Summarisation of the logical structure of XML documents

Zoltán Szlávik[a,*], Anastasios Tombros[b], Mounia Lalmas[c]

[a] Department of Computer Science, Vrije Universiteit Amsterdam, 1081 HV Amsterdam, The Netherlands
[b] Department of Computer Science, School of Electronic Engineering and Computer Science, Queen Mary University of London, E1 4NS, London, United Kingdom
[c] Yahoo! Research Barcelona, Avinguda Diagonal 177, 08018 Barcelona, Spain

Abstract

Summarisation is traditionally used to produce summaries of the textual contents of documents. In this paper, it is argued that summarisation methods can also be applied to the logical structure of XML documents. Structure summarisation selects the most important elements of the logical structure and ensures that the user's attention is focused towards sections, subsections, etc. that are believed to be of particular interest. Structure summaries are shown to users as hierarchical tables of contents. This paper discusses methods for structure summarisation that use various features of XML elements in order to select document portions that a user's attention should be focused to. An evaluation methodology for structure summarisation is also introduced and summarisation results using various summariser versions are presented and compared to one another. We show that data sets used in information retrieval evaluation can be used effectively in order to produce high quality (query independent) structure summaries. We also discuss the choice and effectiveness of particular summariser features with respect to several evaluation measures.

Key words:

Structure summarisation, XML retrieval

[*] Corresponding author
Email addresses: z.szlavik@vu.nl (Zoltán Szlávik), tassos@eecs.qmul.ac.uk (Anastasios Tombros), mounia@acm.org (Mounia Lalmas)

1. Introduction

In information retrieval, and other related areas, summarisation is traditionally used to create short ‘snippets’ of retrieved documents, which are displayed in the ranked list of results. Based on these snippets, and other information such as document title, the user can then decide if the corresponding document might contain relevant information. A summary provides an overview of the textual contents of a document and thus, it facilitates the information finding process of the user (Tombros and Sanderson, 1998).

In structured document retrieval, it is not only documents that are returned in response to a query, but also, portions of documents (Lalmas and Baeza-Yates, 2009). The relevance of these portions can be determined by exploiting the logical structure of documents. Nowadays, structured document retrieval is mainly studied in the context of XML documents where the logical structure of documents is provided via the XML markup (Lalmas and Tombros, 2007). The logical units (e.g. sections, subsections, etc.) of documents, called elements, form a hierarchical structure in an XML document. This hierarchical structure of a document can be overwhelmingly rich, hence, users need to gain an overview of the logical structure in order to find the document portion(s) that might contain the specific information they are looking for. In other words, the structure also needs to be ‘summarised’ and a structure summary needs to be displayed. This paper is concerned with the generation of such structure summaries.

We view text summarisation (Nenkova and McKeown, 2011) and structure summarisation as highly related: while a snippet is a selection of sentences, phrases, etc. of the textual content of a document, a structure summary is a selection of elements that provides an overview of the logical structure of the document. In addition, selected elements can also provide context to one another when displayed to the user. Based on the similarity between text summarisation and structure summarisation, we create structure summaries using similar methods to those known in text summarisation. Structure summaries can then be displayed to users as tables of contents.

Traditionally, one chooses the elements to be displayed in a table of contents (ToC) by simply selecting all the sections, subsections, etc. However, we have shown in previous work that some portions of documents might be more important to a user, and thus, these portions should be made more prominent in the table of contents (Szlávik et al., 2006a). For example, for some sections, we might need to include paragraphs in the corresponding ToC, while other sections (being unimportant or not relevant) might be completely omitted from it. The 'right' ToC should be determined automatically. The structure summarisation discussed in this paper is used to automatically determine which portions of documents are 'worthy' of inclusion in a ToC.

In this paper, we investigate structure summarisation as a means to create overviews of the logical structure of XML documents. We are interested to learn if and how structure summaries can be created automatically. At this stage, it should be noted that we are not focusing on finding a 'best' way to generate structure summaries, nor do we aim at studying the impact of structure summaries on users of retrieval systems, but our focus is on investigating how structure summaries can be created automatically.

After providing background of our work (Section 2), we describe two methods for creating structure summaries (Section 3); we investigate the summaries obtained using the second method in subsequent sections[1]. In Section 4, we present how our structure summariser is trained, and we continue with the description of the evaluation methodology followed in our work (Section 5). In Section 6, we discuss the effectiveness of various element features for structure summarisation, and we compare three methods for training our summariser. We close with the conclusions and future work (Section 7).

[1] Note that this paper focuses on summarisation and XML element features that are query-independent. This is to create general overviews of the logical structure of documents, and the investigated summarisation methods can later serve as bases for query-dependent structure summarisation.

2. Background

XML retrieval, which we consider equivalent to structured document retrieval from this point on in the paper, focuses on the logical structure of documents in a way that allows corresponding IR systems to return document portions, called elements. This way the searching process becomes more focused and information within documents becomes directly accessible: when a user searches for information, the system can display a single element's content. However, elements (i.e. sections, sub-sections, etc.) of a document are strongly related, often parts of each other and they provide context to the element that is retrieved as well as to one another. Also, the logical structure of the whole document is known, and this allows to display not only the content of a particular element but the logical structure of the document, sub-structure of the retrieved element, or sections of the document that are also relevant to the user's query. For example, if a section is relevant then the document it is in is also relevant to some extent (as it contains at least one relevant section). At least some of the subsections of a relevant section are also relevant as they make up the relevant content of the section. All the above elements are related, but their relevance 'levels' are different (Lalmas and Tombros, 2007), and the user should see elements in the context of one another to make an informed decision as to where to start reading a document, or where to go next if the first choice does not lead to the most relevant content. As research has showed, displaying the logical structure of the document, both when the whole document's content is found relevant or when only a relatively small element is returned, is something that XML retrieval system users appreciate (Kazai and Trotman, 2007). Thus, an overview of the logical structure needs to be provided in order to support users in their searching process, to provide context to retrieved elements.

Overviews of hierarchically structured information can be displayed in various ways. Several overview methods have been proposed over time, for example, TileBars (Hearst, 1995) visualise frequency and distribution of terms, and Partial Treemaps (Großjohann et al., 2002) present the relative relevance of elements within a document. Nevertheless, highly graphical overview presentation methods have not made lasting impact yet, mostly because time is needed for searchers to learn the metaphors, structure and navigation (Sebrechts et al., 1999). For hierarchical data, particularly if the hierarchy represents the logical structure of textual documents, an intuitive and natural way to gain an overview of the document is through a table of contents (ToC). ToC-like displays are often used when the set of data is a set of documents, for example, WebTOC automatically generates a hierarchical table of contents of a web site (Nation, 1998), and categories (topics) of documents are automatically determined and displayed in a hierarchical, table of contents like manner in the work by Lawrie (2003). Results of web clustering are also often presented in a way that resembles tables of contents (Carpineto et al., 2009).

A table of contents typically gives an overview of the logical structure of a book or article. When a ToC is not available directly for a document there is a need to create one automatically. Elements corresponding to sections and subsections that are 'ToC-worthy' need to be selected and their titles displayed in the ToC. For example, a ToC-worthy element cannot be too small, e.g. a paragraph containing only one short sentence is probably not ToC-worthy. Figure 1 shows various possible ToCs for a document and illustrates how important it is to select ToC-worthy elements that are meaningful and important to display a reference to. For instance, it is unlikely that a user wants to see the word '1879' in the ToC (leftmost column) if, after clicking on it, they find that the target element contains only this word as text. The section 'Honors' (rightmost column) might or might not be important to display in a ToC, but whether it needs to be shown may depend on several factors, such as how many honours are listed there, how detailed the listing is, how prestigious the honours are, etc. (note that relevance to a particular information need, or query, is not considered in this work).

