The State-of-the-arts in Focused Search

Rongmei Li

Department of Computer Science,
University of Twente, P.O.Box 217 7500AE
Enschede, The Netherlands
lir@cs.utwente.nl

Abstract. The continuous influx of various text data on the Web requires search engines to improve their retrieval abilities for more specific information. The need for relevant results to a user's topic of interest has gone beyond search for domain or type specific documents to more focused result (e.g. document fragments or answers to a query). The introduction of XML provides a format standard for data representation, storage, and exchange. It helps focused search to be carried out at different granularities of a structured document with XML markups. This report aims at reviewing the state-of-the-arts in focused search, particularly techniques for topic-specific document retrieval, passage retrieval, XML retrieval, and entity ranking. It is concluded with highlight of open problems.

Key words: Focused Retrieval, Passage Retrieval, XML retrieval, Entity Ranking, Information Retrieval

1 Introduction

The exponential growth of the internet has accelerated the difficulties to access the available online information for Web users. Due to the nature of large heterogeneity and noise of Web data, search engines have to be improved for retrieving highly relevant and more focused results (e.g. document fragments or answers to a query). In most current scenarios, the retrieved results are still a ranked list of Web pages that are considered to be relevant to the user's topic of interest. To improve the relevance of the results, efforts have been made for a particular type of queries recently. For instance, the specialized search (e.g. Google Scholar[1], hierarchical Web directories (e.g. DMOZ[2], Wikipedia[3], Yahoo! Directory[4]), currently emerging semantic Web applications[5], and many other software agents and collaborative filtering systems. All these approaches try to focus on one information aspect of the entire document content. To make use of

[1] http://scholar.google.com
[2] http://www.dmoz.org
[3] http://wikipedia.org
[4] http://search.yahoo.com/dir
[5] http://swse.deri.org

these limited facilities for focused result, a user is first able to find them. In case of long documents (e.g. book, product catalog) or documents covering a wide variety of topics (e.g. Web blog), a user has to zoom in on the relevant parts inside the retrieved document. Besides answers that can be found in a relevant document, users may search for an answer that has to be derived from several relevant documents.

The recent research on focused search aims at reducing such cognitive load on the user by locating relevant from irrelevant content within a document [53]. Based on the different search goal, the general ad-hoc retrieval task can be refined to be more focused retrieval such as passage retrieval, XML element retrieval, entity ranking, and question answering. All in common are to search for a finer piece of information from documents as the answer to the user's information need.

While progress is made for effective access to Web information, the organization of information publication experiences a major change. The most evident is the introduction of the eXtensible Markup Language (XML) by W3C[6] as the text format standard for data representation, storage, and exchange. Different from the predominant Hypertext Mark-up Language (HTML), XML provides meaning about the stored content in addition to the structure of a document. More precisely, in the context of text documents, XML is used to specify the logical, or tree, structure of documents, in which separate document parts (e.g. chapter, section, abstract) and their logical structure (e.g. a chapter made of sections, a section and its title, an article and its abstract) are explicitly marked-up [27].

The document structure may help focused search to find the most relevant parts of a document more directly than the case where documents are flat and have no logical structure. The continuous growth in XML information repositories has been matched by increasing efforts in the development of XML retrieval systems, in large part aiming at supporting content-oriented XML retrieval [28]. The interest in XML search and retrieval became apparent first in ACM SIGIR[7] workshop on XML and information retrieval in 2000 and was further boosted by the INitiative for the Evaluation of XML Retrieval (INEX) in 2002 [16]. INEX established a framework for cross comparison among content-oriented XML retrieval approaches given same test collections and evaluation measures.

With the goal of removing the onus on the end-user who is searching for fine granularity of information, focused retrieval should not be considered as the replacement of document retrieval. In fact, users like to interact with documents. Interview results showed that users expect the retrieved components to be accompanied by the documents that contain them. They would feel rather uncertain if elements with no context information were retrieved [6]. In the view of research, document retrieval remains valuable as focused retrieval that combines two different aspects: 1) the retrieval of the relevant documents similar to traditional document retrieval, and 2) the retrieval of the relevant text within

[6] http://www.w3.org/XML/

[7] http://www.sigir.org/

these documents [24]. Focused retrieval techniques are appreciated, but need to be accompanied by other views of the entire document to give evidence of the appropriateness of the found information [42].

In this report we will survey the state-of-the-art focused search techniques for the main concerned retrieval tasks, particularly the ranking strategies. We start at the document retrieval designed for specific topic of the user's interest in section 2. In section 3, 4, and 5 we review techniques for focused retrieval namely passage retrieval, XML retrieval, and entity ranking. The report highlights open questions in section 6 and is concluded in section 7.

2 Topic-Specific Document Retrieval

In traditional document retrieval, an entire document is examined for its relevance to a given query based on a ranking model. A document is assumed as *the bag-of-words* [7] and its structure, if any, is ignored. The recent technique for effective document retrieval is to exploit topic coherence from a group of documents within the same topical category. Therefore, the retrieved documents are focused on the topic of interest (e.g. arts, business, health, entertainment) or the type of document (e.g. Web blogs, personal Web pages, FAQs, cultural heritage pages).

