Relaxed Global Term Weights for XML Element Search

Atsushi Keyaki[1,⋆], Kenji Hatano[2], and Jun Miyazaki[3]

[1] Graduate School of Culture and Information Science, Doshisha University
1-3 Tatara-Miyakodani, Kyotanabe, Kyoto 610-0394, Japan
keyaki@ilab.doshisha.ac.jp
[2] Faculty of Culture and Information Science, Doshisha University
1-3 Tatara-Miyakodani, Kyotanabe, Kyoto 610-0394, Japan
khatano@mail.doshisha.ac.jp
[3] Graduate School of Information Science, Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-0192, Japan
miyazaki@is.naist.jp

Abstract. XML element search engines return XML elements which are part of XML documents as search results. Existing studies related to XML element search are brought from the information retrieval techniques for document search. There are some ways to calculate global weights of each term from statistics of XML elements with 1) the same path expression or 2) the same tag. In the first approach, the more complex a path expression is, the less the number of XML elements with the path expression becomes. This is a problem that global term weights may be calculated using statistics of a few XML elements. Such global weights are never *global*. The second approach also has a problem that it does not consider document structures of XML elements. To resolve the problems, we propose a method for calculating accurate global weights. In our method, we regard a path expression as an array of tags. We relax the restriction of appearance order and appearance frequency of tags in a path expression to gather similar path expressions into the same class. Therefore, we try to decrease the number of classes which hardly contain elements. Our experimental results show that our method can integrate path expressions without decreasing search accuracy with a certain test collection.

Keywords: XML element search, accurate global term weights, classification of similar path expressions.

1 Introduction

The XML[1] is a markup language for structured documents that has become the de facto format for data exchange. A large number of XML documents are now available on the Web, so that it continues to be used in the future.

⋆ Current affiliation: Graduate School of Information Science, Nara Institute of Science and Technology.

[1] http://www.w3.org/TR/REC-xml/

S. Geva et al. (Eds.): INEX 2010, LNCS 6932, pp. 71–81, 2011.

An XML element is usually defined as a part of an XML document. The XML element is identified by the surrounding start and end tags. The key goal of XML search is to obtain relevant XML elements to a query instead of just returning the entire XML documents, in particular, with document-centric XML documents [2]. Therefore, XML search engines can generate a ranked list composed of a set of relevant XML elements, while several Web search engines return a set of entire Web documents. By searching XML elements, users do not need to identify relevant descriptions from entire XML documents.

In the field of XML element search, existing term weighting schemes are often brought from the information retrieval techniques of document search. There are two approaches which are often used to calculate global weights of each term. One uses statistics of XML element with the same path expression, while the other uses the statistics of the same tag. In the first approach, the more complex a path expression is, the less the number of XML elements with the path expression becomes. This is one of the problems that global weights may be calculated using statistics of a few elements. Such global weights are not *global*. The second approach also has a problem that it does not consider document structure of XML elements.

To resolve these problems, we should calculate global weights for each term from statistics of XML elements which are *global* enough with considering structures of the XML elements. Therefore, we integrate similar path expressions into the same class and calculate global weights based on the classes. In this paper, we define a *class* as classification with a certain property. To integrate them, we regard a path expression as an array of tags and identify path expressions which are similar to each other in terms of appearance order or appearance frequency of tags. As a result of the integration, we try to decrease the classes which do not contain enough XML elements to calculate accurate global weights. In other words, we propose a method to calculate the relaxed global weights by integrating similar path expressions.

This paper is organized as follows. In Section 2, we explain about information retrieval technique for XML element search and related studies. In Section 3, we propose a new method to calculate relaxed global weights. Experiments results are shown in Section 4, followed by concluding remarks and future work in Section 5.

2 XML Element Search Techniques and Related Studies

In this section, we describe existing XML element search techniques and their ways of calculating global weights. In addition, we also discuss on a related study which relaxes the query restriction related to document structures of XML elements.