When selecting elements that are ToC-worthy, a decision has to be made whether a reference to that element's contents should be included in the ToC. This makes ToC generation a binary classification task, i.e. an element should be

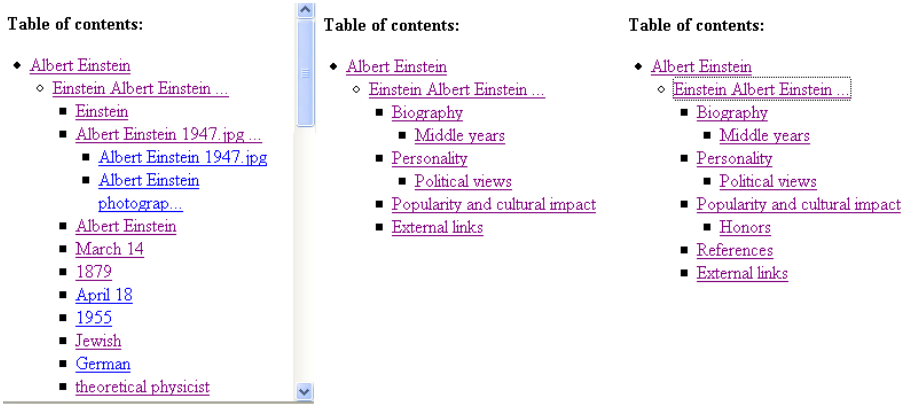


Table of contents:

- Albert Einstein
  - Einstein Albert Einstein ...
    - Einstein
    - Albert Einstein 1947.jpg ...
      - Albert Einstein 1947.jpg
      - Albert Einstein photograp...
    - Albert Einstein
    - March 14
    - 1879
    - April 18
    - 1955
    - Jewish
    - German
    - theoretical physicist

Table of contents:

- Albert Einstein
  - Einstein Albert Einstein ...
    - Biography
      - Middle years
    - Personality
      - Political views
    - Popularity and cultural impact
    - External links

Table of contents:

- Albert Einstein
  - Einstein Albert Einstein ...
    - Biography
      - Middle years
    - Personality
      - Political views
    - Popularity and cultural impact
      - Honors
    - References
    - External links

Figure 1: Several possible ToCs for a document about Albert Einstein.

classified either into the ToC-worthy class or the not-ToC-worthy class. Classification is also used similarly in traditional extractive type summarisation where units of a text, usually sentences, that are selected into the summary-worthy class form a summary of the textual contents of a document. As ToC generation works very similarly to the above mentioned summarisation it can also be regarded as a kind of summarisation, i.e. that of the structure of documents.

Before discussing structure summarisation in more detail, let us consider document text summarisation first. Summarisation of the text can be done manually, as it had been done until the middle of the last century, or automatically. Research into automatic text summarisation dates back to the 1950's (Luhn, 1958). Summaries that are extracts (i.e. not abstracts where the summary sentences are reconstructed grammatically) are usually created by assigning some scores to units that are to be extracted. Units are usually sentences or a series of adjacent words and expressions. They receive scores based on various unit features, such as length, location, number of query words, etc. The individual scores are then combined into a unit score and this score determines whether a unit is to be included in the document summary. A known and successful method for classifying sentences into summary-worthy and not-summary-worthy classes is presented in (Kupiec et al., 1995).

The structure of single documents can also be summarised, and this type of summarisation is the focus of this paper. Summarising the document structure is often done manually resulting in static tables of contents. For example, someone determines that it is sections and subsections that should be in the ToC, and this rule is applied no matter how long a document is, how rich and deep logical structure it has, etc. As manual textual summarisation evolved into automatic summarisation in the middle of the last century, ToC creation should also be done automatically. The static nature of manually created ToCs, or – more precisely – the vague definition of what should be in a ToC (i.e. is it sections and sub-sections to be included or sections only, etc.) has been found to be unsatisfactory in user studies carried out as part of the INEX Interactive Track (Malik et al., 2007), in the context of XML retrieval (Szlávik et al., 2006a). We also found in our study that a ToC should reflect the user's query and that it is not enough to determine ToC-worthiness only based on type (e.g. section, paragraph) of an XML element but other features, such as content length and depth in the structure, need also be considered. In other words, we found that there is a need for automatically identifying ToC-worthy elements, and for dynamically generating tables of contents for single documents.

Using text extraction methods as a basis, we proposed a ToC generation method in (Szlávik et al., 2007), where ToC-worthiness is determined by a score that is a linear combination of feature scores of the element. Scores are given based on element length, depth in the structure, and relevance to the query, and if the score is above a certain threshold the element's title or label is displayed in the document's ToC. As determined through a user study reported in the same paper, the importance of the relevance feature is high when creating tables of contents, thus the ToC is expected to be query-based. However, other features are also important in ToC generation, and so, features such as element length or depth cannot be ignored. In the work above, the threshold of ToC-worthiness and the individual weights of features were determined by the users themselves, a method which provided information about the importance of various features, including relevance. The aim of our current work is to determine these weights automatically. We also believe that several other features can also play important roles in determining whether an element is ToC-worthy and so the quality of automatic ToCs could be improved.

The following sections discuss how ToCs can be generated automatically (Section 3.1), which features might play important roles in structure summarisation (Section 3.2), how a structure summariser can be trained (Section 4), and how the quality of summaries can be measured using objective methods (Section 5). The quality of structure summaries will be measured against manually created ToC-worthy element sets.

3. Structure summarisation

In this section, we describe a basic structure summarisation method, and we propose a probabilistic structure summarisation method (Section 3.1). We used the former in (Szlávik et al., 2007) to explore what elements should be displayed in the tables of contents, i.e. what are the properties of ToC-worthy elements. Based on the outcomes of this earlier approach, in this paper we propose a method that is the adaptation of the method by Kupiec et al. (1995), a successful probabilistic text summarisation method. We also introduce the set of features used in the probabilistic structure summariser (Section 3.2).

3.1. Structure summarisation methods

We view structure summarisation to be closely related to extractive type text summarisation (Gupta and Lehal, 2010). Following this view, we assume that methods used in text summarisation can also be used to select XML elements that are later included in the tables of contents.

In (Szlávik et al., 2007), we presented a method based on early text summarisation methods, where the scores of individual features of an element are combined linearly (Equation 1).

$$S(e) = \sum_{f \in F} W(f) \cdot S_f(e) \tag{1}$$

where $S(e)$ denotes the overall score of element e, F is the set of features, $W(f)$ is the weight of feature f and $S_f(e)$ denotes the score that is given to element e based on feature f.

For the above method to work, one needs users to determine the weights, and also users are those who should determine a threshold value that is used to generate a score cut-off value. Elements with scores above this value are considered ToC-worthy while others are excluded from the ToC of the document. The above approach also requires an assumption, that is, if a subsection is found ToC-worthy then its parent section has to be ToC-worthy, too, no matter what its score is. Asking users to manually set weights is not an ideal solution, and the parent-child ToC-worthiness assumption might also not be entirely valid. One of the main aims of the work presented in this paper is to overcome these shortcomings.

