New Algorithm for Improving the Accuracy of Search Engine Depending on Semi-Structure

Yazed Alsaawy

Faculty of Computer and Information Systems Islamic University of Madinah. Al-Madinah, Saudi Arabia

Summary
Web Search engines are the most important and effective tools for Internet users, and the issues of search algorithms are still interest to many researchers. One of the most important sections for the development of current search engines is to work on parts of document instead of the whole document. XML Information Retrieval XML-IR is one of the techniques used on semi-structured XML files to improve search precision, due to ease of work with large documents, and also enable a new style of advanced search within a specific part of the document.
This research provides an Integrated based indexing approach to improve the indexing system in search engines to avoid the disadvantages of previous methods and to achieve additional performance and accuracy features. In addition, it provides an advanced version of the traditional VSM (Vector Space Model) retrieval algorithm and finally a new way to search within a specific website.
The stage of achieving and testing the proposed system confirmed its superiority in terms of accuracy of results compared to other methods of indexing and search.

Key words:
Information Retrieval, XML, Cos Similarity, XPATH, VSM

1. Introduction

The huge and rapid development of computer technologies and sciences and the exponential growth of information on the Internet have flooded us with a huge amount of information in various forms. To manage this information and extract useful information from it, it was necessary to have special systems for indexing and retrieval [1][2]. Following the widespread use and use of XML files to represent, characterize, exchange, store and transmit data on the Internet, researchers in IR systems have tended to take advantage of the structured environment of these files in order to increase the accuracy of research and to reduce the surplus and cognitive knowledge of the user [3][4]. This has improved the quality of the search results as the most important and most consistent part of the user's request is returned from the document, not all of the document as in conventional retrieval systems. These systems are called XML-IR.
These systems differ from each other in the way of indexing, and from the retrieval algorithm. All of which seek to increase accuracy in results and significantly improve performance. [5]

Manuscript received May 5, 2019
Manuscript revised May 20, 2019

In this work, we propose a new indexing method that would achieve a greater number of features implemented by previous methods of this topic. Some of them will be discussed in the next sections.
Furthermore, a new law and algorithm was also proposed for the XMLIR and previous indexing structures. The idea of this research is different from what researchers did in this field, since the search process is based on content searching or search on the structure and the content together.

2. Background

XML-based retrieval systems differ from traditional ones. Here is a comparison of the most important features and differences between them:

The basic feature of XML-based systems.

The search here will be more precise because each part of the document will have a special weight and when returned, the most importantly part will be returned only to the user in weight and thus improve performance as well [6].

Querying styles where there are two types of queries:

CO Content Only: Search content here there are no user query restrictions (keywords)
CAS Content and Structure: Here is a query about the content and structure of the document. There is respect for the structure and type of elements within the document (keywords + contract names or paths).
RESULT: Here returns a portion of the document, i.e. it returns the most specific part of the document, not each document. [7]

Challenges Facing IR XML Systems

Although the structure of XML documents can be used as a backbone for more focused and customized responses that contain the user-generated information and the least amount of non-related information, there are some key challenges associated with achieving the goal of returning elements of XML documents that the user is keen to have [8].

Word statistics (Term) within items.

In XML retrieval, the problem appears when you need to calculate these statistics within an item and not within a document, and note that the elements in the XML documents are nested.

Structure statistics (the degree of importance of the element itself).

In an xml document set, they are enormous number of XML elements that are not of the same all equal importance. How can we find the type of elements closest to the query [9].

Structure constraints when querying

Determining the required information including content with structural limitations is not easy for the user with a semi-structure and with a great variety of element names. This is a very true assumption with large and variable quantities of elements in XML. There is an important method that is common by constructing a synonym dictionary that is specific to the elements of the structure manually [10].

The interface used here discusses two cases:

Result Presentation: We must decide whether to display the item out of context or within the context of the document to which it returns. So you have to decide how the context will be presented. Only a few have addressed this issue.
Query Formulation : an XML structure that allows for a more accurate but more complex model. This is a burden for the end-user about knowledge of the structure of the query and its language XPATH, XQUERY because it is actually addressed to a specialized user rather than a regular user [11].
To help a user formulate his requests in the query language, usually it depends on XPATH ready sets with certain keywords. Or display notes on a logical, semantic or expression representation in the normal language and then process it automatically [12].