2.1 An Expansion from Document Search to Element Search

Information retrieval techniques of XML element search often make use of the ones used in document search. In general, term weighting schemes of document

search utilize some kinds of the statistics, for example, local weights of each term, global weights of each term, normalization by document length, and statistics given by entire document collection [8]. Term weighting schemes of element search also utilize these statistics; however, we should treat global weights carefully. In the term weighting schemes of document search, we use all documents when we calculate global weights. This is because document is a unit for calculating global weights, that is, all documents are in a class. In element search, on the other hand, each element belongs to one of the classes; therefore, most term weighting schemes of element search are not based on elements but on the classes to which the elements with the same and/or similar path expressions belong. Now, the question is how we identify the "same class." In the next subsection, we refer to some of the most popular XML element search techniques and their standard classification of the same class.

2.2 Existing Methods of Calculating Global Weights

TF-IPF [3], a popular term weighting scheme of XML element search, is a path-based term weighting scheme that extends the well-known TF-IDF [12] approach for document search with the vector space model. In TF-IDF, local weights are calculated by the term frequency (TF) of each term in each document and global weights use inverse document frequency (IDF) of each term in all documents. Finally, a weight of TF-IDF is derived by the product of TF and IDF. In TF-IPF, global weights use inverse path frequency (IPF) which is the inverse of element frequency with the same path expression, while local weights are calculated in the same manner as TF-IDF. A weight of TF-IPF is, then, calculated by the product of TF and IPF. Furthermore, there are some studies which refine TF-IPF into more accurate one. Normalized TF-IPF [6] is one of them and it normalizes each element in its length. These term weighting schemes treat that the XML elements with the same path expression belong to the same class. In other words, if there are the elements which have different path expressions, these elements are classified to different classes. Hence, if the path expressions of elements become more complex, the number of classes increases. In this case, appropriate global weights cannot be calculated, because each class does not have enough XML elements. Therefore, we should consider how to decrease the classes which do not contain enough XML elements to calculate global weights.

Other popular approaches for XML element search are BM25E [7], which is based on Okapi's BM25 [11] term weighting scheme of document search with probabilistic model, and BM25F [10] for field search. Information retrieval techniques for field search are based on a document as a search unit, and used for structured document. BM25F gives a weight to each tag to consider the field that it appears by adjusting important degrees of each tag[2].

Meanwhile, BM25E is an information retrieval technique of element search like TF-IPF. There are two ways to calculate global weights where 1) all elements belong to the same class (inverse element frequency, IEF), and 2) the elements

[2] For example, the weight of `title` tag is given high value, because the element which includes `title` tag tends to be relevant if query keywords appear in such tags.

with the same tag belong to the same class (inverse tag frequency, ITF) [9]. We overview the problems of these approaches. IEF is calculated by counting text nodes repeatedly because there are overlaps among elements in the ancestor-descendant relationship. This means that documents containing numerous tags become influential too much. Even if the feature of each tag is considered in ITF, it does not consider the document structures of XML elements. Since both keywords and structures are used as a query in XML search, it is not appropriate to ignore document structures. Therefore, IEF and ITF are not suitable for global weights.

2.3 Relaxing Restriction of Document Structure in a Query

Keeping above points in mind, we should integrate path expressions if they are supposed to be in the same class. There is an existing study which relaxes query restrictions in terms of document structures [4]. This approach first calculates the global weights in advance based on the path-based classification, i.e., IPF, and then, calculates the term weights of query keywords based on the elements that satisfy query restrictions when the query is posted. However, we cannot identify which class the path expressions included in a query are categorized to before the query is given. Since the class classification is determined by a query, it takes longer search time because the term weights are calculated after the query is posted.

3 Classification of Similar Path Expressions

Concerning the problems of the existing methods of calculating global weights and the related studies of relaxing query restriction introduced in Section 2, we should develop new strategies to decrease the classes each of which contains enough XML elements for calculating global weights. It is better if the classes are classified before a query is processed. To satisfy these requirements, we integrate the path expressions which are similar in terms of their document structures in XML elements. Therefore, we propose a method to calculate relaxed global weights by integrating path expressions.

We define a path expression as an array of tags whose appearance order and appearance frequency of tags in the location steps are considered. Hence, we can relax the restriction of document structures in the order and frequency by our proposed classification, while existing approaches are based on path-based ones. Here, we propose the following three ways for calculating global weights by integrating path expressions; 1) relaxing appearance order of tags, 2) relaxing appearance frequency of tags, and 3) both relaxing appearance order and frequency.