Roughly, there are two approaches for topic-specific document retrieval. One adopts the premise that similar documents will match the same information needs [41]. In this approach [31], [2] documents are classified into clusters and then ranked by cluster-based models. Relevant documents are retrieved from the cluster with highest rank. Another relatively new approach tries to solve the problem of query ambiguity by augmenting topical context information to a query. The topical context is represented by a topical model, particularly, a language model that can be computed from documents relevant to the given query [29], [30] or documents in a topical category [3], [57]. Both approaches compute the relevance likelihood of a document to a query from the term statistics (e.g. within document term frequency *tf* and inverse document frequency *idf*) that can be extracted from the document index. The structure of a document has not been fully used in this task. For the ad-hoc search, the query is also given as keywords since the relevance of the whole document content is the main concern but not that of the document structure.

The topic-specific document retrieval has shown its good performance on average in literature. However, it does not work effectively for relevant documents that have highly relevant but smaller portion of information when compared to the document length. In one case, this kind of documents may be missed because of irrelevance. In another case, a user has to search for the precise piece of information (entry point) in the retrieved document herself. However, in a unstructured document, it is hard to navigate a user directly to the most relevant parts of the document.

3 Passage Retrieval

The early attempt to focused retrieval is to locate relevant paragraphs in documents automatically. "*When the stored document texts are long, the retrieval of complete documents may not be in the user's best interest. In such circumstances, efficient and effective retrieval results may be obtained by using passage retrieval strategies designed to retrieve text excerpts of varying size in response to statements of user interest*" [43]. In addition to the advantage of fast information access for the end users, passage-level evidence can be used alone or in combination with the document level evidence to improve effectiveness on many information retrieval tasks (e.g. document retrieval [11], [56], XML retrieval [19], and question answering [13], [49]). In literature, documents can be decomposed into fixed-length passages, variable-length passages, discourse structures (e.g. sentences, paragraphs, and sections), or topic segments. The performance of passage retrieval strongly affected by the size of the passage, passage boundaries, and the degree of overlap between passages. Using fixed window passages instead of full documents is a more effective approach for relevance feedback. Passages based on paragraph boundaries are less effective than passages based upon overlapping text windows of varying size. It is held for both document retrieval base on a single best passage, and document retrieval based upon combining document-level and passage-level evidence [11].

Basically, methods used in passage retrieval can be categorized into density-based and relation-based framework respectively. The relation-based passage retrieval outperforms the density-based counterparts [13]. The density-based framework relies only on lexical level matching to rank passages. It considers each query term an independent token. However, this simplification does not hold in many cases because dependency relations exist between words. In this approach, statistic information instead of semantic information is used. The term frequency in query, a set of documents, or passages is used to capture the importance of terms. The term co-occurrence statistic is used to add additional connection to the original query terms. But it may bring noisy terms and miss relevant terms that are semantically related to query terms. In general, density-based measures of query terms are important in passage ranking [50].

In the relation-based framework, additional knowledge or linguistic cue is used. The dependency relation paths are first extracted from the query terms and candidate sentence answers and rank the candidates according to similarity between their relation paths with that of the query's. The technique works better for long queries (more than three terms [13]) as they contain more contextual terms. For short queries, relevant contextual terms and additional relations can be extracted, for instance, from Web snippets [49]. In this approach, the dependency relation between query terms can be matched either strictly or at varying degree. The degree of match of pertinent relations in candidate sentences with their corresponding relations in the query can be measured by statistical methods (e.g. mutual information and expectation maximization [13]). When a document is treated as a sequence of words, the word sequence can be model by a stochastic process for extracting a coherent relevant passage with variable length [21].