3. Literature Review

The most important characteristic of XML-based retrieval systems is the indexing algorithm used and the method of calculating the weight of words as well as the retrieval algorithm [13]. The following are the most famous algorithms and methods presented in each of them:

Current INDEXING styles

In traditional systems, there is a single indexing method, the Flat File Index with INVERTED INDEX, which considers the entire file as one text without interest in the structure [14].
With structured files, there are also many patterns and the most important ones are;

Tag (Field) based Indexing,

One of the simplest methods of indexing is to connect each term to the Tag node that falls within it and thus work on a single level of the structure and the loss of the tree structure completely and therefore the structure of the XML file is not really exploited. This research limited the search by the name of the node and then the keyword example of the E-Search engine [15].

Segment Based Indexing,

Here the document is displayed as a set of nested sections. However, there is no model that fully supported this idea, especially the issue of overlap and complexity within the document in addition to the issue of what is the most appropriate section to be returned to the user and on what basis will this section be divided [16].

Path based indexing,

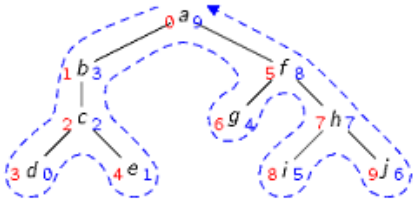
This method is useful for retrieving documents or retrieving parts of documents that contain a known sequence of item values or attributes.
Each index of the word Term is bound with its context in XML or a path from the islands to the deepest node containing the word Term. For efficient purposes, the paths can be replaced with unique OIDs to answer these queries. This approach is based on structural research techniques such as XPATH but the problem is that there may be two different nodes with the same exact path [17].

Tree based Indexing,

This methodology is used to solve the problem of the previous method by creating a unique object id associated with each node or element in the XML tree and this allows for the precise placement of the term within the index.
The GPX system uses this technique and it relies on XPATH to fully search the XML structure [18].

Labeling Schemes,

It is suggested that all nodes in the tree be renumbered by a tree algorithm in either depth or level, so each node will have a unique number, and words within a given node can be associated with this unique number. The problem here is that when you edit the document, the tree will be reconstructed, the item number will be renumbered, and the index will be updated [19].



Sequence Based Indexing,

It encodes the nodes in a series of pairs, in each pair the code is placed in its symbols, in addition to a series of symbols representing the connection to the parent roots. It uses a depth-first algorithm to move within the XML structure
In the previous example, for example, C node can be encoded in the following form (c, ba), but this method may cause duplication and sometimes interference [20].

Position-based Indexing

Here the indexing is by the location (spatial indexing). It depends on the physical or document format. The document is considered a binary tree or multiple tree and is quite similar to the spatial indexing of text files, but the ordered and logical structure of the XML file will be useful for this type of index. But the problem here is that if you modify the document, you must modify the index too [20].

The Retrieval algorithms used are RETRIEVAL MODEL

There are patterns and algorithms for retrieval in traditional systems such as [21]:

Boolean

Is a simple model based on the theory of groups. The word (Term) for the document either exists or not. It is characterized by a very accurate indicator, accurate and clear context and we express here the document in Polish [22]

Negative points:

- There is no partial match and there is no order of results.
- The user may not be able to write a Boolean or complex logical expression and will write only an easy expression and therefore the results would either be very large or few.

Extend Boolean,

This model, is as a development on the previous model, dealt with the issue of partial matching and would take the weights of the terms into consideration and its basic idea is that it creates a form or a special formula for each logical process (AND OR NOT) simulating the principle of the distance between two points and then calculating its value.

Consequently, since there is a partial match, there is a ranking of the documents according to their importance [8]. But the calculations is within it if the complexity of the query is somewhat complex, and the distribution of the processes change the order of the results and this is illogical [23].

$$(X1 \vee X2) \wedge X3 \Leftrightarrow (X1 \wedge X2) \vee X3$$

But Here First Result ≠ Second Result

VSM

Here I have a scale for each word (Term) within the document and within the query, these are the most important features, a simple model and has a solid mathematical formula dealing with local frequency and general frequency and can be programmed in an effective manner. It also achieves partial congruence and gives orderly results and has proved to be practical that its results are good [24].
However, this model contains some shortcomings, it does not care about semantic information or the meanings of words, nor is interest in the reliability of words on each other in some words. This is what the LSI Latent Semantic Index (LSI) approach pointed out, which is based on concepts rather than individual thermodynamics,

Probabilistic Model.