3.1 Inverse Combination Frequency

Concrete examples of path integration with path expressions are shown in Fig. 1. First, we explain inverse combination frequency (ICF) which relaxes the appearance order of tags in path expressions. Tags in structured documents are

1: /article/sec
2: /article/sec/sec
3: /article/sec/person/sec
4: /article/sec/p/
5: /article/person/sec
6: /article/sec/sec/person
7: /article/person/sec/sec
8: /article/sec/sec/p

Fig. 1. Examples of path expressions

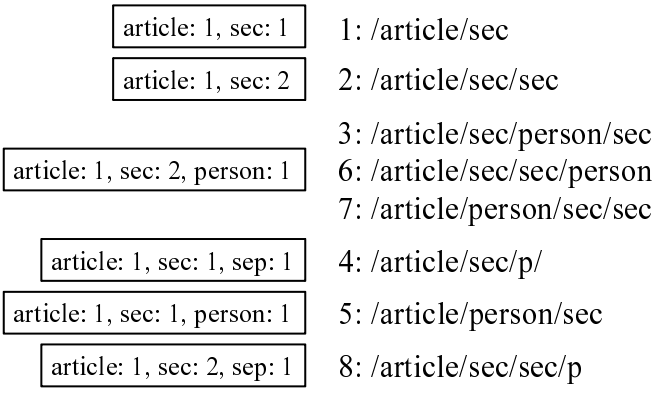

Fig. 2. An example of classification in ICF

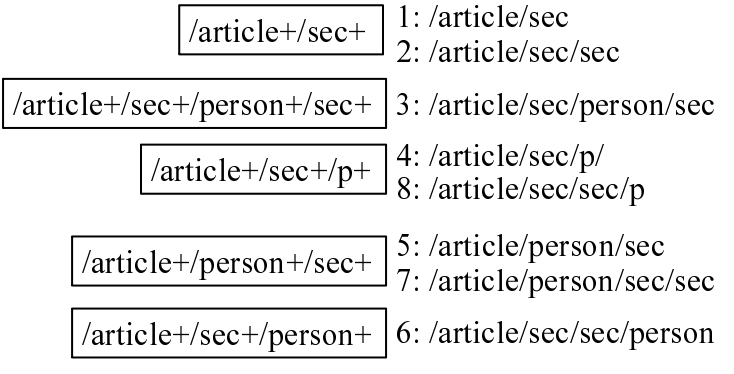

Fig. 3. An example of classification in IAF

categorized into two types of tags. One represents structural classification, and the other indicates something ideas, attributes, specific contents, etc. These two types of tags are supposed to be independent in their appearance. This suggests that a combination of tags can consist of two or more path expressions. However, it is not appropriate that these path expressions are classified to different classes. In ICF, we, therefore, do not consider the order of tags strictly but the combination of tags, i.e., names and frequency of tags.

To illustrate a concrete example of classification by ICF, a classification result of path expressions in Fig. 1 by ICF is shown in Fig. 2. We preliminarily enumerate the names and frequency of tags in each path expression to integrate path expressions classified as the same class. As a consequence, we integrate path expressions 3, 6, and 7 because all of them have one `article` tag, two `sec` tags, and one `person` tag. The global weights are calculated by all XML elements in path expression 3, 6, and 7 because they are in the same class.

3.2 Inverse Aggregated-Path Frequency

We explain inverse aggregated-path frequency (IAF) which relaxes appearance frequency of tags in a path expression. In some path expressions, a tag appears consecutively twice or more times, for example, `col` tags in `table` tag of HTML. In this case, even if the frequencies of the same tag appearing consecutively are different, we suppose that the features of a path expression are not so different because the semantics of each tag is fixed. Therefore, if consecutive tags are the same, such tags can be aggregated into one.

