# Bullseye: A Passage Retrieval and Highlight Algorithm Based On Document Structure

Xi Zheng
The Center for Advanced Research in Software Engineering
The Univeristy of Texas at Austin
jameszhengxi1979@gmail.com

Akanksha Bansal
Department of Computer Science
The Univeristy of Texas at Austin
akankshabansal90@gmail.com

## 1. SUMMARY

### 1.1 Introduction

Search engines like Google have made the lives of researchers easy by providing them with specialized search system like Google Scholar. Now one can find relevant work being done in any of the domains. However analyzing the relevance of documents takes much more time because a researcher has access to huge number of repositories. Additionally, one needs to go over the whole document just to realize that the document returned is irrelevant, which becomes frustrating with time.

As part of this project, we want to build a tool that will help users in making the said decision. Passage retrieval based on structural information in documents has long been suggested as an effective way to retrieve elements of a document with finer granularity. With the help of passage retrieval algorithms, we want to highlight the text that is relevant to the user based on the query, thus making it easier for him to understand the document. We posit that the search engine will not only return the user the best ranked documents but will also be part of the user decision making process.

One might argue that if the search engine just returns passages which it deems to be relevant instead of an document, then such a tool is irrelevant. Because then the user would be able to find relevant material much faster. But this approach leads to loss of context with regards to the usage of the data. Kamps et al. [7] found out that users and assessors still regard whole articles as the meaningful unit of retrieval for XML collection, which adds to our intentions.

Review of a few decades of science research papers suggests that common section headers found in the science research papers have minimally changed over the past decades. Identification of sections within research papers provides important context for locating relevant information. For example, the section *Related work* contains the state of the art literature review in the field which the research paper addresses, while *Evaluation* contains the test methodologies and tools used to support the main research hypotheses and findings in the paper. If the user is interested in the current status quo of the keyword, he most likely shall be able to locate the information in the *Related work* while the user shall be more likely to find information in *Evaluation* if he is particularly interested in how the keyword is related to experiments.

Thus the scope of our research is focused on answering the below research questions:

- How to classify and localize the structural information embedded in documents for passage-level retrieval?
- Whether a retrieval system can improve user satisfaction by combining content and structural conditions in passage-level retrieval?
- Whether a retrieval system can enhance effectiveness by combining content and structural conditions in passage-level retrieval?
- Whether highlighting passage retrieved can improve user satisfaction?

As a first step, we want to build a baseline passage retrieval algorithm to highlight the data which are most relevant to users (documents are research papers) based on their queries.

As a second step, we build the Bullseye algorithm on top of the baseline algorithm. It will highlight the relevant text only from the sections of the document that user wants to find. Search engines nowadays are anyways generating structure maps after analyzing the documents so as to be able to decide if the given document is relevant or not. As part of Bullseye algorithm, we are proposing that they allow the user to specify which section of the document he wants the results from.

### 1.2 Related Work

Hildreth et al. [6] argued that evaluation studies that relied on measures such as user perception of ease of use and subjective satisfaction with the search results did not provide a clear and consistent answer as to how user satisfaction may predict their actual search effectiveness. He found that user perception of ease of use had an effect, possibly greater than the results themselves, on user satisfaction. Kamps et al. [7] listed two ways of Information Retrieval, which can be performed for text :

- *Full Document Retrieval System* A baseline is formed by using a standard document index in which only whole documents are considered as a retrievable unit.
- *Element Retrieval System* The documents are indexed into retrieval units. Elements rather than documents then become the returned results of the user query.

In this paper, we concentrate on passage-level retrieval and highlight. In the measurement, we not only cover user perception of ease of use and satisfaction with the search results, but also cover more objective metrics (e.g., exhaustivity and specificity as covered in Section 1.5).

Lalmas et al. [8] mentioned two types of topics being identified by INEX:

- *Content-only (CO)* topics are requests that do not include reference to the document structure. They are traditional topics used in information retrieval test collections. However, the results to such topics are elements of various complexity, e.g. at different levels of the XML documents structure.