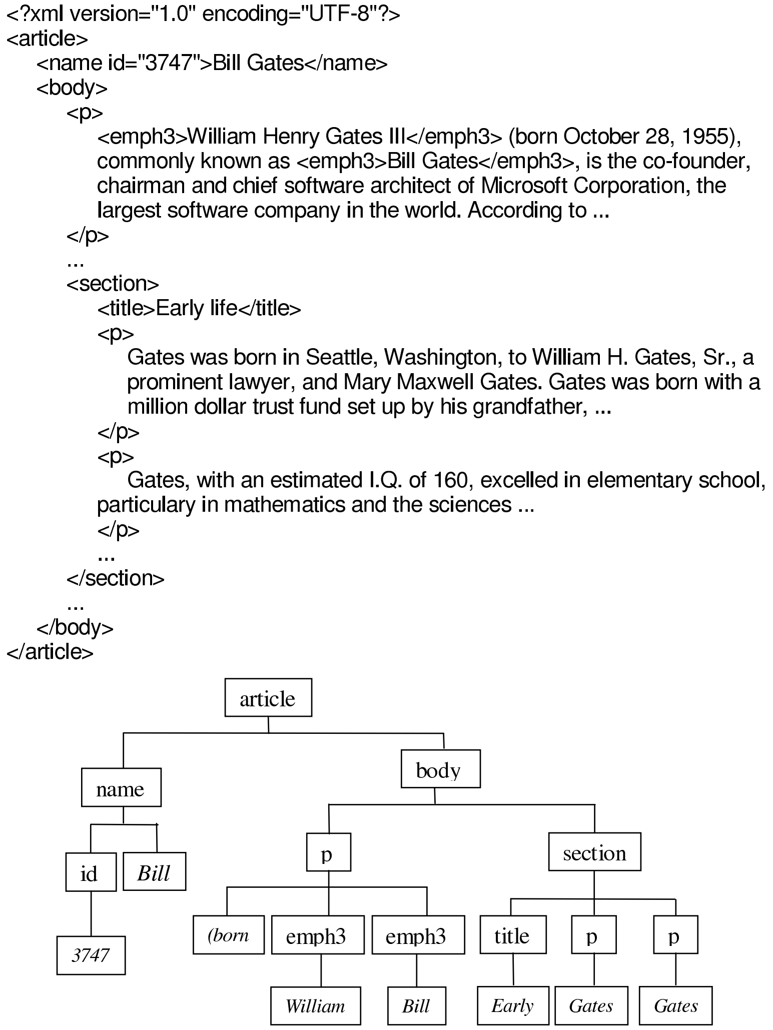
A passage retrieval system typically does not take into account the structure of a document. Compared to document retrieval, the index of a passage retrieval system also needs to maintain word positions inside documents, which typically doubles the size of the term posting lists [42].

4 XML Retrieval

XML (element) retrieval is the task to identify the elements in a document that are relevant to the user's information request. Usually, elements are of a lower granularity than passages and all elements can be described as passages. However only some passages can be described as elements [23]. In this case, techniques for passage retrieval can generate a comparable element retrieval ranking (e.g. fixed window passage retrieval [19]). In a similar study, a direct estimation of the relevance of elements is found superior than that based on passage-evidence [20]. The XML markups are the key features that facilitate focused search as they indicate the logical structure and the meaning of the document content. A example XML document looks like the above subfigure in Figure 1. Its logical structure can be represented as a hierarchical tree in the subfigure below where leaves are the text content of elements or attribute values and the root and branch nodes are element markups. An XML element is identified by its text content and its structure which is the path from the root element to itself.

4.1 XML Document Indexing

As the tradiational document retrieval, the text content of XML documents in a collection has to be indexed before a retrieval algorithm is applied. The classical indexing strategy uses terms statistics (e.g. within document term frequency *tf* and inverse document frequency *idf*). For XML documents, the term statistics have to be at lower level (the element level) for allowing the retrieval of elements at any level of granularity. The simplest approach is to use within-element term frequency *etf* and inverse element frequency *ief*. The approach has problem of over-computing *ief* for elements in such a nested structure. In the example Figure 1, the paragraph containing "Gates was born" will be counted once for the paragraph itself and more for its ancestors *section*, *body*, and *article* for estimating *ief*. In literature, the *ief* value can be calculated across elements of the same type [51] or across documents [12] or is derived from through the aggregation of term statistics of the element's own text and those of each of its children elements [37]. The elements of XML documents can be indexed selectively for 1) leaf elements only whose term statistics are then propagated to those candidate branch elements as their ranking score [15]; 2) elements which contain a text longer than a number of terms; 3) different types of elements in XML documents that appear often in previous relevant data [12].



```
<?xml version="1.0" encoding="UTF-8"?>
<article>
    <name id="3747">Bill Gates</name>
    <body>
        <p>
            <emph3>William Henry Gates III</emph3> (born October 28, 1955),
            commonly known as <emph3>Bill Gates</emph3>, is the co-founder,
            chairman and chief software architect of Microsoft Corporation, the
            largest software company in the world. According to ...
        </p>
        ...
        <section>
            <title>Early life</title>
            <p>
                Gates was born in Seattle, Washington, to William H. Gates, Sr., a
                prominent lawyer, and Mary Maxwell Gates. Gates was born with a
                million dollar trust fund set up by his grandfather, ...
            </p>
            <p>
                Gates, with an estimated I.Q. of 160, excelled in elementary school,
            particulary in mathematics and the sciences ...
            </p>
            ...
        </section>
        ...
    </body>
</article>
```

Fig. 1. A sample XML document in the INEX 2007 document collection (above) and its tree representation (below)

4.2 XML Document Querying

Similar to the document retrieval, XML retrieval is expected to return relevant elements to a given query. The XML query is no longer the keyword type query for ad-hoc retrieval but a language that is able to express various levels of content and structural constraints. The well-known query language is called NEXI that is a path-based language used by the INEX evaluation initiative [54]. An overview on other XML query languages can be found in [27].