The method proposed in this paper uses probabilistic element classification to extract the best elements that are worth including in a structure summary (ToC). Our method is based on the fundamental text summarisation method introduced by Kupiec et al. (1995) which uses a probabilistic framework to extract summary-worthy sentences in order to create text summaries of documents.

Our probabilistic structure summarisation method uses (Naïve) Bayesian classification as follows. For each XML element e, the probability that it is included in a structural summary T given k features f_j $(j = 1..k)$ is computed. This can be expressed using Bayes' rule as shown in Equation 2:

$$P(e \in T | f_1, f_2, .., f_k) = \frac{P(f_1, f_2, .., f_k | e \in T) \cdot P(e \in T)}{P(f_1, f_2, .., f_k)} \tag{2}$$

where $P(e \in T)$ denotes the probability that element e is ToC-worthy, $P(f_1, f_2, .., f_k | e \in T)$ is the probability of the k features being observed given that element e is ToC-worthy, and $P(f_1, f_2, .., f_k)$ is the probability of the k used features.

The Naïve Bayes assumption is then used, which assumes that the features are statistically independent with respect to ToC-worthiness (Equation 3). Note that although this assumption is clearly not accurate in many applications, including structure summarisation, the classification method works well, often outperforming more sophisticated classifiers on many datasets (Witten et al., 2011).

$$P(e \in T|f_1, f_2, .., f_k) = \frac{\prod_{j=1}^{k} P(f_j|e \in T) \cdot P(e \in T)}{\prod_{j=1}^{k} P(f_j)} \tag{3}$$

$P(e \in T)$ has the same value for each element. $P(e \in T)$, as well as $P(f_j|e \in T)$, which is the probability of the jth feature being observed given that element e is ToC-worthy, can be estimated directly from a training set. $P(f_j), i = 1..k$ does not need to be estimated because of the classification method used (see below in Equation 4). The probability estimation can be done by counting feature occurrences. The Bayesian classification function assigns a score for each element e which can be used to select elements for inclusion in a structural summary as described below.

The element selection (classification) function is shown in Equation 4. With the help of this function we can calculate whether the element is more likely to be ToC-worthy (ToC-worthy is one of the classes) or not-ToC-worthy (the other class of the classification). If the value obtained using Equation 4 (i.e. the element's score, the probabilistic log-odds of ToC-worthiness) is higher than zero, the element is more likely to be ToC-worthy than not-ToC-worthy, thus element e is included in the table of contents.

$$log \frac{P(e \in T|f_1 = b_1, f_2 = b_2, .., f_k = b_k)}{P(e \notin T|f_1 = b_1, f_2 = b_2, .., f_k = b_k)} = log \frac{\prod_{j=1}^{k} P(f_j = b_j|e \in T) \cdot P(e \in T)}{\prod_{j=1}^{k} P(f_j = b_j|e \notin T) \cdot P(e \notin T)} \tag{4}$$

where $P(e \notin T)$ denotes the probability that element e is not-ToC-worthy, and b_j is the value ('bucket' or 'bin') of feature f observed for element e. The use of logarithm makes it possible to express Equation 4 using sums (see Equation 5), which prevents representational issues potentially arising from multiplying very low fractions.

$$\sum_{j=1}^{k}(log(P(f_j = b_j|e \in T)) - log(P(f_j = b_j|e \notin T))) + log(P(e \in T)) - log(P(e \notin T)) \tag{5}$$

The above method allows that a user, to whom a ToC will be displayed, does not need to set weights and threshold values manually. Further advantages of this method are that, in addition to often being a machine learning algorithm to try first, it does not require a number of classifier parameters to be set/tuned (unlike other classification methods, such as decision trees, support vector machines, etc.), and the model it produces is easy to interpret (unlike a multi-layer neural network, for example). Nonetheless, other methods could be used, which we leave to future work.

To build models, the method requires a set of features and a set of training data. The following subsection describes the feature set considered in our work.

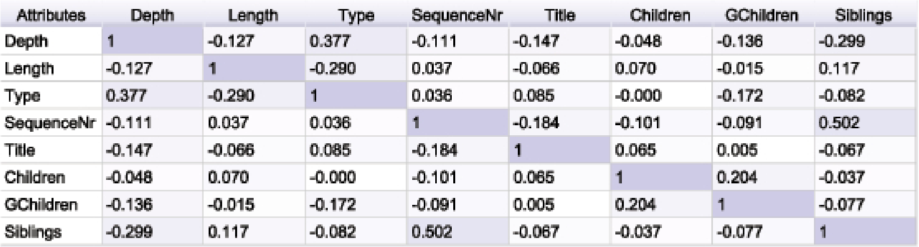
3.2. Features

In this section we present the set of features that is used in our work, together with the justification for their choice, as well as the chosen bins for each feature. We do not look at query-based features because we are focusing on generating tables of contents that are generally useful. Query-based features (such as relevance (Szlávik et al., 2007)) can be added to the list and their effect investigated in the future. The list of chosen features is the following:

1. **Depth of element in the logical structure.** We have found depth to be useful in previous work (Szlávik et al., 2007), and also, users indicated that the depth of an element in the logical structure was an important feature (Szlávik et al., 2006b). Ten discrete values of depth ('bins') are chosen, i.e. depth level 1, level 2, ..., level 10 can be considered. Previous analysis shows that from depth level 8 there are hardly any relevant or ToC-worthy elements (Hammer-Aebi et al., 2006).
2. **Length of the content of element.** Also found useful in previous work (Szlávik et al., 2007). In addition, it is an important feature in XML infor-

mation retrieval, where element length is used for normalisation (Kamps et al., 2004) and for filtering out elements that are too small to retrieve (Malik et al., 2005). The length categories used are determined by the lengths of elements in terms of characters: 1-10, 11-100, 101-1000, 1001-10000 and 10000+ bins are used.

3. **Type of element.** In an initial feature analysis (Szlávik, 2008, Ch. 6), it was found that four types of elements, namely `p` (paragraph), `section`, `article` and `body` tend to occur most frequently in high quality retrieval results. Note that this is a result that we believe would generalise with the type of XML documents typically investigated in content-oriented XML retrieval. As we view element retrieval and structure summarisation as related, elements of the above four types are likely to be helpful in deciding whether a particular element is ToC-worthy. The bins for element type are *top* and *non-top* where top is true if the current element's type is one of the above mentioned four types.
4. **Sequence number of the element.** For example, an element specified by the XPath expression `//section[3]` the number three is recorded, where `section[3]` identifies the third section of a number of sections having the same parent element. Similarly to the location method in text summarisation according to which for certain types of documents the first few sentences are more important than others (Paice, 1990), sequence number can also be useful in structure summarisation. The bins for the sequence number feature are the following: *seqnum1* (meaning that the current element's XPath is in the form `//*[1]`), *seqnum2-3* (`//*[2]` or `//*[3]`), *seqnum4-5* (`//*[4]` or `//*[5]`), *seqnum6+* (`//*[6]` or above). The bins are heuristically chosen based on preliminary analysis of the distribution of sequential numbers in the used data sets.
5. **Title.** The explicit presence of an element's title (e.g. a section title) is also considered as a feature. If an element has a title, the element might have different ToC-worthiness than otherwise. If an element has a title the *hastitle* bin's value is one (true), i.e. *hastitle* is a binary feature.