- *Content-and-structure (CAS)* topics are requests that contain conditions referring both to content and structure of a document. These conditions may refer to the content of specific elements, or may specify the type of the requested answer elements (e.g. sections should be retrieved).

The baseline algorithm is closely related to answering queries in INEX Content-only(CO) topics where the user requests do not include reference to the document structure [8]. The Bullseye algorithm is closely related to answering queries in INEX Content-and-structure topics where the user requests do contain conditions referring both to content and structure of document [8].

Cui et al. [3] proposed a new hierarchical index propagation and pruning mechanism for structured documents and realized a flexible element retrieval system based on this index structure. Their work was carried on XML documents,whereas our research focuses on PDF documents.

Denny et al. [4] designed a SecTag algorithm to identify both labeled and un-labeled (implied) note section headers in "history and physical examination" documents. Their work calculated Bayesian probability for all of a document's section header candidates while our work is much more straightforward as each section header in a research paper is labeled. Our work of automatically identifying section headers is largely built upon the SecTag's method.

## 1.3 Implementation

For the baseline algorithm, we presume user has already got a list of relevant documents for the query from a popular search engine like Google. So the main implementation lies in applying a search algorithm which highlights those relevant parts in the documents.

Based on Tellex's *evaluation of passage retrieval algorithms for Question Answering* [12], we want to implement the passage retrieval algorithm based on the MITRE [9]. This approach simply matches stemmed words between question and answers. It counts the number of terms a passage has in common with the query, where each sentence is treated as a separate passage. This algorithm represents the simplest reasonable passage retrieval technique and serves as a good baseline for comparison. In the baseline algorithm implementation, we also remove the stop words and aggregate sentences where query terms span across.

### 1.3.1 Bullseye

Bullseye is designed to be able to locate section headers as specified by the users to narrow down the search (This is based on the assumption when searching for keywords in the research papers, users have a clear idea of what sections to look for). The Bullseye takes two specific arguments: a search query and sections of the document user wants to restrict his search to. Based on the user query, Bullseye will retrieve relevant passages from the sections specified and then it will highlight the passages retrieved in the document based on the baseline algorithm.

To develop the Bullseye algorithm, we download 200 research papers in Software Engineering from a few top conferences (e.g. FSE, ICSE and ASE) and a few specialized conferences (e.g. ICCPS, MobiCom, OOPSLA) as a training set. By observing the training set, we determine that section headers in the research papers can be generalized into the following sets to satisfy users' information need.

- *Abstract*: It usually contains high level information regarding the keywords searched by users.

- *Introduction*: It often provides context and problem domain of the keywords searched by users.

- *Related Work*: It is highly likely to locate the state of the art for the keywords searched by users.

- *Implementation*: It usually provides how the keywords searched are used in the implementation.

- *Evaluation*: It is likely to provide information of how the keywords are applied in the experimentation.

- *Conclusion/Future Work*: It might give novel ideas of how the keywords can evolve in the future.

We define this set (Fixed Set) as the user search condition for section headers.

We find out that many research papers in the training set have section headers which are either exact matches or equivalent terms (e.g. *Summary* is an equivalent term to *Conclusion/Future Work*) to the headers in the Fixed Set. We also notice some of the research papers have a few random section headers (Implicit Section Headers) which do not match explicitly with headers in the Fixed Set or could not be assigned as generic equivalent terms. However, we are able to identify the following patterns which are later incorporated into the Bullseye (as decision trees) to establish implicit matching.

(1) Section headers have implicit hierarchical structure (e.g. *Implementation* is always before *Evaluation*).

(2) Those implicit Section Headers often have many to one relationship to the sections in the Fixed Set (e.g. two implicit sections is mapped to *Implementation*).

(3) The implicit section headers always fall into one of three section headers in the Fixed Set, namely *Related Work*, *Implementation* and *Evaluation*.

(4) Not all the section headers in the Fixed Set are available in some research papers (e.g. Some research papers simply do not have *Evaluation*).

(5) *Related Work* section either appear before the *Implementation* or after *Evaluation*. But in all explicitly matched scenarios, *Related Work* always appears before *Implementation*.