Consequently, we do not strictly consider the frequency of tags but their order, i.e., names and order of tags in IAF. In the same manner as ICF explained in Section 3.1, a classification result of path expressions in Fig. 1 by IAF is shown in Fig. 3. When the same tag consecutively appears twice or more times, such a tag is aggregated preliminarily. For example, since `sec` tags appear in path expression 2, 6, 7, and 8, path expressions 1 and 2, 4 and 8, 5 and 7 are integrated. This is because path expressions 1 and 2 have one or more `article` tags followed by one or more `sec` tags while expressions 4 and 8 have one or more `article` tags followed by one or more `sec` tags, and one or more `p` tags. Path expressions 5 and 7 also have one or more `article` tags followed by one or more `person` tags, and one or more `sec` tags.

3.3 Inverse Set Frequency

We described the way of integrating path expressions in terms of either appearance order or appearance frequency of tags in Section 3.1 and 3.2. In contrast, inverse set frequency (ISF) relaxes both appearance order and frequency of tags in path expressions. Accordingly, we do not consider the order and frequency of tags but the names of tag in ISF. Therefore, we classify the path expressions which are composed of the same tag name as the same class.

A classification result of path expressions in Fig. 1 by ISF is shown in Fig. 4. Path expressions 1 and 2 are in the same class because they are both composed of `article` and `sec` tags. Path expressions 3, 5, 6, and 7 are composed of `article`,

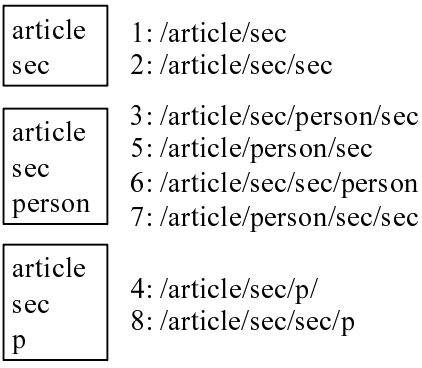

Fig. 4. An example of classification in ISF

Table 1. Effects of integrating path expressions in the INEX 2008 test collection

method	number of classes	the ratio compared to IPF
ICF	56,369	.85
IAF	32,421	.49
ISF	16,007	.24
IPF	66,210	1.0
ITF	495	.0075

Table 2. Effects of integrating path expressions in the INEX 2010 test collection

method	number of classes	the ratio compared to IPF
ICF	22,743,778	.97
IAF	23,383,388	.99
ISF	19,587,224	.83
IPF	23,502,448	1.0
ITF	22,110	.000094

`sec`, and `person` tags, while path expressions 4 and 8 are composed of `article`, `sec`, and `p` tags.

4 Experiments

In this section, we show some experimental results with the INEX 2008 test collection [5] and the INEX 2010 test collection [1], so as to confirm the usefulness of our proposed methods. In more detail, we conducted three kinds of experiments; 1) assessing the effectiveness of integrating path expressions, 2) analyzing the number of XML elements in each class, and 3) evaluating search accuracy.

4.1 Experiments for Integrating Path Expressions

Table 1 and 2 explain that how many path expressions are integrated. Each table shows the number of classes of each method and their ratios compared to IPF.

First, we discuss on Table 1. The result shows that the proposed methods could appropriately integrate the path expressions in the INEX 2008 test collection. IAF which relaxes appearance frequency cloud integrate more effectively than ICF which relaxes appearance order. The number of classes of ISF decreased to a quarter compared to IPF.

Table 3. The number of XML elements in a class and the ratio of the classes in the INEX 2008 test collection

numbers of elements	IPF	ITF	ICF	IAF	ISF
1	.57	.58	.53	.42	.37
2	.13	.14	.14	.16	.13
3	.062	.055	.067	.080	.063
4	.041	.026	.042	.057	.052
5	.027	.018	.028	.031	.036
6	.021	.0080	.022	.027	.032
7	.014	.010	.017	.017	.019
8	.014	.0040	.014	.019	.017
9	.011	.010	.012	.012	.013
10	.090	.0060	.010	.011	.010
11 or more	.10	.14	.12	.17	.25

In contrast, the proposed methods did not integrate path expressions effectively in the INEX 2010 test collection as shown in Table 2. Though both ICF and IAF did not work well, it seemed to be effective for ISF comparatively. This indicates that our methods did not seem to be sufficient. In particular, we need to consider that whether IAF which hardly integrates path expressions works well or not.

