approach combines the browsing and querying paradigms to return *best entry points* to structured documents. A best entry point (BEP) is a document component from which the user can obtain optimal access, by browsing, to relevant document components [1]. The use of BEPs is thus intended to support users' information-seeking behaviour, and enable them to gain more effective and efficient access to relevant information.

CIKM '03, November 3-8, 2003, New Orleans, Louisiana, USA.

This paper describes two small-scale studies, using experimental data from the Shakespeare user study, which developed and evaluated different approaches to the problem of automatic identification of BEPs. Study 1 involved the design and implementation of algorithms to select BEPs automatically from a ranked list of retrieved document components. Study 2 involved the design of rules, based on statistical analysis of the data, to select BEPs automatically from a set of known relevant document components.

2. PREVIOUS WORK
In this section, we describe the Shakespeare user study data and previous work on BEP algorithms.

2.1 Shakespeare User Study Data
The data consists of a document collection, queries, relevance assessments and BEPs. The experimental methodology used to acquire the data is described in detail in [4].

The document collection contains 12 Shakespeare plays marked up in XML by Jon Bosak (available at http://www.ibiblio.org/bosak/). Each piece of content enclosed by XML tags is a retrievable element. The maximum depth of nested XML elements is 6. Each play contains, on average, 5,096 elements, including 5 acts, 21 scenes, 892 speeches and 3,311 lines.

The test collection includes 43 queries. Each query relates to a single play, and was produced by an experimental participant as the result of a real information need. The queries are of varying complexity: *factual*, generally answered by reference to a small number of short, simple elements, or *essay-topic*, usually requiring reference to many, complex elements. The queries also reflect the additional ability of structured query languages to query by structure, so are of two possible types: *content-only*, the standard, topical type of query in information retrieval, or *content-and-structure*, which combine topic and structural requirements.

Relevance assessments are binary, and indicate which elements the experimental participants considered they would consult in order to answer the query. All relevance assessments were made at the lowest structural level, known as *leaf level*, which corresponds mostly to Line objects. The total number of relevant leaf level elements is 4,894 (average 113.81 elements per query).

BEPs are document components from which the user can obtain optimal access, by browsing, to relevant document components. Participants were asked to identify as BEPs those elements they would wish to appear in the retrieved list. BEPs

are not always relevant elements, but are, in some cases, non-relevant container or contained elements. The total number of BEPs is 521 (average 12.12 BEPs per query).

2.2 BEP Algorithms

In [5], BEPs were identified by applying two criteria that were elicited from user studies [3, 5, 6]. According to the first criterion, C1, a parent node[1] is retrieved instead of its sub-nodes (children nodes) if many of the sub-nodes have been estimated relevant to the query; otherwise the sub-nodes are retrieved individually. The second criterion, C2, is applied to the results of C1, and selects only the *first node* from a linear sequence of closely related nodes.

C1 was implemented by basing the estimation of relevance, i.e. the retrieval status value (RSV), of a parent node on a content description derived from its own content and the content of its sub-nodes [1]. For this purpose, the content of a parent node was aggregated with that of its sub-nodes, taking into account the sub-nodes' *accessibility*, which reflects the extent to which an individual sub-node's content contributes to its parent node's aggregated representation [8]. The accessibility of a node *c* with *m* sub-nodes was modelled through an *accessibility function*, estimated in the following ways: (a) 1/m, (b) $1/m^2$, (c) $1/m^{0.5}$, and (d) fixed values of 0.2, 0.4, 0.6 and 0.8. (a) reflects the probability of a user, randomly browsing the structure of a document, arriving at a sub-node of *c*. (b) and (c) are based on the findings in [2] and allow the effect of the number of sub-nodes in the aggregation to be emphasised or de-emphasised. Both the aggregated representation, and the retrieval of nodes were formally expressed using a probabilistic framework (full details can be found in [5]).