(6) *Implementation* and *Evaluation* almost have equal probabilities of having multiple implicit section headers matched.

(7) *Related Work* has much lower probability of having multiple implicit section headers matched compared with *Implementation* and *Evaluation*.

Based on these hierarchical and statistical information for the section headers in the research papers, we built explicit and implicit section header matching algorithm into the Bullseye. As a first step, the Bullseye algorithm will process the document, sentence by sentence, to find all possible section headers. A section header is one that marks the beginning of sentences and consists of only Capital letters and a leading number or Roman numbers (From manual observation, we found all the papers in the training set have major sections defined in this format). All the possible section headers are inserted into a linked list which preserves the insertion order. Each section header candidates are matched against the equivalent terms for sections defined in the Fixed Set. The matched results are recorded into the linked list. Bullseye will then check in the linked list whether it can find matching for all the section search conditions specified by the user. If it could not locate all matching, Bullseye uses the decision trees based on the hierarchical and statistical information found previously to locate section headers specified by the user. Fig. 1 is one of the decision trees in the Bullseye algorithm. This decision tree specificially is used to determine implicit *Related Work* and *Implementation* when *Evaluation* section is found through explicit matching process. The decision process is entirely based on the hierarchical and statistical information retrieved in the training set.

Through the explicit and implicit section header matching, Bullseye maintains a state machine of how many potential section headers have been explicitly matched and how many are implicitly matched. The intersection of matched section headers in the state machine and user specified section query condition are used to pinpoint those sections for locating the keywords submitted in the user query.

### *1.3.2 Tools Used*

KSTEM [2] is an open source Java tool to stem words and Frank [5] developed open-source software which removes stop words. We use both in our implementation.

The Apache PDFBox [10] library is an open source Java tool for working with PDF documents. We use this library to extract sentences from PDF documents and highlight sentences in the documents.

## 1.4 Future Work

We wish to port Bullseye algorithm to a Google Chrome Application so after a user retrieves the documents from a search engine (e.g. Google Scholar), the browser can automatically highlight the most relevant information for the user. The application would have the feature that a user can specify some additional search conditions as to which sections of the paper the user is more interested in. This will provide a seamless integration of document retrieval (e.g., provided by the search engine) and passage retrieval (e.g., provided by the Bullseye) in the Chrome Browser (shall be a better user experience).

## 1.5 Evaluation

As for any tool designed to enhance user experience, our evaluation will be based on measuring user satisfaction. With evaluation metrics(e.g., user effort as covered later), we will try to capture if users feel that Bullseye enhances their satisfaction with a search engine (e.g. Google Scholar). The user study will help us understand how effectively our algorithm

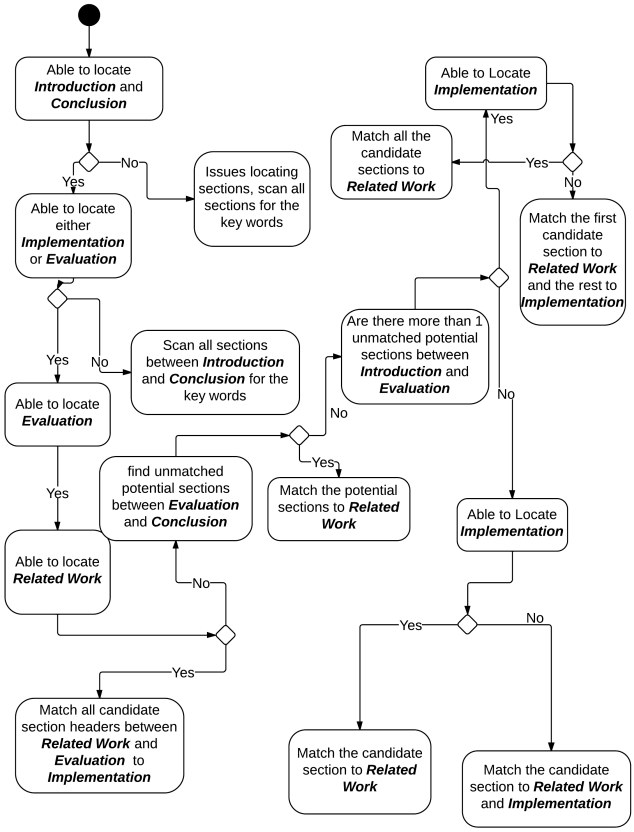


**Figure 1: Decision Tree**

improves user's experience by highlighting relevant passage.

