Automatic identification of the most important elements in an XML collection

Alexander Krumpholz
ICT Centre
CSIRO
ACT 2601 Australia
krumpholz@acm.org

Amir Hadad
RSCS
Australian National University
ACT 0200 Australia
amir.hadad@anu.edu.au

Nina Studeny
University of Applied Science
Technikum Wien
A-1200 Vienna, Austria
nina.studeny@gmail.com

Tom Gedeon
RSCS
Australian National University
ACT 0200 Australia
tom.gedeon@anu.edu.au

David Hawking
Funnelback and Australian National University
Canberra, Australia
david.hawking@acm.org

Abstract *An important problem in XML retrieval is determining the most useful element types to retrieve – e.g. book, chapter, section, paragraph or caption. An automated system for doing this could be based on features of element types related to size, depth, frequency of occurrence, etc. We consider a large number of such features and assess their usefulness in predicting the types of elements judged relevant in INEX evaluations for the IEEE and Wikipedia 2006 corpora. For each feature we automatically assign Useful / Not-Useful labels to element types using Fuzzy c-Means Clustering. We then rank the features by the accuracy with which they predict the manual judgments. We find strong overlap between the top-ten most predictive features for the two collections and that seven features achieve high average accuracy (F-measure* $> 65\%$*) acrosss them. We hypothesize that an XML retrieval system working on an unlabelled corpus could use these features to decide which retrieval units are most appropriate to return to the user.*

Keywords XML Retrieval, Fuzzy C-Means Clustering, F-Measure.

1 Introduction

Information retrieval (IR) systems attempt to find the documents in a corpus which best match a given query. In traditional IR systems the document is the obvious unit of retrieval. However, when documents are explicitly structured, e.g. in the Extensible Mark-up Language (XML), it may be more natural to retrieve sub-elements, such as sections of a paper, or chapters of a book. This raises a number of additional questions: What is the optimal granularity of result elements? Should retrieval results be presented which overlap or subsume each other? What element types make or do not make good retrieval units?

In the present work we address the latter question using resources developed by INEX (Initiative for the Evaluation of XML Retrieval)[4]. The INEX organisers have provided XML corpora and participants have contributed queries and assessments in an annual cycle since 2002.

Given the large number of distinct element types typically found in an XML corpus (e.g. 1257 in the 2006 INEX Wikipedia corpus), an automatic method for determining which element types (identified by their tag) make useful units of retrieval would be of considerable value.

In a first step toward achieving this we calculate feature scores for each tag in a corpus. For each feature we use a the Fuzzy c-Means (FCM) clustering method to label each tag as Useful or Not-Useful. We then compute the accuracy (F-measure) with which the automatically assigned labels align with the sets of tags appearing in the official INEX judgments. We do this for both the IEEE and Wikipedia 2006 corpora from INEX[1] and identify the features which best predict the usefulness of an element type as a unit of retrieval.

Proceedings of the 16th Australasian Document Computing Symposium, Canberra, Australia, 2 December 2011.

[1] The only XML corpora for which we had topics and assessments at the time. The XML version of Wikipedia pages was compiled by Ludovic Denoyer [3].

Naturally, a method for estimating the usefulness of element types as retrieval units must be combined with a normal retrieval system which estimates the relevance of the content of an element to the query.

In previous work in this area:

- Mihajlović et al. [5] examined structural knowledge for Information Retrieval in XML Databases, but explicitly exclude background statistics like element frequency.
- Ali et al.[1] performed statistical analyses of XML structure. They used structural summaries of the XML documents in the corpus to answer queries with structural constraints.

2 Data used

Collection	No. doc.s	No. unique tags
IEEE	16,819	178
Wikipedia	659,388	1.257

Table 1: Basic characteristics of the INEX collections examined.

The IEEE collection comprises journal articles which are well marked-up in XML, including citations. The Wikipedia articles are far less homogeneously structured.

3 Candidate features

An XML file is made up of different elements. Each element exists in a context defined by its parent nodes (elements) and its child nodes. Any new collection to be indexed will have new element types with unknown a-priori probability for their likelihood to be relevant.

To gain more information about each element type as well as to be able to classify certain nodes, we analysed XML elements of different XML corpora. For each element we collected various characteristics for comparison later on. The main characteristics relevant to this study were:

Name The name of the element is saved to identify it.

Frequency The number of occurrences of this element within the whole corpus.

Size (CharExclKids, CharInclKids, AVGCharExclKids, AVGCharInclKids) The size is counted in characters. Two values are collected in character size: Character excl. Character of child elements – meaning that the text had to be in the very element itself – and Character incl. Character of child elements – meaning the sum of characters occurring in the element itself or in any of its child elements. Listing 1 shows an example for an XML element without text nodes.