Attributes	Depth	Length	Type	SequenceNr	Title	Children	GChildren	Siblings
Depth	1	-0.127	0.377	-0.111	-0.147	-0.048	-0.136	-0.299
Length	-0.127	1	-0.290	0.037	-0.066	0.070	-0.015	0.117
Type	0.377	-0.290	1	0.036	0.085	-0.000	-0.172	-0.082
SequenceNr	-0.111	0.037	0.036	1	-0.184	-0.101	-0.091	0.502
Title	-0.147	-0.066	0.085	-0.184	1	0.065	0.005	-0.067
Children	-0.048	0.070	-0.000	-0.101	0.065	1	0.204	-0.037
GChildren	-0.136	-0.015	-0.172	-0.091	0.005	0.204	1	-0.077
Siblings	-0.299	0.117	-0.082	0.502	-0.067	-0.037	-0.077	1

Figure 2: Correlations between individual features.

6. **Number of child elements.** The number of descendants might also carry information about an element's ToC-worthiness. The following bins are assigned to this feature: *children0* (no descendants), *children1-5* (one to five child elements), *children6-10* and *children11+* (at least 11 direct descendants).
7. **Number of grandchild elements.** The same description and bin numbering applies to grandchild elements as to child elements.
8. **Number of sibling elements.** Defined as the number of child elements of the current element's parent, this feature might also contribute to distinguishing ToC-worthy elements from others. Bucket numbering is the same as for the previous two features.

The items of the above list were chosen heuristically, based on personal experience with the used document collection (described in Section 4.1) and with users of IR and summarisation experiments (Malik et al., 2006; Szlávik et al., 2007). Also, an attempt was made to capture as different aspects of ToC-worthiness as possible. The correlation matrix of our attributes (Figure 2) shows that, indeed, there tends to be low correlation between attributes, which indicates that using the Naïve Bayes method to build structure summaries was appropriate.

There are several other types of features that could be investigated (such as query-based features, link information within an element's content, etc.) which we leave to future work. In the next sections, we describe how to estimate the various probabilities described previously in this paper, and then we describe the evaluation methodology followed in our work.

4. Training the structure summariser

In this section we introduce the document collection used, and discuss how our structure summariser can be trained.

4.1. Document collection

There are several XML document collections that can be considered for the ToC generation task. The most frequently used collections in structured document retrieval experiments are the IEEE, Wikipedia and Lonely Planet collections. These can also be used when investigating ToC generation. The logical structure of these collections are relatively similar to one another. For example, when investigating the usefulness of three element features in structure summarisation we found that users generated ToCs that were not significantly different when using the IEEE or Wikipedia collections (Szlávik et al., 2007). The Wikipedia collection (Denoyer and Gallinari, 2006), that is used in the work presented in this paper, is an XML version of the Wikipedia[2] articles. The collection was chosen because it has been used more recently in XML retrieval research (in the context of INEX (Fuhr et al., 2008)), and also, because the documents are accompanied by retrieval result sets (runs) and relevance assessments, which are used in the evaluation of our structure summaries (see next section). The document collection consists of the full-texts, marked-up in XML, of 659,388 articles of the Wikipedia project, and totalling more than 60 GB (4.6GB without images) and 30 million in number of elements[3]. In the next section, we describe the training data sets in relation to the documents of the above collection.

[2] http://wikipedia.org

[3] More recent versions of the collection have also been used in INEX (Geva et al., 2011), though our work had been carried out before these became available.

4.2. Training data

According to the summarisation method proposed in this paper, after the set of features is selected, their weights need to be determined. The weights can be expressed in terms of probabilities, as described in Section 3.1. These weights are determined in an automatic manner through training. Sections 4.2.1, 4.2.2 and 4.2.3 describe three sets of training data that are used to train the structure summariser.

4.2.1. Manually created ToCs

Ideally, a training set for structure summarisation would consist of a series of structure summaries. For the Wikipedia collection, however, such tables of contents were not available, or when they were they consisted of elements selected by their type or depth only (We will use such ToCs as baselines for our experiments). To be able to investigate the effectiveness of several features and feature combinations, a set of example ToCs needs to be created. Therefore, we recruited 25 users with various levels of computer science experience. Their task was to select ToC-worthy elements for up to 20 documents. These elements were used as positive examples of ToC-worthiness, while other elements from the same document were considered as negative examples. Documents were randomly selected from the collection and assigned to each user. The interface shown in Figure 3 was used. Through this experiment, we acquired ToC-worthy element sets for 322 documents. The number comes from that fact that users were allowed to quit the experiment at any time for any reason. For further details of this experiment, including an analysis of agreement levels between users, see (Szlávik, 2008, Ch. 6).

We would like to emphasise that user involvement here is different from that described in (Szlávik et al., 2007): here we use users only to create a data set for training and evaluation, and we do not ask every individual who uses the summariser to set some weights manually. Once the data set is used for training a particular summariser, no further user involvement is needed, unless one incorporates query-based or user-activity (e.g. click) based features into the

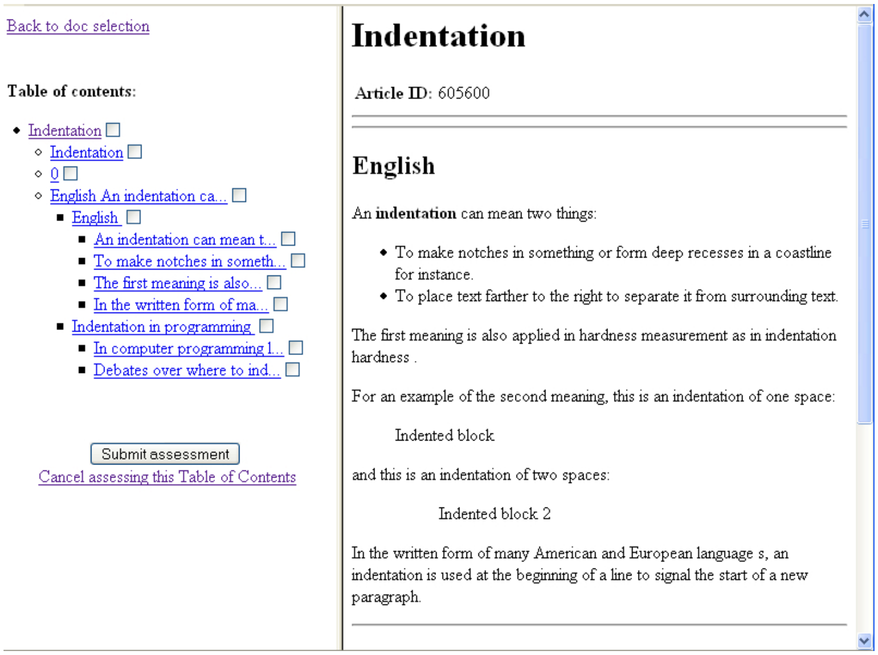


Table of contents:

- Indentation □
  - Indentation □
  - 0 □
  - English An indentation ca... □
    - English □
      - An indentation can mean t... □
      - To make notches in someth... □
      - The first meaning is also... □
      - In the written form of ma... □
    - Indentation in programming □
      - In computer programming l... □
      - Debates over where to ind... □

Indentation

Article ID: 605600

English

An **indentation** can mean two things:

- To make notches in something or form deep recesses in a coastline for instance.
- To place text farther to the right to separate it from surrounding text.

The first meaning is also applied in hardness measurement as in indentation hardness .