This model uses the probability theory for non-deterministic modeling in recovery processing [25].
Assumptions were made in an indistinct way. The word weight was calculated without word repetition information in the document or the length of the document. This is a weak point .

Methods of weight calculation

Some relied on a predefined definition of specific types of elements during indexing independently [26-27].
Others relied on the use of calculations within elements that are considered only papers
Or divide the elements into groups and for each set a specific score will be equal between all elements.
Some suggested that only high-weight items such as abstraction ...
Some considered that the length of the element (the number of words) is taken into account.
Finally, one way of calculating the distance or degree of similarity between the query and the document can be used in one of the famous ways:
1. Euclid's distance: It turned out not to be a good choice because he would prefer the large document from the small document, which is not always true.
2 - Manhattan continued: Uses absolute value, but also has the same previous negative

3-Continued Inner Product: Here is the output of the weighting but also has the same previous negative
4- Continue with Cosine Similarity: It is the best and most used app among previous dependencies.

$$Cos(Q, Dj) = \frac{\sum_{1}^{number\_of\_Terms\_in\_Q} W(TiQ) * W(TiDj)}{length(Q) * Length\ (Dj)}$$

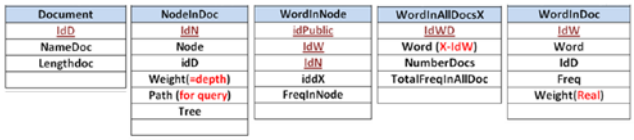
4. Proposed Method

Proposed Database Index

The basic idea in the proposed index is to create a hybrid index containing the basic keys of previous indexing methods in order to access a comprehensive indexing method that supports any traditional query or structure query easily, achieves the features of all previous indexing algorithms, and avoids their disadvantages. Therefore, we have integrated the following indexing algorithms:
(Tree Based indexing + Path Based indexing + Tag Based indexing).

Proposed Database and Indexing algorithm

As shown in the form where we note the new tables on the structure of search engines in traditional retrieval systems are the table of words within the node and node table within a file.
The Node column represents the node name and is important to the simple search process as in the Tag based index.
The Path column represents the path of the node (/ bookstore / book / title /) and the importance of the structure + content search.
Finally, the Tree column represents the path of the single node such as (/ bookstore [1] / book [1] / title [2] /) and its importance in removing the problem of repetition and overlap and returning a single result with a distinct XPATH path to the node



An automated contract discovery code was built into the XML file and stored according to the proposed structure and previous database using XMLTEXTREADER object predefined by .NET [2]

```
Function Create_Index (Depth, List of Nodes, Path)
 Next =1;
 Prev_Node = "";
 Path1="";
 Foreach (XmlNode node in Nodes)
   Path1 = Path + "/" + node.Name;
   If (node.NodeType == XmlNodeType.Element)
       If (node.Name == Prev_Node)
          Next++;
          Prev_Node = node.Name;
       Else
          Prev_Node = node.Name;
          Next = 1;
       End if
   End If
 End Foreach
 Depth = Depth + 1;
 Re-Call   Create_Index   (Depth,   node.ChildNodes,
Path1+"[" + Next +"]");
End Function
```

Proposed Algorithm for Create Indexing

Advantages of the proposed index.

- Tree column Provide greater accuracy in the placement of any word within the index where we obtained a unique object id associated with each node or element in the XML tree and thus solve the problem of overlap and repetition in these documents.
- Secure the direct access of the data when the result is returned because the format of the reference will be exactly the same as the XPATH query needed to accurately access the desired section within the XML document.
- The structure search process will be more efficient with the Path column for advanced users, for example

// BOOK / TITLE: JAVA

The search process will be easier with the Node column for ordinary users, for example

AUTHOR: ADNAN

Disadvantages of the proposed index.

The increasing in size of the previous indexes is due to a hybrid of several indexes with each other. However, now, size is a simple and unimportant subject compared to performance and effectiveness.

B. Proposed retrieval algorithm.

The research provides an idea of the development of the VSM (Vector Space Model) algorithm adopted in famous retrieval systems to be suitable for XML-based systems. A

special parameter will be added to the node site that contains the term within the document so that this parameter is included in the law of calculating the weight and importance of the term, which depends on the order of the search results. In this way, the great complexity of the researchers who have considered each element of the document as a separate document is bypassed, which has also resulted in the integrity of the document integration unit, which has been exceeded in our proposed method.
Moreover, the proposed method ensures that the structure of the document is preserved and that the problem of increasing the time required for retrieval resulting from doubling the number of documents is greatly reduced in the existing methods, while achieving the most important advantage in XML-based retrieval systems is to pay attention to the weight of the word according to its position within the document.
VSM provides a good way of estimating the importance of the word within the document so the addition to it in the serious form representing the importance of the word by its location within an item in the document.