We will not consider documents from multiple languages, nor are non-structured documents going to be part of our study. The main reason for our choice is because research documents are fairly long and more importantly can be broken into understandable sections. Our algorithm has been specifically designed for scientific research papers and users of Google Scholar, and we feel that assuming all the research papers retrieved by Google scholar are structured, is safe.

Since there is no existing test collection available, which consists of a diverse enough pdf documents, with a set of queries with structural constraints and a gold set of relevant information annotated, we will generate the test collection ourselves for the preliminary evaluation and generate test collection based on participants input in the user study.

The preliminary evaluation is conducted by us without any participants being involved, which includes processing of arbitrary queries that specify not only a retrieval element at any level in research documents but also some structural conditions (i.e., to search inside *Implementation* and *Evaluation*) in themselves. The purpose will be to validate the algorithm in a qualitative sense and see if the retrieval results are reasonable. We will download another 50 papers from the same sources as in the training set (e.g., confer-

ence papers from Software Engineering), to generate queries containing sections from the documents and annotate the documents to create the gold set as relevance judgement.

Maskari et al. [1] reported that user satisfaction is measured in terms of the following four factors: system effectiveness, user effectiveness, user effort, and user characteristics. Our tool focuses on improving :

- *User effectiveness* is defined as the accuracy and completeness with which users achieve certain goals. It has the following user effectiveness measuring criteria:

  (a) the number of tasks successfully completed,

  (b) the number of relevant documents obtained, and

  (c) the time taken by users to complete set tasks.

  User effectiveness is directly influenced by the amount of time required to find the information sought: the less time spent searching, the greater the effectiveness. Our tool aims at helping the users by reducing the time taken to search for the given topic. We will measure *User effectiveness* by asking users to quantify the time reduction using Bullseye and the baseline algorithm to make a relevance judgment for the given document in a graded scale:

  (1) highly reduced

  (2) slightly reduced

  (3) no change

  (4) increased

- *User effort* is defined as the total effort taken by the user to complete the given task. It quantifies user effort using the number of clicks, the number of queries and the number of query reformulations, and rank position accessed to obtain relevant information. We will measure *User effort* by asking participants to quantify their overall user satisfaction to make a relevance judgment using Bullseye and the baseline algorithm for the given document in a graded scale:

  1 highly satisfied

  2 slightly satisfied

  3 not satisfied

Maskari et al. [1] also mentioned the actual contribution of the system is ambiguous and difficult to quantify from the users perspective, because users tend to discount the contribution of the computer system when things go well and to blame the system when things go poorly. Therefore for our user study, we not only measure the above two metrics but also bringing the relevance measurements into the evaluation, which are used initially to evaluate XML Retrieval Effectiveness at INEX [8].

By the relevance measurements [8], relevance can be grouped into two categories :

1. *Topical relevance*, which reflects the extent to which the information contained in an element satisfies the information need, i.e. measures the exhaustivity of the topic within an element.

2. *Component coverage*, which reflects the extent to which an element is focused on the information need, and not on other, irrelevant topics, i.e. measures the specificity of an element with regards to the topic.

We want users to be able to judge if the passages retrieved are able to cover topical relevance or are more towards component coverage.

This is measured by defining the following two dimensions:

1. *Specificity*, which measures the extent to which an element focuses on the topic of the request.

2. *Exhaustivity*, which measures how exhaustively an element discusses the topic of the user's request.