4.3 Retrieval with Content Constraints

Once an appropriate index is built for a XML collection, the corresponding retrieval strategy can be applied. The main approaches are the variant ranking models for traditional document retrieval, such as vector space model, BM25, and language models. The most straight forward ranking method is to treat all indexed elements as candidate answers and rank them with or without structural constraints by their relevance to a given query. The relevance score can be estimated from indexed term statistics at the element level or the document level. Since elements in XML documents are not equally important, their prior probability of relevance provides important evidence for relevance estimate. In literature, various XML features are used to estimate prior probability such as the length of an element [25], the path length of an element [18], the type of an element (its tag), the location of an element in the original document [18], the number of topic shifts in an element [1]. For selective indices that are built for chosen elements, the ranking score of elements has to be derived. According to [27], there are at least three combination strategies to provide a rank list of all potentially retrievable elements. Some good examples are presented as follows:

Score Propagation. This strategy (e.g. [15], [37], [39]) is used to rank elements based on a leaf-node index and finer granularity of statistics (e.g. query term frequency in the leaf element times inverted collection frequency). First the relevance judgment score is computed for all leaf-elements that contain at least one query term. The scores are then propagated upward to their ancestors. The resulting relevance score of a branch element is a weighted sum of ranking scores of its children. The propagation weight can be defined as: 1) equal weight to each element so that the sum of score is the average score of all children elements; 2) weight that is equal to the length of the element divided by the length of its parent element; 3) weight that reflects the importance of specific element types or the degree of the dependence between the element and its parent, for instance, the number of relevant children that a branch has. For branch elements having only one relevant child element, the child element should be ranked higher. Otherwise, this branch element ranks higher.

Score Aggregation. This strategy is to rank elements based on the aggregated representation of its own content and other content (e.g. the text of children elements, parent elements, and the document [37] or the “attractive” parts of a XML document [18]). Each representation contributes differently to the final aggregated ranking score. The simplest aggregation is linear interpolation of all contributions (e.g. [37]). A mixture language model is a natural choice for this. However, the effectiveness of the aggregation depends heavily on the appropriate settings of the weighting factor of each component, whose values are usually estimated through learning methods [27]. The content of “attractive” parts are derived from a user study on which parts of document or an XML element are more likely to attract a reader’s attention. To boost the retrieval effectiveness of this strategy, non-content priors can be integrated with this ranking model such as the location of the element in the original document and the length of the element path [18]. It is worthy to mention that the score combination in the aggregation strategy for branch elements is different from that in the propagation strategy as the former is to used for direct element ranking and the later is for representation [27].

Score Merging. This strategy is adopted for the indexing strategy for different type of elements in 4.1. With selective indexing strategy, a separate index is created for each selected type of elements (e.g. article, abstract, section, paragraph, etc). To a given query, a ranking model has to run against each index seperately and retrieve separate ranked lists of elements (e.g. article elements, section elements, paragraph elements, etc). These lists are merged to provide a single rank across all element types. To merge the lists, normalization is performed to take into account the variation in size of the element in the different indices (e.g. paragraph index vs article index) so that scores across indices are comparable [34].

4.4 Retrieval with Content and Structural Constraints

In the section 4.3, we have summarized the state-of-the-art ranking strategies that only consider the content relevance of retrieved XML elements to a given keyword query. If the given query contains structure constraints such as a location path `chapter/section`, the retrieved elements must comply with both the content and structure conditions. The query having both constraints is called *content-and-structure* (CAS) query in INEX which is expressed by the NEXI query language. In literature the structural constraints in CAS queries can be fulfilled after the content constraint is satisfied or with the content constraint when the relevant XML fragments are retrieved. The specified structural constraints do not have to be strict in order to preserve content relevant elements or to validate path more efficiently. The examples of vague interpretation of the structural constraints are:

- allow equivalent tags (e.g. paragraph `<p>` and the first paragraph `<p1>` of its siblings in a document tree are in the same synonym group) [35];

- rank query path by similarity between the query tree and the document tree (e.g. element with the shorter document path is ranked higher and the element violating the query tree is ranked lower) [44];

- retrieve structurally similar elements that have the structural characteristics of the relevant elements for a give query [36];

- generate the overall ranking score of a document or sub-tree for a CAS query by combining its content and structural scores. The structural scoring model essentially counts the number of navigational (i.e., tag-only) conditions that are completely satisfied by a result candidate and assigns a small and constant score mass for every such condition that is matched [8].

- apply the path factor extension to a (ranked) list of elements and push those elements to the top of the list that (partially) matches the path elements specified in a query [60].

- use the request penalty factor to re-rank the final set of results of a query so that the relevance scores of the elements contain excessive elements in their path will be decreased [60].

While the evaluation of ad-hoc information retrieval system based on content constraints is well established (e.g. TREC[8] and INEX), the evaluation on stuctural satisfaction is still questionable. In other words, there was no assessment of whether, for instance, a section element was a better element type to return than another element type (if both were relevant according to their contents) [27]. On the other hand, structural hints in queries do not improve retrieval accuracy because users are particularly bad at giving structural hints [52].