The law for searching content within the file using the modified retrieval algorithm

$$W(i,j) = TF(i.j) * IDF(i) \rightarrow Weight\big(T(i) in\ D(j)\big) = \frac{frequency of term(i)}{Max frequency of any term in doc(j)} * LOG\ \frac{Number of docs}{Number of docs that contain term(j)}$$

In the previous law the weight of Term I is calculated within document j
The first fraction of the law is (TFIJ):
Frequency of a word (Term) within a document divided on the largest frequency of the word (Term) in all documents as a type of Normalization.
The second fraction in the law is the IDFI:
The binary logarithm of the total number of documents divided on the number of documents that contain the specified term to determine the degree of recognition of the word within the documents.
As conclusion, a word that is frequently repeated within a document is important within it, but provided that this word is not repeated in all other documents then it will be a common word and not an important one. But the previous law concerned the weight of the word within the document as a whole and did not care to weight the word according to its location within the document, which has been processed in the developed law as it is clear that the depth of the element within any XML document determines the importance of this element and therefore the importance of words within it, hence the less depth of the element the more important. (So the depth was stored in DB). Thus, the proposed law added a special part of the importance of the word within the structure of the document itself, depending on the depth of the element within the tree of this document.

$$New_{Weight(T(i) in\ D(j))} = Old_{Weight(T(i) in\ D(j))} * W(i)_{in\ xml} * P1$$

$$W(i)\_in\_XML = \sum_{n=1}^{Number\_nodes\_in\_doc\_that\_contain\_term(i)} \frac{freqTerm(i) in\ Node(n)}{P2 + depth\ (n)}$$

Default Value for P1 and P2 is 1

The smaller the P1, the greater the importance of the site (1, 0.5, 0.25, ...)
The larger the P2, the less important the site (1, 2, 3, ...)

Specific Structure Search Law

In this research, a new proposal that differs from the complexity of the laws presented in the previous research is presented. We used linear search for the query within the path column stored in our database for each node because the maximum length of any node in any document is often between 6-7. This is confirmed by the INEX organization competent in the field of XMLIR in its test group, which means that the possibility of applying such research without a real impact on performance.
Finally, after calculating and storing the weights of each term in the database, we use the COS code to study the distance between any query and between each document that is indexed, just as in conventional retrieval systems and the document that achieves the largest distance is the optimal and most relevant. Do not forget that the query treats any document without affecting the total number of documents or the number of documents containing the terms in this query.

Evaluation of the proposed system.

One of the most important things to consider is how to evaluate the performance of our retrieval system in terms of the database used and its effectiveness or the time needed to query and obtain the result and then the effectiveness and quality of this result and perhaps the last one is of the most important systems to evaluate the recovery at all.
The most important criteria used in the evaluation of search engines regarding relevance results are (Precision & Recall), which are advanced metrics applied and calculated on a set of documents known as the test set to determine the effectiveness and importance of the result of the retrieval system [28].
Precision: Percentage of relevant documents returned on the number of all retrieved documents.
Recall: the relative ratio in the returned set for all documents related to the test group.
Thus, to evaluate effective retrieval systems, there is a test set and methods of calculating the total, i.e., the importance of each result. The test set consists of a set of documents, a

set of user requests, and a set of rulers to determine the relevance and quality of the result for queries.
However, it should be noted that the implicit assumption in the traditional test of the group is at the document level. This is not a valid rule in the case of XML, which can return a part or components of the document as a result of a query in order to evaluate the effectiveness of such systems Knowledge structure.

The evaluation applied in our research

First case: Due to the lack of a special corpus for these systems, 30 documents were applied (articles on a variety of topics). The results were achieved for the user in both systems (proposed by us and the tradition retrieval system). The returning results in each system are approximate with differences in order results. The benefit was that we obtained first evidence of the system's effectiveness in finding the required information.
We then tested again by adding five specially built documents, which would be identical in number, name, depth, and total number of words in each document, so the length of the document would be insignificant and the importance of the site would be quite clear.
In the proposed amended law, we observed how the site of the word is more important, or less, by controlling the values (P1, P2) of these two factors at the expense of their overall importance. Moreover, the importance of the document itself. The results were consistent with the port's expectations for these queries.
In order to test the results of operations, we searched for number of (30) queries. The user evaluated the effectiveness of the results of these queries and it was also very good.