The resulting retrieval output consists of an ordered list of document components that may be structurally related to each other. The ranking of document components varies depending on the aggregation strategy applied. To select BEPs, *redundant* nodes are removed from the results list using the criteria C1 and C2 described above. For C1, redundant nodes are those components that are hierarchically related to BEPs with a *higher* RSV. For C2, all elements of a linearly connected chain of nodes, except the first element, are considered redundant, where two nodes are defined as linearly connected if they are no further from each other than a preset threshold value, derived from our user study [5].

Experiments were performed to identify which combinations of aggregation strategies and BEP algorithms produce the best BEP retrieval effectiveness. Percentage precision values were obtained for the seven accessibility functions by averaging across standard recall points and all queries (Table 1). Our results show that these BEP algorithms, which were based on our initial analysis of user BEP criteria, led to poor performance.

Table 1. Average precision values (%) for previous BEP algorithms.

Accessibility function	1/m	$1/m^2$	$1/m^{0.5}$	0.2	0.4	0.6	0.8	Ave
C1	4.21	3.83	4.21	5.54	7.13	4.00	1.33	4.32
C2	4.68	4.78	4.75	5.95	7.23	3.51	1.21	4.59

[1] In this paper, *node* and *component* are used interchangeably.

3. STUDY 1: DERIVING BEPs FROM RETRIEVED DOCUMENT COMPONENTS

This study involved the design of new algorithms for automatically identifying BEPS from an aggregation-based ranked results list of retrieved document components.

3.1 Methodology

Four BEP algorithms were developed and implemented. The algorithms employ different combinations of information related to the individual components themselves, their structural level, and their hierarchically related components.

Algorithm 1 (**A1**): BEP selection is based on a comparison of the RSV of each parent node with the RSVs of its sub-nodes. There are two variants of this algorithm. **A1a** selects a parent node as a BEP if its RSV is *greater* than the average RSV of those of its retrieved sub-nodes; otherwise, the retrieved sub-nodes themselves are selected as BEPs. **A1b** is the inverse, selecting a parent node as a BEP if its RSV is *less* than the average RSV of its retrieved sub-nodes; otherwise, the retrieved sub-nodes themselves are selected as BEPs.

Algorithm 2 (**A2**): This algorithm, like **A3** below, ignores the parent-child relationship between a node and its sub-nodes. Nodes are selected as BEPs based purely on their location within the layers of the hierarchical structure. For each layer, the average RSV of all retrieved nodes in that layer is calculated, and the layer with the highest average RSV is selected. Nodes belonging to that layer, with an RSV above a given threshold, are selected as BEPs.

Algorithm 3 (**A3**): BEP selection is based on the ratio of retrieved nodes in a layer to the total number of nodes in that layer. There are two variants. **A3a** identifies the layer with the highest ratio, and selects nodes belonging to that layer that have an RSV higher than a given threshold as BEPs. **A3b** applies a threshold to individual nodes first, to pre-select only nodes with high RSVs, and the ratio is calculated across the selected nodes only. The selected nodes that belong to the layer with the highest ratio value are then returned as BEPs.

Algorithm 4 (**A4**): This algorithm, like **A3** above, uses a ratio based on the number of retrieved nodes and the total number of nodes in a layer; however, in this case, the ratio also takes into account the parent-child relationship between a node and its sub-nodes. For each parent node, the ratio of its retrieved sub-nodes to the total number of its sub-nodes is calculated. A threshold is then applied to select BEPs according to one of two variants: **A4a** selects the parent node as a BEP if the ratio is *below* the threshold; **A4b** selects the parent node as a BEP if the ratio is *above* the threshold.

3.2 Results

The four BEP algorithms and their variants were applied to the same ranked list of aggregation-based document components. For all algorithms, apart from A1a and A1b, we tested different threshold values; the results reported here were obtained with the optimal values. Average percentage precision values were calculated across standard recall points (Table 2).