For an example of the second meaning, this is an indentation of one space:

 Indented block

and this is an indentation of two spaces:

  Indented block 2

In the written form of many American and European language s, an indentation is used at the beginning of a line to signal the start of a new paragraph.

Figure 3: The manual ToC building interface.

summariser, which are beyond the scope of this paper.

Creating a training data set manually is an expensive and time consuming task. Also, the number of 322 ToCs (that would need to be used both for training is testing) is relatively low. We therefore look at alternative data sets with more documents to train the summariser. We discuss these in Sections 4.2.2 and 4.2.3. In our work, the manually created ToCs are also used to evaluate the "quality" of these alternative sets. The sizes of training data sets are shown in Table 1.

manual ToC training	run training	assessment training
322	7438	5460

Table 1: The size (number of documents) of each training data set.

4.2.2. Creating ToCs using retrieval runs

In our work, we assume that the averages of properties of relevant elements over a number of queries (in IR) can predict ToC-worthiness (in structure summarisation). For example, in relation to the length feature, if a high portion of relevant elements (over a number of queries) are longer than 10,000 characters, then an arbitrary element of an arbitrary document from the collection is more likely to be ToC-worthy if its textual content is longer than 10,000 characters. Based on this assumption, XML elements that are returned in high quality retrieval results can be used to estimate ToC-worthiness, i.e. to train a structure summariser. Retrieval runs were chosen as an alternative training data set because, unlike example ToCs, a high number of retrieval runs are available for the Wikipedia collection.

The training method described in this subsection takes the INEX 2006 "Relevant In Context" retrieval result set's high quality runs as training examples (containing 7438 documents)(Kamps et al., 2007). The INEX Relevant In Context task "required systems to return for each article an unranked set of non-overlapping elements, covering the relevant material in the document" (Lalmas and Tombros, 2007). The runs of this task are chosen because they consist of sets of elements for each document, which is very similar to sets of example ToC-worthy elements. Elements listed in chosen runs are considered to be positive examples of ToC-worthiness for the corresponding documents, while not listed elements from the same document are treated as not-ToC-worthy. Non-overlapping elements in the data set are not considered to be an issue because of the training method and the number of documents in the set. High quality runs, that received the highest evaluation scores at INEX, were considered for training in order to maximise the quality of generated ToCs.

4.2.3. Creating ToCs using IR relevance assessments

Another alternative training set to the manually created ToCs, and that presented in the previous section, is obtained by analysing IR relevance assessments in which element relevance values (in response to a query) are determined by human assessors. To follow the assumption introduced in Section 4.2.2, XML elements that are assessed relevant at INEX can be used to estimate ToC-worthiness.

The assessments set used is that of INEX 2006, where documents for 114 topics have been assessed. This gives 5460 documents with relevant elements.

Using the above three training data sets, i.e. the manual ToC set, retrieval run set, and assessment set, the probabilities shown in Tables 2 and 3 can be obtained. As Table 2 shows, using the run training, an average of 2.35% of a document's elements are ToC-worthy. This number is lower than that of 8.16% we found previously in (Szlávik et al., 2006a), while the corresponding values for the assessment and manual ToC trainings are closer to 8.16%. The above low number might indicate that run training might result in ToCs with low recall values which we will examine in Section 6.

run training		assessment training		manual ToC training	
$P(e \in T)$	$P(e \notin T)$	$P(e \in T)$	$P(e \notin T)$	$P(e \in T)$	$P(e \notin T)$
0.0235	0.9765	0.0900	0.9100	0.0664	0.9335

Table 2: The calculated values of $P(e \in T)$ and $P(e \notin T)$ for the three training methods.

The other training probabilities are shown in Table 3, where f_{ij} denotes the jth bin of feature i. (The meaning of numbers in bold will be explained in Section 6.2.)

5. Evaluation

In this section, we describe the evaluation methods adopted in this paper for evaluating structure summariser versions, as well as the evaluation measures used.

5.1. Evaluation methods

For training using retrieval runs (Section 4.2.2) and relevance assessments (Section 4.2.3) the holdout method is used while cross-validation is employed

	run training		assessment training		manual ToC training	
f_{ij}	$P(f_{ij} \mid e \in T)$	$P(f_{ij} \mid e \notin T)$	$P(f_{ij} \mid e \in T)$	$P(f_{ij} \mid e \notin T)$	$P(f_{ij} \mid e \in T)$	$P(f_{ij} \mid e \notin T)$
depth_1	**0.1292**	**0.0017**	**0.0464**	**0.0000**	**0.1321**	**0.0022**
depth_2	0.0379	0.0134	0.0610	0.0078	0.0711	0.0296
depth_3	0.3830	0.0553	0.1302	0.0524	0.4828	0.1036
depth_4	0.2650	0.1766	0.2316	0.1709	0.2161	0.2708
depth_5	0.1401	0.2586	0.2548	0.2759	0.0810	0.2892
depth_6	0.0328	0.2741	0.1572	0.2986	0.0085	0.1926
depth_7	0.0069	0.1497	0.0791	0.1297	0.0085	0.0862
depth_8	0.0007	0.0417	0.0200	0.0353	0.0000	0.0191
depth_9	0.0006	0.0110	0.0039	0.0111	0.0000	0.0066
depth_10	0.0005	0.0038	0.0025	0.0052	0.0000	0.0001
type_top	0.8669	0.0745	0.3074	0.0695	0.8716	0.0701
type_nontop	0.1331	0.9255	0.6926	0.9305	0.1284	0.9299
seqnum_1	0.4760	0.4978	0.5259	0.4755	0.4933	0.5152
seqnum_2-3	0.2695	0.2186	0.2378	0.2266	0.2947	0.2246
seqnum_4-5	0.1212	0.0925	0.1033	0.1033	0.1234	0.0973
seqnum_6+	0.1333	0.1910	0.1329	0.1946	0.0886	0.1628
length_0-10	0.0001	0.3124	0.2373	0.3324	0.0121	0.3225
length_10-100	0.0234	0.5798	0.3885	0.5657	0.1557	0.5883
length_100-1000	0.6806	0.0896	0.2326	0.0886	0.5335	0.0790
length_1000-10000	0.2664	0.0162	0.1123	0.0131	**0.2816**	**0.0094**
length_10000+	0.0295	0.0021	**0.0293**	**0.0003**	**0.0171**	**0.0009**
title_yes	0.4198	0.0277	0.1267	0.0262	**0.5966**	**0.0082**
title_no	0.5802	0.9723	0.8733	0.9738	0.4034	0.9918
children_0	0.0000	0.0095	0.0090	0.0043	0.0000	0.0040
children_1-5	0.3734	0.9153	0.7181	0.9213	0.4584	0.9373
children_6-10	0.3253	0.0438	0.1451	0.0407	0.3255	0.0300
children_11+	0.3012	0.0314	0.1278	0.0337	0.2161	0.0287
siblings_0	**0.1292**	**0.0017**	**0.0464**	**0.0000**	**0.1321**	**0.0022**
siblings_1-5	0.1895	0.3940	0.3698	0.3670	0.1738	0.3943
siblings_6-10	0.1905	0.1970	0.2064	0.1952	0.2105	0.1849
siblings_11+	0.4908	0.4073	0.3773	0.4379	0.4837	0.4187
grandchildren_0	0.0703	0.6781	0.3889	0.7099	0.0660	0.7326
grandchildren_1-5	0.2944	0.2554	0.3420	0.2254	0.2761	0.2198
grandchildren_6-10	0.1877	0.0288	0.0812	0.0322	0.1813	0.0218
grandchildren_11+	0.4476	0.0377	0.1879	0.0324	**0.4766**	**0.0258**

Table 3: Individual features' probability distributions.

for the training with manual ToC sets (Section 4.2.1) (Kohavi, 1995). The holdout method involves using one set of documents for training and another set for testing or evaluation. The data set used for evaluation is the set of the 322 manually created ToCs introduced earlier. The document sets used both in training by run and assessment data are disjoint from the set of documents used for evaluation, to avoid any unwanted effect of a document being in both sets.