4.5 Retrieval for Overlap-free Result

In the XML retrieval setting, it is difficult to locate the most exhaustive and specific elements in the document tree and return only these as the answer to a user. Because of the nested structure of XML documents, when an element has been estimated relevant to a given query, it is likely that its ancestor will also be estimated as relevant [27]. Consequently, the text fragments of that element will be returned once from its own element and once from its ancestors'. Such overlapping result is redundant and should be removed.

The most intuitive solution for removing overlapped XML elements is to identify the elements with highest rank in the result list and remove any ancestor and descendent elements from lower ranks. This post-filtering process is applied recursively. However, these solutions do not necessarily select most relevant overlap free results if it depends on the initial ranking result which is the combined result of independent ranking algorithms (e.g. [35]). The better approach is to make use of the tree structure of the XML documents for selecting the most relevant element from a list of overlapping results. Some effective examples are summarized as follows:

- use both relevance estimate and relative *usefulness* compared to other elements in the same path to obtain the over-lap free element. The *usefulness* of each element (node) is estimated by a *utility* function that are the product of the estimated relevance score, its length, and the amount irrelevant information contained in its children elements. An element with a utility value higher than the sum of the utility value of its children is selected over its children. Otherwise, the children elements whose utility values exceed a

[8] http://trec.nist.gov/

threshold are selected as the final answer [36].

- apply two-stage filtering on the XML document tree from bottom up [33]. At the first stage, clusters of highly ranked results in the tree are identified and the most relevant element are retained for each cluster based on its relevance score and the distribution of retrieved elements in the tree. The algorithm deals with three main cases: 1) select the node over its parent when it is substantially more relevant than its parent; 2) select the node over its parent when it is not substantially relevant than its parent but has many relevant descendant nodes; 3) select the node over its descendants when it has many relevant descendants evenly distributed in its sub-tree. At the second stage, a brute-force filtering is carried out from bottom up over the result tree. A node having higher relevant score than all descendants will be retained. Otherwise, it is removed from the result set.

- remove overlapping elements by two-step post filtering based on refined relevance score. At the first step, a new relevance score is computed for each retrieved element in a bottom-up manner from the leaves to the highest overlapping free nodes. The score value is either the max or the arithmetic average of the relevance score of its own and all descendants. At the second step, either the highest ancestors or the most relevant overlapping free nodes are selected from the newly formed answer list [39].

5 Entity Ranking

Entity ranking aims at locating the finest information granularity (e.g. *named entities*) from text descriptions (e.g. Web pages) as an answer to fact-based and short-answer questions. This task is relatively newer to other focused search tasks. It appeared in the TREC enterprise track as the expert search task in 2005 and in INEX as the entity ranking task in 2007. The expected entities are described by short labels or precise Web pages (e.g. an article in the Wikipedia collection) whose primary purpose is to serve as a unique and complete description of an entity [45]. Entities provide *types* of information [58] such as named entity types (e.g. *person*, *organization*, *product*, *location*, *nationality*), nominal entity types (e.g. *GPE*, *substance*, *plant*, *animal*, *person*), and numeric types (e.g. *date*, *time*, *money*, *quantity*, *ordinal and cardinal*). They are scored and ranked by their relevance to a given query. For example, a TREC query searching for *organization* entity that has been started by *person* Michael Stonebraker looks like the following:

```
<query>
<entity_name>Michael Stonebraker</entity_name>
<entity_URL>
http://www.csail.mit.edu/people/Michael_Stonebraker/
</entity_URL>
```

```
<target_entity>organization</target_entity>
<narrative>
Which database companies have been started by Mike Stonebraker?
</narrative>
</query>
```

The retrieved relevant entities are:

```
Vertica Systems
Streambase Systems
Relational Technology, Inc.
```

Depends on the type of searched entities, approaches for traditional document retrieval can be tailored for the similar problem in entity ranking in literature. The most active research field is searching for people with expertise that is relevant to the topic of interest (expert finding). The latest approaches for this task are given as follows:

- Expert finding on the Web can be treated as the task of document clustering with the assumption that similar documents tend to represent the same person [4]. The traditional clustering algorithms and probabilistic latent semantic analysis can be applied.

- The language modeling framework has been adapted for *document-centric* (e.g. [5], [46], [14]) or *profile-centric* (e.g. [47], [5], [38]) models for expert finding. The *document-centric* approach models an expert's knowledge (expert profile) from associated documents and ranked profiles using document retrieval techniques. The association between the document and the candidate expert can be any textual evidence such as document containing the candidate's name, emails sent and received by the candidate, the candidate's home page, Web pages visited by the candidate, doument written by the candidate, etc.. The *profile-centric* locates documents on topic and finds the associated expert. Empirically the *document-centric* approach is superior to the *profile-centric* approach [5].

- The *document-centric* approach can also be implemented on the top of other document retrieval models (e.g. BM25, PL2, DLH13). To boost the expert ranking performance, an aggregated view of associated documents can be used as votes for the candidate expert. Besides, document structures (e.g. title, anchor text of incoming hyperlinks) can also bring improvement as they do in the traditional document retrieval [32].