Apart from A1a and A3a, all the algorithms perform better than our previous two algorithms. Overall, algorithms that consider only the parent-child relationship (A1a, A1b) perform less well than the other algorithms, while those that consider only structural level information (A2, A3b) seem to perform best. Better results are generally achieved by those accessibility

functions that consider the number of child nodes rather than using a static value. This result is in accordance with previous findings [5, 8]. Comparing A3a and A3b, it appears that pre-selecting nodes by applying a threshold, before calculating the average ratio, was highly beneficial. More significantly, accessibility functions that consider the number of child nodes were best served by A3b and A2, which both focus on structural level, while accessibility functions that include a static value were generally best served by A4b, which focuses on hierarchical structure. This may indicate that the most important factor is hierarchical relationships between nodes, but, where that factor is already represented in the accessibility equation, further improvement in performance may be achieved by taking structural level into account.

Table 2. Average precision values (%) for BEP algorithms.

Accessibility function	A1a	A1b	A2	A3a	A3b	A4a	A4b	*Ave*
$1/m$	2.50	3.30	63.00	0.27	27.00	3.90	3.20	*14.74*
$1/m^2$	4.80	5.30	0.27	0.27	27.00	5.40	5.40	*6.92*
$1/m^{0.5}$	2.80	3.20	27.00	1.00	27.00	3.20	1.60	*9.40*
0.2	4.00	5.50	0.36	0.27	54.00	6.80	6.80	*11.03*
0.4	6.00	5.40	0.27	0.27	2.00	6.00	6.50	*3.72*
0.6	5.50	5.40	1.80	1.80	1.80	5.60	21.00	*6.13*
0.8	5.10	5.10	0.36	0.55	0.55	5.10	6.10	*3.26*
Ave	4.39	4.61	13.30	0.63	19.91	5.14	7.23	*7.89*

Overall, the results are still poor: one reason for this may lie in the double complexity of translating criteria for *manual* selection of BEPs from known *relevant* document components into algorithms for *automatic* selection of BEPs from *retrieved* document components. Our second study, therefore, focussed on the problem of encoding human criteria in automatic rules.

4. STUDY 2: DERIVING BEPs FROM RELEVANT OBJECTS

This study explored the feasibility of encoding human criteria for selection of BEPs from relevance assessments. A second aim was to examine the effect of query complexity and type on the effectiveness of the resulting rules.

4.1 Methodology

This study was restricted to Line and Speech objects (94% of all BEPs were of these types [4]). Four rules were implemented, each with several threshold values. Each rule/threshold combination was applied to the relevant components from the collection to give a set of automatically derived BEPs, *BEP_Aut*, which was then compared with the equivalent manual BEP assessments, *BEP_Man*, using the following measures:

Misses = | BEP_Man – BEP_Aut | / |BEP_Man|: percentage of manual BEPs not identified

Falses = | BEP_Aut – BEP_Man | / |BEP_Aut|: percentage of automatic BEPs wrongly identified

Rule 1 (**R1**): This rule is based on the proportion of relevant sub-nodes (lines) contained within a parent node (container speech). The rule states that if the proportion of relevant lines exceeds a given threshold then the container speech is identified as a BEP; otherwise, the relevant lines themselves are selected as BEPs. The thresholds applied to this rule were: 20 (0-20% of sub-nodes relevant); 40 (21-40% of sub-nodes relevant); … ; 100 (81-100% of sub-nodes relevant).

Rule 2 (**R2**): This rule is based on the distribution of sub-nodes (lines) within a parent node (container speech). The distribution of relevant lines is calculated as follows:

$$\sum_{i=1}^{n-1} \frac{|R_{i+1} - R_i|}{(n-1)}$$

where $R_i = 1$ if the ith line of the speech is relevant, and 0 otherwise, and n is the number of lines in the speech. The distribution value will lie in the range 0-1, where a high distribution value indicates a uniform spread of relevant lines within the speech, and a low distribution value indicates a non-uniform spread. The rule states that if the distribution value for a particular speech exceeds a given threshold, the speech itself is identified as a BEP; otherwise the relevant lines are selected as BEPs. The thresholds applied to this rule were: 0.1 (distribution values in range 0-0.1); 0.2 (distribution values in range 0.11-0.2); … ; 0.5 (distribution values in range 0.41-0.5).