The results of some queries are applied in both systems.

Finally, we provided some of the queries we have implemented on the (30) indexed documents, which are uploaded with the online system for trial and verification. The results are as follows:

Query	Retrieval Docs in XML IR	Retrieval Docs in CLASSIC IR
Vaccine	1(1)	1(1)
Vaccine disease	1,18,29	1,18,21,25,29,28,7
Vaccine disease child	1,21	1,18,21,25,29,28,22,7
Face book	2,13,17,14	2,13,17,14
Face book finance	2,13,17,14	2,13,17,14
Face book heart	0,2,1,13,17,19,14	0,2,1,13,17,14,19
Apple	9	4,9
Apple lion	4,9	4,9
Computer	8,5,4,7,23	8,5,4,7,23,3
Computer term	8,5,7,4,23,1,11,19	8,7,5,4,23,11,19,3
Super computer	5	5
Super school	20,21,10,7	20,5,21,13,25,4,17
Steve jobs	21,11,29,3,13,18,4,27,24	3,29,13,25,4,17
Tree	9,28,17	9,28,17,3
Tree term	9,7,1,28	9,7,28,17
Sport	11	11
Sport term	11,7,1,8,19	7,11,817
Christmas	9	9
Economic	14,12	12
Failure	6	6
Failure bank	12,6,22,2	12,6,22,2
Egypt face book	13,14,2,17	13,14,2,17
Egypt lemon	13,14,17,18	13,14,17,18
Sadness	26,0,22	26,0,15,22,3
Famous	22,15,25,3	22,15,25,3
Famous English	3,17,13,22,15,0	3,17,22,13,15,0
Famous seven	15,22,3	15,22,3
Employer	24,27	24,27
Poet	28,24	28,24
Poor	27,29,26,22	27,26,29,22

Interface of the application

5. Conclusion

This research presents a new idea for an integrated and hybrid index among several technologies to achieve a large number of features to be compatible with the indexing of structured files that have a tree structure such as XML files. The research presented a developed law on the famous VSM law for compatibility with modern retrieval systems that work on structured documents. The content search is done by taking into account the importance of the location of the word within the document. Finally, a new feature was added to the search in the structure and content together, where the experiments proved the effectiveness of the proposed system.

With regard to the future prospects of this research, we suggest working on:

- Integrate these systems with massive databases.
- Use Semantic Web techniques to reduce the size of indexes and increase the real understanding of user queries.
- Pay attention to comments and user opinions on documents or parts to improve the accuracy of results based on user experiences
- Use Doc Classification with Data Mining techniques to improve returned results.
- Design a new query language for XML IR systems with its own advanced query interface that makes it easier for the average user to work on complex structure queries.

Acknowledgment

The author would like to thank Dr. Ahmad Alkhodre and Dr. Adnan AbiSen for their valuable comments and suggestions to improve this article.

References

[1] Canessa, John C., Giancarlo Canessa, and Gino G. Canessa. "System and methods for metadata management in content addressable storage." U.S. Patent Application No. 15/331,775.

[2] Manning, C., Raghavan, P., & Schütze, H. (2010). Introduction to information retrieval. Natural Language Engineering, 16(1), 100-103.

[3] Keith, M., Schincariol, M., & Nardone, M. (2018). XML Mapping Files. In Pro JPA 2 in Java EE 8 (pp. 593-654). Apress, Berkeley, CA.

[4] Habi, A., Effantin, B., & Kheddouci, H. (2017, August). Search and Aggregation in XML Documents. In International Conference on Database and Expert Systems Applications (pp. 290-304). Springer, Cham.

[5] Büttcher, S., Clarke, C. L., & Cormack, G. V. (2016). Information retrieval: Implementing and evaluating search engines. Mit Press.

[6] Park, D. H., Liu, M., Zhai, C., & Wang, H. (2015, August). Leveraging user reviews to improve accuracy for mobile app retrieval. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 533-542). ACM.

[7] Guezouli, L., & Essafi, H. (2016). CAS-based information retrieval in semi-structured documents: CASISS model. Journal of Innovation in Digital Ecosystems, 3(2), 155-162.