- Besides the direct association between relevant documents and candidate experts, there are many indirect connections (e.g. directly and indirectly linked documents or persons, the mailing list) which form a more complicated social network. The link analysis approach uses such link structure to propagate relevance information for ranking experts in a specified topic. For instance, experts can be found by computing their centrality in the organizational social network [22], [59], [10]. The most popular methods for web link analysis (e.g. PageRank algorithm and HITS algorithm) are applicable for finding an expert in the social network. In addition to the social network, another connection [48] can be derived from the initial result of the traditional document retrieval on a given topic. From the ranked documents, a second set of contained candidate experts is extracted. Their containment relations are represented in an *expertise graph.* Then the relevance probability can be propagated from documents to the related candidates by multi-step random walks on this graph for finding an expert.

- In addition to other focused retrieval techniques, natural language processing (NLP) techniques can be exploited to recognize and classify named entities within the text corpus [26].

A few work [17] extended expert search to other entity search (e.g. time search). Besides, many entities of a fact type are searched by factoid and list queries in the question answering task [55].

The entity-specific models (e.g. expert finding) are not directly applicable for ad-hoc entity ranking because there will be too many models to be built for each type. In this scenario, the work [58] proposed three approaches: 1) rank passages based on their relevance to a query and then rank entities by the maximum score of the passage in which the entity appears; 2) create a bipartite graph between every passage and every entity and apply different graph centrality measures to rank entities in the graph; 3) rank entities by computing their correlation to the query on the Web using correlation measures (e.g. Jaccard-coefficient). In another work [9], an information retrieval engine is built to rank any type of entities by proximity features (e.g. aggregation function of the selectors, their frequency in the corpus, and their distance from the candidate answer).

6 Open Problems

There are notable techniques having been proposed for different focused retrieval tasks. They mainly depend on the traditional approaches for document retrieval to estimate relevance. The retrieval cue is still dominated by the document text. Attempts have been made to use other evidences such as the URL, manually assigned metadata, and the document structure (e.g. XML documents) for document retrieval but have not been enough yet for focused search. The retrieval results for XML elements and entities are still far from desired. Apart from the ranking models, the presentation of the retrieval results (interface) and user interaction are not well studied. To have a full access to document fragments at different level of granularity, the following questions need to be answered:

- How to express a user's search request in a simple query language (e.g. natural language) so that both content and structure constraints are clear? As we know, ordinary users are not good at formulating their complex information needs in keywords. It is even harder for them to specify their structure constraints in a complicated query language.

- What is the proper granularity of document fragments (e.g. paragraph, section) to answer a user's request? If the retrieved information is too large, a user still has to make effort to locate her interested piece. If it is too small, some relevant information may be lost or a user can not judge its relevance because of lack of information context.
- For XML retrieval, there are many indexing strategies with which different ranking strategies are designed. The choice has to be made based on the collection, the types of elements (i.e., the DTD) and their relationships. It is interesting to investigate all indexing strategies within a uniform and controllable environment to determine those leading to the best performance, across or depending on the ranking strategies [27].

7 Conclusion

In this report, we review the state-of-the-art techniques for focused search. They are presented and discussed according to the different requirements from different retrieval tasks. We start from the biggest retrieval unit, topic-specific document retrieval, then passage retrieval, XML element retrieval, and end at the smallest retrieval unit, entity ranking. In order to pinpoint the relevant document fragments, efforts have been made for both the document representation (e.g. XML markups) and retrieval techniques. The need for focused result is common for search in digital library but new for ordinary users who search for specific information on the heterogeneous Web. Though research on focused search is making progress, researchers need to pay special attention on some fundamental problems while inventing more effective retrieval models.

References

1. Ashoori, E., Lalma, M., Tsikrika, T.: Examining Topic Shifts in Content-Oriented XML Retrieval. International Journal on Digital LIbraries. 8(1), pp. 39-60. (2007)
2. Azzopardi, L., Girolami, M., van Rijsbergen, C.J.: Topic Based Language Models for Ad Hoc Information Retrieval, In *Proceedings of IJCNN*, pp. 3281-3286. (2004)
3. Bai J., Nie, J.Y., Cao, G., Bouchard, H.: Using Query Contexts in Information Retrieval, In *Proceedings of SIGIR*, pp. 15-22. (2007)
4. Balog, K., Azzopardi, L.A., de Rijke, M.: Resolving Person Names in Web People Search, Weaving Services, Location, and People on the WWW. (2009)
5. Balog, K., Azzopardi, L., de Rijke, M.: Formal Models for Expert Finding in Enterprise Corpora, In *Proceedings of SIGIR*, pp. 43-50. (2006)