For all elements within the highlighted passages, the participants will be asked to assess *Specificity* based on the below scale:

1 Not specific (0): the topic of request is not a theme discussed in the element.

2 Marginally specific (1): the topic of request is a minor theme discussed in the element.

3 Fairly specific (2): the topic of request is a major theme discussed in the element.

4 Highly specific (3): the topic of request is the only theme discussed in the element.

For all elements within highlighted passages, the participants will be asked to assess their *Exhaustivity* based on the below scale:

1 Highly exhaustive (2): the element discussed most or all aspects of the query.

2 Partly exhaustive (1): the element discussed only few aspects of the query.

3 Not exhaustive (0): the element did not discuss the query.

Our evaluation metric will be based on the Exhaustive and Specificity metrics used in INEX 2003 and INEX 2005 [8]. We will use the value $quant_{gen}$ to measure the relevant level for the passages that have been deemed to be relevant to the users. The generalized ($quant_{gen}$) [8] rewards retrieved elements according to their degree of relevance, thus allowing to reward fairly and marginally relevant elements. $Quant_{gen}$ shows slight preference towards the exhaustivity dimension, assigning high scores to exhaustive, but not necessarily specific elements. It is measured from the values of exhaustivity and specificity calculated based on the above mentioned scales.

$$quant_{gen} = \begin{cases} 1 & if(e,s) = (3,3) \\ 0.75 & if(e,s) \in \{(2,3),(3,\{2,1\})\}; \\ 0.5 & if(e,s) \in \{(1,3),(2,\{2,1\})\}; \\ 0.25 & if(e,s) \in \{(1,2),(1,1)\}; \\ 0 & if(e,s) = (0,0) \end{cases}$$

In the user study, we ask the participants to provide us with the test collection (document sets, content and structural queries, and manually annotated gold set). It is expected that users who are familiar with search topics would be able to judge the effectiveness of the system more prudently than unfamiliar users, which might introduce unnecessary confounding variable into the evaluation. To mitigate this risk, we will have each participant evaluate additional 5 query and document sets which are not provided by the user.

### 1.5.1 Empirical Design

The goal in the user study is to answer the following research questions.

- Whether Bullseye can improve user satisfaction and effectiveness?
- Whether Bullseye can improve the relevant level for the passages retrieved in terms of $quant_{gen}$?
- Whether highlighting the retrieved passages, by itself, can improve user satisfaction?

Various IR evaluation studies indicate that users are overly impressed with new electronic retrieval technologies, and this accounts for inflated levels of satisfaction with actual search results. In order to account for this bias, we plan to conduct our tests with users who have wide exposure to various evaluation techniques and would thus be able to provide us with a relatively less biased evaluation of our tool.

Around 20 participants will be recruited for this user study. All participants will be in the range of between the ages of 20 and 35 years old. Each participant is required to submit a query containing the keywords and relevant sections they want to search for, a document downloaded from Google Scholar by searching the keywords, a "gold" true optimal passages that have been annotated, which are the most relevant based on the query and the document downloaded. We will collect this information from each participant to create a test collection. Singhal et al. [11] observed that the likelihood of a document being judged relevant by a user increases with the document length. Based on this information, we restrict the document submitted by the participants to be minimum 6 pages and maximum 10 pages (excluding reference pages) to rule out possible bias introduced by using documents of various length, which in turn would affect the quality of the passage retrieval as the document itself might carries bias (documents with more pages might have higher chance of being chosen irrespective of relevant level while very small documents containing a very few words have too little information to be useful to a user [11]).

For each participant's submission (document and query), we will run the baseline algorithm and Bullseye, to create two separate documents (Algorithm Documents) with random names. We will have a registry to associate each randomly generated file with the algorithm creating it (e.g., a pair of file name and the algorithm name). For each topic (all together 20 topics), there is a set (Test Set) which contains the original document, query, "gold" true passages and these two documents generated by the algorithms. For each participant, we will give the participant six Test Sets, with one set containing his own submission. The participant will evaluate each Algorithm Document based on the "gold" true passages, the document and the query. The evaluation metrics are user effectiveness, user effort, specificity and exhaustivity as covered previously.