The reason of the result obtained was due to growth in the number of kinds of tags included in the INEX 2010 test collection. This causes an increase in the number of the combination of path expressions. As a result, we could not integrate path expressions effectively. Therefore, we should consider another condition for relaxing path expressions. For example, we need to screen valid tags, while we did not consider in the proposed methods. Because we supposed that the classes of path expressions are mainly based on the *structural tags*, it might be reasonable that we do not use *content tags* but the structural tags only. Otherwise, it might also be reasonable to integrate the tags with the same semantics. We should further consider a new relaxation method because the proposed relaxing approaches are not well sufficient.

We conducted some more experiments in the next section to investigate the results.

4.2 Analyzing the Number of XML Elements in Classes

The experimental result in Section 4.1 indicated that the usefulness of the proposed methods depends on the test collection. We, then, analyze whether the proposed methods can reduce the number of classes which do not contain enough XML elements to calculate global weights. Table 3 and 4 represent the number of XML elements in each class and the ratios of the number of the classes to that of all classes.

Table 3 explains that all of the proposed method reduce the ratio of the classes which contain only an element compared to IPF in the INEX 2008 test

Table 4. The numbers of XML elements in a class and the ratio of classes in the INEX 2010 test collection

numbers of elements	IPF	ITF	ICF	IAF	ISF
1	.65	.61	.66	.65	.65
2	.13	.14	.11	.13	.12
3	.047	.053	.050	.048	.052
4	.028	.031	.029	.028	.031
5	.017	.018	.018	.017	.020
6	.013	.016	.015	.013	.017
7	.015	.012	.013	.015	.0084
8	.011	.012	.0088	.011	.0070
9	.0061	.0052	.0059	.006	.0033
10	.0057	.0078	.0059	.006	.010
11 or more	.081	.10	.081	.081	.086

collection. In addition, the ratio of the classes which contain eleven or more XML elements increases in all our methods. For these reasons, we conclude that we could reduce the ratio of the classes which contain few elements by relaxing restrictions of appearance order and appearance frequency of tags in path expressions. Therefore, it is natural that ISF which relaxes both order and frequency is the most effective as our goal. Although it is arguable how many XML elements are enough to appropriately calculate global weights, we verified some effects on reducing the ratio of the classes containing few XML elements when path expressions are integrated.

Moreover, it is notable that a lot of classes in ITF have only a few elements. Though the INEX 2008 test collection has 495 kinds of tags, 298 kinds of tags (58%) appear only once. In other words, only a few tags are used repeatedly in all documents. It suggests that we should focus on the low-frequent tags to improve our approach.

As expected, we could not observe difference between our methods and IPF when using the INEX 2010 test collection, as long as we see Table 4. We should take another ways to achieve our goal, as mentioned in Section 4.1.

4.3 Evaluation on Search Accuracies

We examined search accuracy of three term weighting schemes by using the global weights obtained by our proposed methods.

More precisely, we used BM25E[3] by varying global weights. In addition, we also examined search accuracy of four more methods, three existing methods and a method without global weight, to compare to our proposed methods. Note that we evaluated only with the INEX 2008 test collection[4] because the proposed methods did not work well with the INEX 2010 test collection.

[3] The weight of term j in an XML element i is calculated as follows: $\frac{(k_1+1)tf_{i,j}}{k_1((1-b)+b\frac{el}{avel})+tf_{i,j}} \log \frac{N-df_i+0.5}{df_i+0.5}$ [7]D.

[4] We used 68 queries in the experiment.

Table 5. Search accuracy of our proposed methods

method	iP[.01]	MAiP
ICF	.6169	.1713
IAF	.6178	.1724
ISF	.6166	.1716
IPF	.6146	.1723
IEF	.6107	.1321
ITF	.6135	.1719
no global weights	.2364	.04689

We show search accuracy of these methods in Table 5. All of the proposed method slightly improved search accuracy. In other words, they do not affect the accuracy. Since the search accuracies of IEF and ITF are comparatively lower, we needs to consider the document structures of XML elements. Furthermore, we should give proper global weights to each term, because the method without global weights decreased its search accuracy significantly.