6. Betsi, S., Lalmas, M., Tombros, A., Tsikrika, T.: User Expectations from XML Element Retrieval. In *Proceedings of SIGIR*, pp. 611-612. (2006)
7. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation, J. Mach. Learn. Res., 3, pp. 993-1022. (2003)
8. Broschart, A., Schenkel, R., Theobald, M., Weikum, G.: In *Proceedings of INEX Workshop*, pp. 49-56. (2007)
9. Chakrabarti, S., Puniyani, K., Das, S.: Optimizing Scoring Functions and Indexes for Proximity Search in Type-annotated Corpora. In *Proceedings of WWW*, pp. 717-726. (2006)
10. Chen, H., Shen, H., Xiong, J., Tan, S., Cheng, X.: Social Network Structure behind the Mailing Lists: ICT-IIIS at TREC 2006 Expert Finding Track, In *Proceedings of TREC*, (2006)
11. Callan, J.P.: Passage-level Evidence in Document Retrieval. In *Proceedings of SIGIR*, pp. 302-310. (1994)
12. Clarke, C.L.A.: Controlling overlap in content-oriented XML retrieval. In *Proceedings of SIGIR*, pp. 314-321. (2005)
13. Cui, H., Sun, R., Li, K., Kan, M.Y., Chua, T.S.: Question Answering Passage Retrieval Using Dependency Relations In *Proceedings of SIGIR*, pp. 400-407. (2005)
14. Fang, H., Zhai, C.X.: Probabilistic Models for Expert Finding. In *Proceedings of ECIR*, pp. 418-430. (2007)
15. Geva, S.: GPX - Gardens Point XML IR at INEX 2005. In *Proceedings of INEX Workshop*, pp. 240-253. (2005)
16. Gvert, N., Kazai, G.: Overview of the INitiative for the Evaluation of XML Retrieval. In *Proceedings of INEX*, pp. 1-17. (2002)
17. Hu, G.P., Liu, J.J., Li, H., Gao, Y.B., Nie, J.Y., Gao, J.F.: A Supervised Learning Approach to Entity Search, Information Retrieval Technology, 4182, pp. 54-66. (2006)
18. Huang. F.: Using Language Models and Topic Models for XML Retrieval. In *Proceedings of INEX Workshop*, pp. 94-102. (2007)
19. Huang, W., Trotman, A., OKeefe, R.: Elemement Retrieval using a Passage Retrieval Approach. In *Proceedings of ADCS*, pp. 80-83. (2006)
20. Itakura, K.Y., Clarke, C.L.A.: From Passages into Elements in XML Retrieval. In *Proceedings of SIGIR Workshop on Focused Retrieval*, pp. 17-22. (2007)
21. Jiang, J., Zhai, C.X.: Accurately Extracting Coherent Relevant Passages Using Hidden Markov Models. In *Proceedings of CIKM*, pp. 289-290. (2005)
22. Jurczyk, P, Agichtein, E.: Discovering Authorities in Question Answer Communities by Using Link Analysis. In *Proceedings of CIKM*, pp. 919-922. (2007)
23. Kamps, J., Geva, S., Trotman, A., Woodley, A., Koolen, M.: Overview of the INEX 2008 Ad Hoc Track. In *Pre-Proceedings of INEX*, pp. 1-28. (2008)
24. Kamps, J., Koolen, M., Lalmas, M.: Locating Relevant Text within XML Documents. In *Proceedings of SIGIR*, pp. 847-848. (2008)
25. Kamps, J., de Rijke, M., Sigurbjornsson, B.: Length Normalization in XML Retrieval. In *Proceedings of SIGIR*, pp. 80-87. (2004)
26. Krishnan, V., Das, S., Chakrabarti, S.: Enhanced answer type inference from questions using sequential models. In *Proceedings of EMNLP/HLT*, pp. 315-322. (2005)
27. Lalmas, M.: XML Information Retrieval. Encyclopedia of Library and Information Sciences, M.J. Bates and M.N. Maack (Eds), Taylor & Francis Group. (2009)