[8] Schweiger R, Hölzer S, Dudeck J. Advanced Information Retrieval Using XML Standards. Studies in Health Technology and Informatics 2005; 116: 677-682

[9] Evangelos Kotsakis, Structured information retrieval in XML documents, Proceedings of the 2002 ACM symposium on Applied computing, March 11-14, 2002

[10] Vutukuru V, Pasupuleti K, Khare A and Garg A Conceptemy: An issue in XML information retrieval. In: Proceedings of theinternational world wide web conference (WWW), 2002.

[11] Ranked Information Retrieval on XML Data, Seminar "Information organization und -suche mit XML" Dr. Ralf Schenkel, avaliable on : http://resources.mpi-inf.mpg.de/d5/teaching/ss03/xml-seminar/talks/BlumNicolausUhl.ppt

[12] Norbert Fuhr , Kai Großjohann, XIRQL: a query language for information retrieval in XML documents, Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, p.172-180, September 2001, New Orleans, Louisiana, USA

[13] Raut, A. D., & Atique, M. (2014). A survey of indexing techniques for xml database. Compusoft, 3(1), 461.

[14] Mariano P. Consens and Ricardo Baeza-Yates. Database and information retrieval techniques for XML. In Proceedings of the 10th Asian Computing Science conference on Advances in computer science: data management on the web (ASIAN'05), Stéphane Grumbach, Liying Sui, and Victor Vianu (Eds.). Springer-Verlag, Berlin, Heidelberg, pp. 22-27, 2005

[15] Stockton, John K., and Ari K. Tuchman. "System and methods for units-based numeric information retrieval." U.S. Patent No. 8,756,229. 17 Jun. 2014.

[16] Arngren, Tommy, Joakim Soderberg, and Marika Stalnacke. "Apparatus and methods for indexing multimedia content." U.S. Patent No. 9,846,696. 19 Dec. 2017.

[17] Liu, Zhen Hua, et al. "Techniques of efficient XML query using combination of XML table index and path/value index." U.S. Patent No. 9,436,779. 6 Sep. 2016.

[18] Maskare, M. P. R. (2017). Answering XML Query Using Tree Based Association Rule.

[19] Alghamdi, N. S., Rahayu, W., & Pardede, E. (2014). Semantic-based Structural and Content indexing for the efficient retrieval of queries over large XML data repositories. Future Generation Computer Systems, 37, 212-231.

[20] Posonia, A. M., & Jyothi, V. L. (2014, December). XML document retrieval by developing an effective indexing technique. In 2014 Sixth International Conference on Advanced Computing (ICoAC)(pp. 120-123). IEEE.

[21] Büttcher, S., Clarke, C. L., & Cormack, G. V. (2016). Information retrieval: Implementing and evaluating search engines. Mit Press.

[22] Fkih, F., & Omri, M. N. (2016). IRAFCA: an O (n) information retrieval algorithm based on formal concept

analysis. Knowledge and Information Systems, 48(2), 465-491.
[23] Lv, F., Zhang, H., Lou, J. G., Wang, S., Zhang, D., & Zhao, J. (2015, November). Codehow: Effective code search based on api understanding and extended boolean model (e). In 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE) (pp. 260-270). IEEE.
[24] Pardede, J., & Husada, M. G. (2015). Comparison Of Vsm, Gvsm, And Lsi In Information Retrieval For Indonesian Text. Jurnal Teknologi, 78(5-6).
[25] Dahak, F., Boughanem, M., & Balla, A. (2017). A probabilistic model to exploit user expectations in XML information retrieval. Information Processing & Management, 53(1), 87-105.
[26] Pradhan, N., Gyanchandani, M., & Wadhvani, R. (2015). A Review on Text Similarity Technique used in IR and its Application. International Journal of Computer Applications, 120(9).
[27] Xu, R. (2014, July). POS weighted TF-IDF algorithm and its application for an MOOC search engine. In 2014 International Conference on Audio, Language and Image Processing (pp. 868-873). IEEE.
[28] Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PloS one, 10(3), e0118432.

Yazed B. ALSAAWY Yazed B. ALSAAWY is the Dean of Information Technology deanship at Islamic University (IU). Earlier, he has served as an Assistant Professor at Faculty of Computer and Information Systems, and a Vice Dean for academic affairs at FIT. He received his Ph.D. in computer science from De Montfort University, England in 2014. He works in the fields of software engineering, and has authored a number of research articles in software engineering, eLearning, security and privacy. He participates in many funded projects, including ones with King Abdul-Aziz City of Science and Technology (KACST).