Listing 1: XML element with only child nodes.

```
<doc>
    <from>Peter</from>
    <body>Just a body</body>
</doc>
```

In this case the characters exclusive of child element characters for 'doc' would be 0, whereas the characters including the characters of the child elements equals 15. The size was believed to be the most important charateristic of an element. Too small elements might include good key words, but are too small to return to the user since they might not include all the needed information or might be useless out of context. As well, elements, which are too large include most likely good information, but it might be hard for the user to locate it.

Text nodes (TextNodeOcc) The number of text fields within an element. Two text fields can just be separated by other elements (children). Therefore this value can never be higher than the number of children nodes of an element + 1. Listing 2 shows an example.

Listing 2: XML element with text and child nodes.

```
<doc>
   first text node
    <from>Peter</from>
    <body>Just a body</body>
   second text node
</doc>
```

Child nodes (CountKids, AVGCountKids) The number of child nodes was saved, as well as the number of different types of children occurring within an element. For each child element different attributes were saved as well, like the occurrence, if this child element occurred every time in this certain parent element, or if its content was always numeric. Minimum, maximum, average and median values were generated.

Attributes (AttCount) The attributes were saved just the way the children nodes are, so their number as well as their label were marked down for later evaluation. The number of attributes that had been found was stored as AttCount.

Depth A list keeps track of the depth in which the element occurred and delivered the min, max, average and median values.

File Occurrence (FileOcc) A Boolean variable supported the process of finding 'Must elements', meaning elements, which occured in every file. The number of files where the element was found was saved as well.

Number of different child nodes of this element (NumbKidTypes) The number of child element types the XML element had within the collection.

In order to be able to sub-sequentially evaluate which elements are more important than others, the

INEX assessments were used. The elements were enriched by information about their assessment as being relevant.

Relevant If an element was ever marked in an assessment, this value was set to 1, else 0

Occurrence in Assessment(OccAss) The number of times an element is marked relevant.

Occurrence as Cover node We defined a cover node to be the node that covers all data marked relevant in a single assessment, i.e. the root node of the minimal sub tree containing relevant sections. When a passage was marked in an assessment it included most of the times a number of nodes, as shown in figure 1. The cover node would have been in this case the best node to return to the user, since it just enfolded all relevant information. In this case the C element is the cover node.

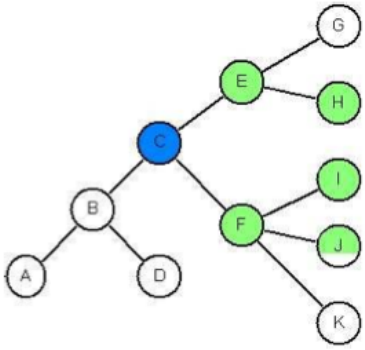

Figure 1: The cover node C, covering the whole passage marked as relevant by an INEX assessor.

4 Method

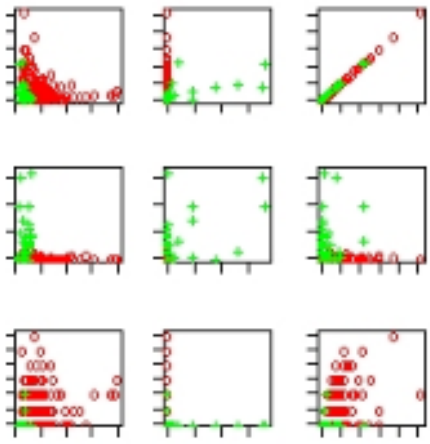

Figure 2: Small extract of the scatter plot matrix. Each plot compares the distribution of feature values for elements judged relevant and those judged irrelevant. [Unfortunately, this figure must be viewed in colour.]

Initially we compared the collected characteristics in a huge scatter plot matrix to get a better understanding of the collections. A small extract can be seen in figure 2. It shows elements marked as relevant in another colour to allow the visual identification of relevant element characteristics.

This approach indicated interesting characteristics, but we needed an objective and deterministic way of identifying the most useful features (properties). We employed Fuzzy c-Means clustering (FCM), first introduced by Bezdek in 1981 [2].

As the first step, we created two clusters for each individual feature based on the value calculated for each tag. As the second step, we divided the tags into relevant and non relevant set of tags based on these two clusters and their cut-point for each feature. As the final step, we measured the alignment of the automatically labeled tags with the published INEX relevance labels. We measured Precision, Recall and F-Measure and sorted the features on decreasing F-measure.

We applied this method for both IEEE and Wikipedia 2006 INEX collections. Among the results judged relevant for Wikipedia at INEX, 72 out of 1257 element types appeared. The corresponding ratio for IEEE was very different: 122 out of 177.