28. Lalmas, M., Tombros, A.: INEX 2002 - 2006: Understanding XML Retrieval Evaluation. Digital Libraries: Research and Development, C. Thanos, F. Borri, and L. Candela (Eds.), Springer-Verlag Berlin Heidelberg. (2007)
29. Li, R.M.: Improving Web Page Retrieval using Search Context from Clicked Domain Names. In *Proceedings of DEXA Workshop on Text-based Information Retrieval*, (2009) to appear
30. Li, R.M., Kaptein, R., Hiemstra, D., Kamps, J.: Exploring Topic-based Language Models for Effective Web Information Retrieval. In *Proceedings of DIR*, pp. 65-71. (2008)
31. Liu, X.Y., Croft, W.B.: Cluster-based Retrieval Using Language Models. In *Proceedings of SIGIR*, pp. 186-193. (2004)
32. Macdonald, C., Ounis, I.: Voting for Candidates: Adapting Data Fusion Techniques for an Expert Search Task. In *Proceedings of CIKM*, pp. 387-396. (2006)
33. Mass, Y., Mandelbrod, M.: Using the INEX Environment as a Test Bed for Various User Models for XML Retrieval. In *Proceedings of INEX Workshop*, pp. 187-195. (2005)
34. Mass, Y., Mandelbrod, M.: Component Ranking and Automatic Query Refinement for XML Retrieval. In *Proceedings of INEX Workshop*, pp. 73-84. (2004)
35. Mass, Y., Mandekbrod, M.: Retrieving the most relevant XML Components. In *Proceedings of INEX Workshop*, pp. 53-58. (2003)
36. Mihajlović, V., Ramírez, G., Westerveld, T., Hiemstra, D., Blok, H.E., de Vries, A.: TIJAH Scratches INEX 2005: Vague Element Selection, Image Search, Overlap, and Relevance Feedback. In *Proceedings of INEX Workshop*, pp. 72-87. (2005)
37. Ogilvie, P., Callan, J.: Hierarchical Language Models for XML Component Retrieval. In *Proceedings of INEX Workshop*, pp. 224-237. (2005)
38. Petkova, D., Croft, W.B.: Hierachical Language Models for Expert Finding in Enterprise Corpora. In *Proceedings of ICTAI* pp. 599-608. (2006)
39. Popovici, E., Mnier, G., Marteau, P.F.: SIRIUS XML IR System at INEX 2006: Approximate Matching of Structure and Textual Content. In *Proceedings of INEX Workshop*, pp. 185-199. (2006)
40. Reid, J., Lalmas, M., Finesilver, K., Hertzum, M.: Best entry points for structured document retrieval - part ii: Types, usage and effectiveness, Inf. Process. Manage. 42(1):89-105. (2006)
41. van Rijsbergen, C.J.: Information Retrieval, Butterworths, London. (1979)
42. Rode, H.: From Document to Entity Retrieval: Improving Precision and Performance of Focused Text Search, Ph.D. thesis, University of Twente. (2008)
43. Salton, G., Allan, J., Buckley, C.: Approaches to Passage Retrieval in Full Text Information Systems. In *Proceedings of SIGIR*, pp. 49-58. (1993)
44. Sauvagnat, K., Boughanem, M., Chrisment, C.: Answering Content and Structure-based Queries on XML Documents using Relevance Propagation, Information System, 31, pp. 621-635. (2005)
45. Serdyukov, P.: Search for Expertise Going beyond Direct Evidence, Ph.D. Thesis, University of Twente. (2009)
46. Serdyukov, P, Hiemstra, D.: Being Omnipresent To Be Almighty: The Importance of the Global Web Evidence for Organizational Expert Finding. In *Proceedings of SIGIR Workshop on Future Challenges in Expertise Retrieval*, (2008)
47. Serdyukov, P, Hiemstra, D.: Modeling Documents as Mixtures of Persons for Expert Finding. In *Proceedings of ECIR*, pp. 309-320. (2008)

48. Serdyukov, P., Rode, H., Hiemstra, D.: Modeling Multi-step Relevance Propagation for Expert Finding. In *Proceedings of CIKM*, pp. 1133-1142. (2008)
49. Sun, R.X., Ong, C.H., Chua, T.S.: Mining Dependency Relations for Query Expansion in Passage Retrieval. In *Proceedings of SIGIR*, pp. 382-389. (2006)
50. Tellex, S., Katz, B., Lin, J., Fernandes, A., Marton, G.: Quantitative Evaluation of Passage Retrieval Algorithms for Question Answering. In *Proceedings of SIGIR*, pp. 41-47. (2003)
51. Theobald, M., Schenkel, R., Weikum, G.: TopX & XXL at INEX 2005, In *Proceedings of INEX Workshop*, pp. 282-295. (2005)
52. Trotman, A., Lalmas, M.: Why Structural Hints in Queries do not Help XML-Retrieval, In *Proceedings of SIGIR*, pp. 711-712. (2006)
53. Trotman, A., Geva, S.: Passage Retrieval and other XML-Retrieval Tasks. In *Proceedings of SIGIR Workshop on XML Element Retrieval Methodology*, pp. 43-50. (2006)
54. Trotman, A., Sigurbjrnsson, B.: Narrowed Extended XPath I (NEXI). In *Proceedings of INEX Workshop*, pp. 16-40. (2005)
55. Voorhees, E.M, Dang, H.T.: Overview of the TREC 2005 Question Answering Track. In *Proceedings of TREC*, (2005)
56. Wang, M.Q., Si, L.: Discriminative Probabilistic Models for Passage Based Retrieval. In *Proceedings of SIGIR*, pp. 419-426. (2008)
57. Wei, X., Croft, W.B.: Investigating Retrieval Performance with Manually-built Topic Models. In *Proceedings of RIAO*, pp. 12. (2006)
58. Zaragoza, H., Rode, H., Mika, P., Atserias, J., Ciaramita, M., Attardi, G.: Ranking Very Many Typed Entities on Wikipedia. In *Proceedings of CIKM*, pp. 1015-1018. (2007)
59. Zhang, J., Ackerman, M.S., Adamic, L.: Expertise Networks in Online Communities: Structure and Algorithms. In *Proceedings of WWW*, pp. 221-230. (2007)
60. van Zwol, R.: In *Proceedings of INEX Workshop*, pp. 146-160. (2005)