Both ICF and IAF could improve search accuracy. However, ISF reduced its search accuracy. It suggests that the deterioration of search accuracy might be caused by excessive relaxation.

We summarize the experiments in Section 4.1 through 4.3. Our methods do not work well with the INEX 2010 test collection but are effective with the INEX 2008 test collection. In the INEX 2008 test collection, ISF is the most effective approach in terms of the integration of path expressions, while IAF is the most effective one in terms of the search accuracy.

5 Conclusion

In this paper, we proposed methods to calculate relaxed global weights for XML element search.

In these methods, we integrate path expressions which are similar in terms of the order and frequency of tags to reduce the number of classes containing only a few elements. Our methods could reduce the ratio of such classes with the INEX 2008 test collection, whereas they do not work well with the INEX 2010 test collection. The experimental evaluations with the INEX 2008 test collection indicated that we could attain more accurate search. However, the results also showed that it might cause deterioration of search accuracy by excessive relaxation.

As future works, we should consider how we treat low-frequent tags, and more precisely explore to reveal why the proposed methods did not work well with the INEX 2010 test collection.

Acknowledgements. This work was supported in part by MEXT (Grant-in-Aid for Scientific Research on Priority Areas #21013035, #22240005, and #22700248).

References

1. Arvola, P., Geva, S., Kamps, J., Schenkel, R., Trotman, A., Vainio, J.: Overview of the INEX 2010 Ad Hoc Track. In: INEX 2010 Workshop Pre-proceedings, pp. 11–40 (December 2010)
2. Blanken, H., Grabs, T., Schek, H.-J., Schenkel, R., Weikum, G.: Intelligent Search on XML Data: Applications, Languages, Models, Implementations, and Benchmarks. LNCS, vol. 2818. Springer, Heidelberg (2003)
3. Grabs, T., Schek, H.-J.: PowerDB-XML: A Platform for Data–Centric and Document–Centric XML Processing. In: Bellahsène, Z., Chaudhri, A.B., Rahm, E., Rys, M., Unland, R. (eds.) XSym 2003. LNCS, vol. 2824, pp. 100–117. Springer, Heidelberg (2003)
4. Hatano, K., Amer-yahia, S., Srivastava, D.: Document-Scoring, for XML Information Retrieval using Structural Condition of XML Queries. In: IEICE technical report, pp. 13–18, DE2007-117 (October 2007)
5. Kamps, J., Geva, S., Trotman, A., Woodley, A., Koolen, M.: Overview of the INEX 2008 Ad Hoc Track. In: Geva, S., Kamps, J., Trotman, A. (eds.) INEX 2008. LNCS, vol. 5631, pp. 1–28. Springer, Heidelberg (2009)
6. Liu, F., Yu, C., Meng, W., Chowdhury, A.: Effective Keyword search in Relational Databases. In: Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data, pp. 563–574. ACM, New York (2006)
7. Lu, W., Robertson, S., MacFarlane, A.: Field-Weighted XML Retrieval Based on BM25. In: Fuhr, N., Lalmas, M., Malik, S., Kazai, G. (eds.) INEX 2005. LNCS, vol. 3977, pp. 161–171. Springer, Heidelberg (2006)
8. Manning, C.D., Raghavan, P., Schutze, H.: Introduction to Information Retrieval, pp. 157–159. Cambridge University Press, Cambridge (2008)
9. Piwowarski, B., Gallinari, P.: A Bayesian Framework for XML Information Retrieval: Searching and Learning with the INEX Collection. Journal of Information Retrieval 8(4), 655–681 (2005)
10. Robertson, S., Zaragoza, H., Taylor, M.: Simple BM25 Extension to Multiple Weighted Fields. In: Proceedings of the 13 ACM International Conference on Information and Knowledge Management, pp. 42–49 (November 2004)
11. Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M., Gatford, M.: Okapi at TREC-3. In: The Third Text Retrieval Conference (TREC-3), pp. 109–126 (1995)
12. Salton, G., Buckley, C.: Term-Weighting Approaches in Automatic Text Retrieval. Journal of Information Processing and Management 24(5), 513–523 (1988)