Rule 3 (**R3**): This rule is based on the observation that users often select as a BEP the first node in a linear sequence of closely related relevant nodes [4]. **R3** states that if the number of consecutive relevant lines exceeds a given threshold, the first line of that sequence should be selected as a BEP. A variant of the rule, **R3'**, states that where the first relevant line of a sequence is selected as a BEP, its container speech should also be selected as a BEP. The thresholds applied to this rule were: 2 (0-2 relevant lines in sequence); 4 (3-4 relevant lines in sequence); … ; 10 (9-10 relevant lines in sequence).

Rule 4 (**R4**): In contrast to **R3** above, this rule is based on the number of *non-relevant* consecutive lines in a sequence. **R4** states that if the number of consecutive non-relevant lines exceeds a given threshold, the first relevant line after that sequence should be selected as a BEP. A variant of the rule, **R4'**, states that where the first relevant line after a sequence of non-relevant lines is selected as a BEP, its container speech should also be selected as a BEP. The thresholds applied to this rule were: 2 (0-2 non-relevant lines in sequence); 4 (3-4 non-relevant lines in sequence); … ; 10 (9-10 non-relevant lines in sequence).

4.2 Results

The average effectiveness of each rule was analysed, across all thresholds, for individual query categories (Tables 3 and 4). Overall, R1 and R4' show the best performance in terms of misses. R4' is better than R1 for all query categories except factual queries. R3' and R4' consistently outperform R3 and R4, respectively, since they return speech BEPs as well as line BEPs. The average percentage of falses is high for all rules and query categories. R3' and R4' again outperform rules 3 and 4. Although R2 produces the best overall performance here, its poor performance in terms of misses outweighs this. Overall, R1 and R4' seemed to merit further investigation: this focussed on effectiveness in terms of misses at different thresholds for different query categories (Tables 5 and 6). The best threshold values are 2 for R4' and 100 for R1 (threshold 80 gives a very similar level of effectiveness).

We explored possible reasons for the difference between factual queries and other query categories. One potential explanation, related to the particularly poor performance of R4' with factual queries, is that factual queries can generally be answered by reference to fewer contexts, often at line level. This implies that a rule that returns parent speech objects as well as line objects may be less effective than one that returns only line objects. To test this hypothesis, we compared the effectiveness of R4' with R4 for factual queries: although R4 shows better results for factual queries than for other query

categories, R4' consistently outperforms R4 for all query categories. Another potential explanation, related to the particularly good performance of R1 with factual queries, is that many of the relevant contexts for factual queries consist of individual lines, i.e. they do not cross speech boundaries. Thus, manual BEP identification is often based only on the criterion of hierarchical structure (whether to choose a line or its parent speech as BEP) and *not* linear structure (which of the relevant objects in a sequence to choose as BEPs). R1 may, therefore, be better suited to factual queries because it focuses specifically on the importance of hierarchical structure. A combination of these explanations may explain the differences between factual queries and other query categories.

Table 3. Percentage of misses for rules, across all thresholds, for individual query categories.

Query category	R1	R2	R3	R3'	R4	R4'	*Ave*
Essay-topic	65	89	94	72	90	53	*77*
Factual	44	92	92	69	84	64	*74*
Content-only	59	90	93	73	89	56	*77*
Content-and-structure	62	86	96	65	87	52	*75*
All	60	90	93	72	88	55	*76*

Table 4. Percentage of falses for rules, across all thresholds, for individual query categories.

Query category	R1	R2	R3	R3'	R4	R4'	*Ave*
Essay-topic	84	73	89	76	90	78	*82*
Factual	48	69	70	64	76	74	*67*
Content-only	74	70	84	74	87	77	*78*
Content-and-structure	78	66	97	76	86	75	*80*
All	75	70	86	74	87	77	*78*

Table 5. Percentage of misses for R1, for differentthresholds and individual query categories.