As the training and evaluation data sets are the same in case of summarisation using manual ToC training (due to the set's high creation costs), another evaluation methodology should be followed. To maximise efficiency, robustness and reliability, the k-fold cross validation method (Kohavi, 1995) is used. This method splits the document set into two parts, uses one part for training and the other for evaluation, then the initial document set is split again and the newly obtained two parts are used as previously. This procedure is repeated k times and the average of evaluation results from each 'fold' are averaged to obtain overall evaluation scores. Research shows that the choice of $k = 10$ gives one of the most reliable results (Kohavi, 1996), hence $k = 10$ is used in this paper as well.

5.2. Evaluation measures

To evaluate structure summarisation results (i.e. ToC-worthy elements selected for XML documents), the measures of recall and precision are used. Recall and precision are widely used in text summarisation (Mani, 2001), and – because of the extractive nature of our summariser – adopting content based text summarisation evaluation measures such as ROUGE (Lin, 2004) is not needed. To calculate recall and precision values, macro evaluation is used:

$$R = \frac{1}{N} \cdot \sum_{i=1}^{N} \frac{|ToCworthy_i \cap Selected_i|}{|ToCworthy_i|} \tag{6}$$

$$P = \frac{1}{N} \cdot \sum_{i=1}^{N} \frac{|ToCworthy_i \cap Selected_i|}{|Selected_i|} \tag{7}$$

where N is the number of documents whose ToCs are used in evaluation, $Selected_i$ is the set of elements in document i that are selected by the summariser being evaluated and $ToCworthy_i$ denotes elements that are selected manually for document i by participants of manual structure summary building.

It is often desirable to use only one number to describe the performance of a system. This way, various methods can directly be compared and a ranked order easily obtained. Thus, results by the F_2 measure (Equation 8), that places double emphasis on recall, are also reported in addition to recall and precision values.

$$F_2 = \frac{1}{\frac{0.25}{P} + \frac{0.75}{R}} = \frac{(2^2 + 1) \cdot (P \cdot R)}{(2^2 \cdot P + R)} \tag{8}$$

This measure is chosen because we found that user would rather tolerate not-ToC-worthy elements displayed (and possibly ignores them) than accept if important (i.e. ToC-worthy) elements are omitted from the structural overview (Szlávik, 2008, Ch. 4). This finding is a clear indication that recall should be favoured over precision.

6. Results and discussion

This section discusses the evaluation results of our proposed structure summarisation method. First, we introduce two baseline functions that allow us to compare our summariser with static ways of building ToCs, and present results based on them. Then we provide structure summarisation results that consider features individually, which is followed by results and their discussions obtained when features are combined and various training methods are used.

6.1. Baseline results

In order to compare our results with traditional ToC creation practice, we defined two baseline functions. The first, named *DepthMax3*, considers a situation in which every element up to depth level three[4] is considered ToC-worthy, and anything deeper as not-ToC-worthy. This function would produce an exactly three level deep ToC. The other baseline function, named *SectionsAndArticle*, identifies section and article elements as ToC-worthy, and every other type of element as not-ToC-worthy. The assumption behind it is, naturally, that it is only sections and the root element (a necessity with XML documents) that should be shown in a table of contents.

The results based on the baseline functions are shown in Table 4. It is clear that a fixed depth ToC brings unacceptably low precision, showing that, in many cases, deeper elements are also ToC-worthy, or elements not deep in the hierarchy are not-ToC-worthy. Also, a sections-and-articles-only approach produces considerably low recall, despite high precision scores. This shows that users indeed find sections and articles ToC-worthy, but several other types of elements should also be displayed in a ToC, as indicated by a relatively low recall score. Furthermore, subsequent subsections of this paper will demonstrate that our proposed method can easily outperform these two baselines in terms of the F_2 measure (our focus measure).

Baseline method	recall	precision	F_2
DepthMax3	0.8825	0.3499	0.6737
SectionsAndArticle	0.6162	0.9060	0.6576

Table 4: Baseline evaluation scores.

6.2. Individual features

Individual features generally do not perform particularly well. Similarly to text summarisation (and other areas where learning is used), combinations of features rather than individual features, should be used to yield better performance (Edmundson, 1969; Kupiec et al., 1995). However, it is still worth

[4] Apart from this level having been found most important, our results show that versions of the same function with other depth levels perform significantly worse.

studying individual features as their discrimination power might suggest which features are more important than others, which ones are more worth selecting for a summariser that uses feature combinations.

The estimated probability values of various features and bins are shown in Table 3 where some values are marked bold. The features corresponding to the marked numbers are those that can produce non-empty ToCs on their own, i.e. their discrimination power is high enough to allow at least some of the elements to be classified as ToC-worthy. The ratio of the probability of ToC-worthiness ($P(f_{ij}|e \in T)$) to the probability of non-ToC-worthiness ($P(f_{ij}|e \notin T)$) has to be high because of the generally low ratio of ToC-worthy elements (Table 2) in the training set. As Table 3 shows, only few of the features can produce non-empty ToCs on their own. For instance, when using run training (Table 3, columns 2-3, , see features whose corresponding probability values are marked bold), an element can only be ToC-worthy if it is at depth level one or it has no siblings. This would, most of the time, return the article element when the any of these two features are used. Returning the article only can be acceptable for very short documents but not for all documents in the collection, as only one item in the ToC cannot really be considered as an overview of the logical structure. With respect to assessment training, an element can also be ToC-worthy when its textual content is longer than 10,000 characters (Table 3, middle columns). With manual ToC training, an element is also identified as ToC-worthy if it has a title, has no siblings, has more than 11 grandchild elements or its corresponding text is longer than 1000 characters.

The evaluation results when using individual features only are shown in Table 5, where the IDs of the summariser versions are made up of two character groups as follows: the training used is denoted by either R (run), A (assessment) or M (manual); the second group of characters identifies the used features (D - depth, Ty - type, Se - sequence number, L - length, Ti - title, C - children, Si - siblings, G - grandchildren). It seems that the presence of a title (Ti) can lead – on its own – to reasonably good results, however, only in the case of training by manual ToC sets. Indeed, if an author gives a title to a piece of text they probably want that part of the document mentioned in the corresponding ToC. For the other training types, title information is not discriminative enough to classify any element as ToC-worthy, however it might still be a useful feature in combination with others, which is also going to be examined in the next section.

ID	recall	precision	F_2
M_Ti	**0.6727**	**0.8331**	**0.7067**
M_G	0.5480	0.5607	0.5512
A_D	0.2895	0.8136	0.3452
A_Si	0.2895	0.8136	0.3452
M_D	0.2895	0.8136	0.3452
M_Si	0.2895	0.8136	0.3452
R_D	0.2895	0.8136	0.3452
R_Si	0.2895	0.8136	0.3452
M_L	0.2973	0.4609	0.3263
A_L	0.0058	0.0418	0.0074

Table 5: Individual features' evaluation scores.