5 Results and Discussion

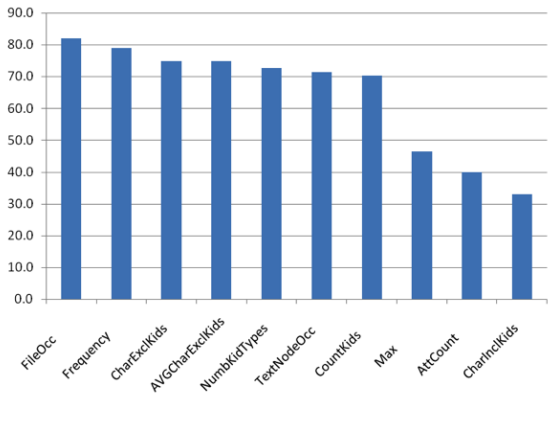

Figure 3: Top 10 F-Measure percentages for Wiki corpus features.

Figure 3 and figure 4 show the top 10 F-measure percentages for Wiki and IEEE corpora. These F-measure values were based on calculated precision and recall for the features. Note that despite the marked differences between the two corpora, eight features appear in both top-ten lists.

Finally we selected the 8 common features between the two corpora as the best list of features which can represent a corpus. We ranked these features based on the average of their two F-measure values. It can be seen that seven of the features achieve an F-measure score in excess of 65%. We propose that these features con be used to identify and select important tags for other corpora automatically.

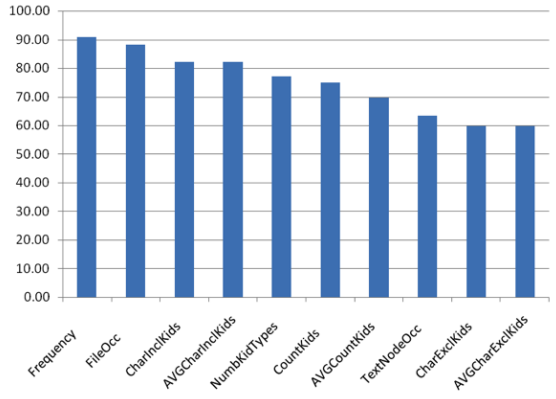

Figure 4: Top 10 F-Measure percentages for IEEE corpus features.

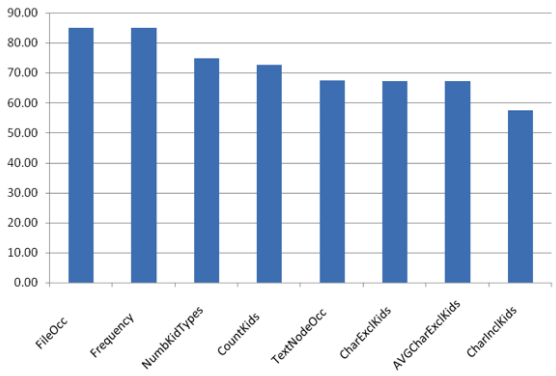

Figure 5: F-Measure percentages averaged across Wikipedia 2006 and IEEE for the eight features in common between the two Top Tens.

In a hypothetical retrieval system which calculated probabilities of relevance to a query for all elements, estimates of the usefulness of each element type as a retrieval unit could be fed into the ranking function as prior probabilities.

One limitation of our work is that we used the name of a tag to uniquely identify an element type. In general, XML allows the same tag name to appear as an element at different levels in the hierarchy. For example, the element <name> as child of an element <person> might have different child nodes and attributes, than the element <name> within <project>. This limitation would need to be removed in transferring our method into practice.

Obviously, Fuzzy c-Means clustering is far from the only possible method which could be used to select features. Future work may discover alternative methods which outperform even the relatively promising results reported here.

6 Conclusion

We have applied Fuzzy c-Means clustering to a number of statistical features of element types within an XML corpus in an attempt to label the element types as "useful unit of retrieval" or otherwise. We computed the accuracy with which these automatically assigned labels align with the manual judgments in two very different INEX test collections. We found substantial overlap between the best features across the two collections. We identified seven features whose average prediction accuracy (F-measure) across the collections exceeded 65%.

We hypothesise that these features could be used to improve performance of an XML retrieval system operating over a corpus for which no judgments are available.

References

[1] Mir Sadek Ali, Mariano P. Consens, Xin Gu, Yaron Kanza, Flavio Rizzolo and Raquel Kolitski Stasiu. Efficient, effective and flexible XML retrieval using summaries. In *INEX 2006 Revised and Selected Papers*, 2006.

[2] J.C. Bezdek. *Pattern recognition with fuzzy objective function algorithms*. Kluwer Academic Publishers Norwell, MA, USA, 1981.

[3] L Denoyer and P Gallinari. The wikipedia XML corpus. *ACM SIGIR Forum*, Jan 2006.

[4] N. Fuhr, N. Gövert, G. Kazai and M. Lalmas. INEX: INitiative for the Evaluation of XML retrieval. In *Proceedings of the SIGIR 2002 Workshop on XML and Information Retrieval*, 2002.

[5] V Mihajlovic, D Hiemstra, H Blok and P Apers. Utilizing structural knowledge for information retrieval in XML databases. *wwwhome.cs.utwente.nl*, Jan 2005.