Query category	Threshold					*Ave*
	20	40	60	80	100	
Essay-topic	79	73	70	69	36	*65*
Factual	56	43	54	35	34	*44*
Content-only	73	64	64	59	36	*59*
Content-and-structure	71	68	73	67	31	*62*
All	73	65	66	60	35	*60*

Table 6. Percentage of misses for R4', for different thresholds and individual query categories.

Query category	Threshold					*Ave*
	2	4	6	8	10	
Essay-topic	39	54	55	57	59	*53*
Factual	51	64	65	67	71	*64*
Content-only	43	56	58	60	62	*56*
Content-and-structure	36	53	53	57	57	*52*
All	41	56	57	59	61	*55*

5. CONCLUSIONS

The two studies described in this paper used experimental data from the Shakespeare user study to develop and evaluate two different approaches to the problem of automatic identification of BEPs for focussed retrieval: the first approach employed algorithms that incorporated different combinations of information related to individual retrieved nodes, their structural level, and their hierarchically related nodes; the second approach employed statistical analysis of the data in order to derive rules for automatic identification of BEPs from known relevant document components.

In Study 1, accessibility functions that consider the number of sub-nodes were generally well served by A3b and A2, which exploit structural level as the principal factor determining BEP identification, while accessibility functions that use a static value were generally best served by A4b, which exploits the parent-child relationship as the principal factor determining BEP identification. In Study 2, the most effective rule overall was R4', which exploits both hierarchical and linear relationships between relevant components; in fact, it is the inclusion of the parent-child factor that makes this rule so effective compared to R4. These results indicate that exploiting a combination of information related to, primarily, the hierarchical relationships between nodes and, secondarily, structural level, may provide the most promising strategy for automatic identification of BEPs.

Secondly, the process of encoding human criteria for BEP selection is clearly very complex, and our algorithms and rules are not yet sophisticated enough to achieve this goal effectively. Manual BEP selection takes into account both hierarchical and linear relationships between components, and is highly subjective. It is also clear from Study 2 that a further complicating factor arises from the different characteristics of individual query categories, particularly factual queries.

6. ACKNOWLEDGEMENTS

Study 1 was carried out by Jiganasha Pithia [7]. Some initial work for Study 2 was carried out by Ganan Sriganananthan [9].

7. REFERENCES

[1] Chiaramella, Y., Mulhem, P., and Fourel, F. A model for multimedia information retrieval, Technical Report Fermi ESPRIT BRA 8134, University of Glasgow, 1996.

[2] Fuhr, N., and Großjohann, K. XIRQL: a query language for information retrieval in XML documents, in ACM SIGIR, New Orleans, USA, 2001.

[3] Hertzum, M., Lalmas, M., and Frøkjaer, E. How are searching and reading intertwined during retrieval from hierarchically structured documents? in INTERACT, Japan, 2001.

[4] Kazai, G., Lalmas, M. and Reid, J. Construction of a test collection for the focussed retrieval of structured documents, in ECIR, Pisa, 2003.

[5] Kazai, G., Lalmas, M. and Roelleke, T. Focussed structured document retrieval, in SPIRE, Lisbon, 2002.

[6] Lalmas, M., Reid, J. and Hertzum, M. Best Entry Points for Structured Document Retrieval. In preparation.

[7] Pithia, J. Best entry point algorithms for focussed structured document retrieval. MSc Project, Queen Mary University of London, 2002.

[8] Roelleke, T., Lalmas, M., Kazai, G., Ruthven, I., and Quicker, S. The accessibility dimension for structured document retrieval, in ECIR, Glasgow, 2002.

[9] Sriganananthan, G. An investigation into methodologies for the automatic construction of test collections of XML structured documents, BSc Final Year Project, Queen Mary University of London, 2002.