6.3. Feature combinations and training methods

This section presents and discusses summariser evaluation results when combinations of features are considered. We also discuss the effectiveness of the three training methods used. The evaluation results are presented in Table 6.

The top 15 summariser versions with respect to recall are shown in Table 6(a), together with their recall scores. The best summariser in this sense is *A_DTyLSeTiCG* which is the summariser version that is trained by retrieval assessments (hence the *A* in the ID) and uses all features but the number of siblings (only *Si* is missing from the feature identifiers from the summariser ID). Table 6(a) shows that to obtain high recall not less than four features need to be incorporated into the summariser. We can also observe that the type (Ty) and length (L) features are present in all 15 summariser version listed in Table 6(a) which shows that these features are important to use if the goal is to achieve high recall. As we can also see, the assessment trainings (ID-s starting with an

A in Table 6(a)) usually outperform the manual and run trainings for recall. This shows that manual ToCs training can indeed be substituted with another type of training if high recall is to be achieved.

The top 15 summariser versions with respect to precision are listed in Table 6(b) where we can observe the following: Firstly, none of the training methods used (denoted by R,A,M, respectively) dominate the list of top 15 summarisers which shows their comparability, i.e. training with assessments and runs can produce comparable results to those obtained by training with manual ToCs. Secondly, all the summariser versions in the top 15 use at most four features. For example, the top summariser with respect to precision uses the depth feature (D), the sequence number (Se), title information (Ti) and number of sibling elements (Si). This shows that to achieve high precision, the number of features used should be low. Also, we can see in the second column of Table 6(b) that eleven summariser versions have exactly the same precision values. As further examination showed that their corresponding recall values are also the same, it is suspected that the outputs of these summarisers are exactly the same ToC-worthy element sets. In addition to these eleven summarisers, the remaining four of the top summarisers with respect to precision also use the title feature (Ti). This shows that title information is very effective to obtain high precision, and – because of the eleven equal scores – it is a particularly dominant feature in our feature set. Table 6(b) further shows that the top four summarisers also use the depth (D) feature. Thus, in order to achieve high precision, the title and depth features should be included in a summariser, and an additional one or two other features might be also used.

Table 6(c) shows the top 15 summarisers with respect to the F_2 measure. As we can see from the listed ID-s, none of the summarisers listed in Tables 6(a) and (b) are listed in Table 6(c), which shows the trade-off between recall and precision. The first letters of the ID-s in Table 6(c) show that the effectiveness of the three used training methods are comparable. This means that training with manual ToCs can be generally substituted by training with assessments or runs, hence, there is no need to create expensive data sets specifically for structure summarisation but other, existing, data sets (from IR) can be used.

ID	recall	ID	precision	ID	F_2
A_DTyLSeTiCG	0.8812	A_DSeTiSi	0.8440	R_DTySeTiSiG	0.7613
A_DTyLSeTiSiG	0.8768	R_DSeTiSi	0.8414	M_DTyLTiSi	0.7589
A_DTyLSeTiCSiG	0.8762	R_DLTi	0.8410	R_DTyLTiSi	0.7555
A_DTyLTiCSiG	0.8746	M_DSeTiSi	0.8347	M_DTyLTi	0.7507
A_TyLSeTiCSiG	0.8720	A_TySeTiSi	0.8331	R_DTyLTi	0.7492
A_DTyLTiSi	0.8713	M_TySeTiSi	0.8331	M_DTyLSeTiSiG	0.7480
M_DTySeL	0.8710	R_TySeTiSi	0.8331	M_DTyLSeTiCSi	0.7477
A_DTyLTi	0.8686	A_TySeTi	0.8331	M_DSeTiCSiG	0.7477
M_DTyLTi	0.8683	A_TyTiSi	0.8331	A_DTySeTiSi	0.7472
M_DTyLSeTiCG	0.8677	M_SeTiSi	0.8331	A_DTyLTi	0.7471
M_DTyLSeTiSiG	0.8674	M_TySeTi	0.8331	R_DTySeTiCSiG	0.7470
A_DTyLSeTiCSi	0.8673	M_TyTiSi	0.8331	M_DTySeTiCSiG	0.7466
M_DTyLSeTiCSi	0.8662	R_TySeTi	0.8331	A_DTyLTiSi	0.7463
M_DTyLTiCSiG	0.8646	R_TyTiSi	0.8331	R_DTyLSeTiSiG	0.7460
R_DTyLTi	0.8643	M_Ti	0.8331	A_DTyTiSi	0.7434
(a)		**(b)**		**(c)**	

Table 6: Top 15 feature combinations by recall (a), precision (b) and the F_2 measure (c).

Our work shows that, generally, individual features do not perform particularly well, which is in accordance with findings in text summarisation (Kupiec et al., 1995; Fattah and Ren, 2009). However, we have identified particularly important features, i.e. the type and length features for recall-oriented summarisers, and the title and depth features for precision-oriented summarisers.

We have also found that for high recall, more than four features need to be used, however, to obtain high precision scores, not more than four features should be incorporated into the summariser. The number and particular choice of features to be used, therefore, depends on what is to be emphasised (i.e. precision or recall). In this work, we have also used the F_2 measure for evaluation which combines recall and precision. Accordingly, the top summariser versions with respect to the F_2 measure (Table 6(c)) use between four and seven features (combination of what we have found desirable for high precision and recall), and tend to include features that have been found important in order to obtain either high recall or precision, i.e. the title and depth, type, and length features (discussed in previous paragraphs). In addition, there is one feature that appears to be increasingly important when the F_2 evaluation measure is used: the number of siblings (Si). The number of siblings appears in 12 out of 15 top summarisers in Table 6(c) (F_2 measure), as opposed to lower occurrences in Tables 6(a) and (b) (recall and precision, respectively). It seems that the siblings feature captures something that leads to high evaluation scores when recall and precision are combined. The above results show that, depending on how effectiveness is measured, or what combinations of measures are used in evaluation, different features might emerge as highly effective ones. Hence, it is important to determine the purpose of the summariser, i.e. what is meant to be "effectiveness". This could be determined via user studies which we consider for future work.

To examine various features and their effects, we propose that several data sets from IR can be (re)used. We have found that the effectiveness of summarisers trained using retrieval runs and assessments (which are available in IR test collections) are comparable to that of summarisers trained using manual ToC sets (which are expensive to create). With the use of already existing data sets, the investigation of various structure summarisers can be made quicker and cheaper.

7. Conclusions and future work

In this paper, we have investigated a method for the automatic summarisation of the logical structure of documents. The proposed summariser selects elements of XML documents that are worth displaying in tables of contents. We have presented and studied several element features that have been used in the summariser to identify ToC-worthy elements. We have also shown that automatically generated structure summaries easily outperform traditional ways of creating them (baselines). Despite the fact that, as a feature, relevance to a query is considered highly important by users (Szlávik et al., 2007; Kazai and Trotman, 2007), in this work, we have considered query-independent features only. We believe that adding query-dependent features would significantly improve the quality and usefulness of ToCs in the searching process of users; however, we also believe that establishing which query-independent features play important roles in the identification of ToC-worthy elements has also high importance. We propose that the use of query-independent features should serve as basis for structure summarisation, while query-dependent (relevance-oriented) features should be used to refine the ToCs and tailor them to the user's information need.

We have also shown in this paper, that it is possible to effectively (re)use several data sets from information retrieval evaluation for training a structure summariser. The quality of summarisers trained with the alternative data sets have matched the quality of those trained using the data set created specifically for the purpose of structure summarisation. With the use of these alternative data sets, results obtained in further investigations of features become more reliable as the whole data set containing example ToCs can be used for evaluation purposes only.

A possible future step in this research can be the thorough investigation of the impact of structure summarisation in an information seeking setting. We believe that the appropriate use (display) of structure summaries will enhance user experience, bring increased user satisfaction, and users will also find relevant content more efficiently as well as effectively.

8. Acknowledgements

This work was partly funded by the Nuffield Foundation (grant NAL/01081/G) and the DELOS Network of Excellence in Digital Libraries, to which we are very grateful. We are especially thankful to participants of the manual ToC creation study for their precious time and great effort. We would also like to acknowledge the reviewers for their useful comments that have helped us improving the paper.

References

C. Carpineto, S. Osiński, G. Romano, and D. Weiss. A survey of web clustering engines. *ACM Comput. Surv.*, 41:17:1–17:38, July 2009. ISSN 0360-0300.

L. Denoyer and P. Gallinari. The Wikipedia XML Corpus. *SIGIR Forum*, 40 (1):64–69, 2006.

H. P. Edmundson. New methods in automatic extracting. *J. ACM*, 16(2): 264–285, 1969. ISSN 0004-5411.

M. A. Fattah and F. Ren. GA, MR, FFNN, PNN and GMM based models for automatic text summarization. *Computer Speech & Language*, 23(1):126–144, 2009.

N. Fuhr, J. Kamps, M. Lalmas, and A. Trotman, editors. *Focused Access to XML Documents, 6th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2007, Dagstuhl Castle, Germany, December 17-19, 2007. Selected Papers*, volume 4862 of *Lecture Notes in Computer Science*, 2008. Springer.

S. Geva, J. Kamps, R. Schenkel, and A. Trotman, editors. *Comparative Evaluation of Focused Retrieval : 9th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX 2010)*, volume 6932 of *LNCS*, 2011. Springer.

K. Großjohann, N. Fuhr, D. Effing, and S. Kriewel. Query formulation and result visualization for XML retrieval. In *Proceedings ACM SIGIR 2002 Workshop on XML and Information Retrieval.* ACM, 2002.

V. Gupta and G. Lehal. A survey of text summarization extractive techniques. *Journal of Emerging Technologies in Web Intelligence*, 2(3), 2010.

B. Hammer-Aebi, K. W. Christensen, H. Lund, and B. Larsen. Users, structured documents and overlap: interactive searching of elements and the influence of context on search behaviour. In *Proceedings of IIiX*, pages 46–55, 2006. ISBN 1-59593-482-0.

M. A. Hearst. TileBars: visualization of term distribution information in full text information access. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 59–66, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co. ISBN 0-201-84705-1.

J. Kamps, M. de Rijke, and B. Sigurbjörnsson. Length normalization in XML retrieval. In *Proceedings of ACM SIGIR*, pages 80–87, 2004. ISBN 1-58113-881-4.

J. Kamps, M. Lalmas, and J. Pehcevski. Evaluating relevant in context: Document retrieval with a twist (poster). In *ACM SIGIR Conference on Research and Development in Information Retrieval*, Amsterdam, The Netherlands, July 2007.

G. Kazai and A. Trotman. Users' perspectives on the usefulness of structure for XML information retrieval. In *Proceedings of the 1st International Conference on the Theory of Information Retrieval*, 2007.

R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, pages 1137–1145, 1995.

R. Kohavi. *Wrappers for performance enhancement and oblivious decision graphs.* PhD thesis, Stanford, CA, USA, 1996.

J. Kupiec, J. Pedersen, and F. Chen. A trainable document summarizer. In *SIGIR'95*, pages 68–73. ACM Press, 1995. ISBN 0-89791-714-6.

M. Lalmas and R. Baeza-Yates. Structured document retrieval. In O.M. Tamer and L. Ling, editors, *Encyclopedia of Database Systems.* Springer, May 2009.

M. Lalmas and A. Tombros. Evaluating XML retrieval effectiveness at INEX. *SIGIR Forum*, 41(1):40–57, 2007. ISSN 0163-5840.

D. J. Lawrie. *Language models for hierarchical summarization.* PhD thesis, University of Massachusetts Amherst, 2003. Director-W. Bruce Croft.

C-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In *ACL-WS2004A*, 2004.

H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165, 1958. ISSN 0018-8646.

S. Malik, G. Kazai, M. Lalmas, and N. Fuhr. Overview of INEX 2005. In *Advances in XML Information Retrieval and Evaluation, 4th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2005, Dagstuhl Castle, Germany, November 28-30, 2005, Revised Selected Papers*, pages 1–15, 2005.

S. Malik, C.-P. Klas, N. Fuhr, B. Larsen, and A. Tombros. Designing a user interface for interactive retrieval of structured documents - lessons learned from the INEX interactive track. In *Proceedings of ECDL 2006*, pages 291–302, 2006.

S. Malik, B. Larsen, and A. Tombros. Report on the INEX 2005 interactive track. *SIGIR Forum*, 41(1):67–74, 2007. ISSN 0163-5840.

I. Mani. Summarization evaluation: An overview. In *NAACL-WS2001A*, 2001.

D. A. Nation. WebTOC: a tool to visualize and quantify web sites using a hierarchical table of contents browser. In *CHI '98: CHI 98 conference summary on Human factors in computing systems*, pages 185–186, New York, NY, USA, 1998. ACM. ISBN 1-58113-028-7.

A. Nenkova and K. McKeown. Automatic summarization. *Foundations and Trends in Information Retrieval*, 5(2-3):103–233, 2011.

C. D. Paice. Constructing literature abstracts by computer: Techniques and prospects. *Information Processing and Management*, 26(1):171–186, 1990.

M. M. Sebrechts, J. Cugini, S. J. Laskowski, J. Vasilakis, and M. S. Miller. Visualization of search results: A comparative evaluation of text, 2D, and 3D interfaces. In *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 15-19, 1999, Berkeley, CA, USA*, pages 3–10. ACM, 1999.

Z. Szlávik. *Content and structure summarisation for accessing XML documents*. PhD thesis, Department of Computer Science, Queen Mary, University of London, 2008.

Z. Szlávik, A. Tombros, and M. Lalmas. The use of summaries in XML retrieval. In *Proceedings of ECDL 2006*, pages 75–86, 2006a.

Z. Szlávik, A. Tombros, and M. Lalmas. Investigating the use of summarisation for interactive XML retrieval. In F. Crestani and G. Pasi, editors, *Proceedings of ACM SAC-IARS'06*, pages 1068–1072, 2006b.

Z. Szlávik, A. Tombros, and M. Lalmas. Feature- and query-based table of contents generation for XML documents. In G. Amati, C. Carpineto, and G. Romano, editors, *ECIR*, volume 4425 of *Lecture Notes in Computer Science*, pages 456–467. Springer, 2007. ISBN 978-3-540-71494-1.

A. Tombros and M. Sanderson. Advantages of query biased summaries in information retrieval. In *Proceedings of ACM SIGIR*, pages 2–10, 1998. ISBN 1-58113-015-5.

I. H. Witten, E. Frank, and M. A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, Amsterdam, 3rd edition, 2011. ISBN 978-0-12-374856-0.