Before the user study, we will organize a training session with all the participants regarding the overall aim of the study, the evaluation metrics, evaluation process and the submission requirement of the user study. Each participant will be made aware that the Test Set to be provided with two Algorithm Documents with relevant passages highlighted by Bullseye algorithm and the baseline algorithm. Then we will show them how they are supposed to evaluate the document set. In this way, we can train each participant how to create gold set by himself and compare the gold set with the algorithms' output.

We will ask participants whether highlighting the passage retrieved in the document is better than simply returning the passage retrieved without the document. The participants will be asked to assess this based on the below scale:

1 much better

2 slightly better

3 no difference

4 worse

We will ask one more question for the participants to justify our future work:

(a) If this tool were to be integrated with the Google Scholar search results, would it improve your satisfaction with the search results ?

### 1.5.2 Evaluation procedure

We want to keep the test setup required by the participants simple and less time consuming so that we ask the participants to email us the submission(document, query and the gold set) instead of having to come in person to a controlled environment for the user study. After having generated the Test Sets for all the topics, we then email the participants the six Test Sets and spreadsheet for the measurement. The participants can then email us just the measurement spreadsheet, based on which we will generate the following graphs to analyze the results of the user study:

- A Box-and-Whisker Plot with the details of User Effort calculated for both Bullseye and baseline document.
- A Box-and-Whisker Plot with the details of User Effectiveness calculated for both Bullseye and baseline document.
- A Box-and-Whisker Plot for the $Quant_{gen}$ calculated for both Bullseye and baseline document.
- A Box-and-Whisker Plot with the details of the users feedback for highlighting the passage retrieved within the document.

## 1.6 Challenges

We have resolved these research challenges in the last status report. However, there are a few new research challenges, some of which have been resolved and some of which target the future work.

1. How Bullseye can locate implicit section headers? - Resolved
2. How to highlight relevant passages in the PDF by only giving highlighted texts as inputs? - Resolved
3. How to make the user study easy for the participant? - Resolved
4. What is the ideal size of document as input to the user study? - Resolved
5. How to create test collection for Bullseye? - Resolved

6. How we can port the Bullseye algorithm into a Chrome application? - Future Work

7. Can we use the Bullseye algorithm for research documents other than software engineering? - Future Work

## 2. MILESTONES

There are many unexpected research and engineering challenges, but we were able to resolve them within the original time-line for the second milestone, which is from Oct/24 to Nov/7. We spent a whole week thinking about the details of the user study and possible risks associated with it. We believe it is doable and we will carry out the user study, analyze the result, write the poster in the last milestone which is from Nov/7 to Nov/28. We will spent Nov/28 and Dec/5 to write the full paper, which hopefully can be a good base for a short paper submission to SIGIR 2014.

In the second milestone, we divided the work similarly as to what had been implemented for the implementation of the first milestone. The challenges which were relevant to the technical aspects of the algorithm, were be handled by James, and the aspects which were be dealing with the parsing the information from the documents itself were handled by Akanksha. Both of us contributed 50% to the implementation, design and related works reading.

As far as division of work with respect to the user study is concerned, Akanksha will take the responsibility of recruiting participants needed for the user study and James will analyze the results of the user study. Both of us will contribute 50% to the final write-up of posters and paper.

## 3. RISKS

We provide a novel algorithm to highlight passages in the research papers to help users to locate most relevant information with very little effort; and the user study will substantiate this claim. But this task comes with the following risks.

The implicit section location algorithm of Bullseye is based on manual analysis of the training set. The derived rules are implemented in the form of decision trees. There might be some exceptions to the decision trees introduced by some other documents not included in the training set.

The second risk lies in choosing the right test collection for evaluation. We plan to ask our user study participants to prepare their own topics and create gold true passages themselves. But will this be a sufficiently representative test is yet to be determined.

Then we run in the risk of not being able to highlight all the sections which are important for the user and might lead to a user experience opposite of what we want (e.g., to enhance it).

There are various risks associated with the user study design. We might run into the risk of users not being able to get an idea of what our application will actually look like if we are not able to provide them with the sufficient information that we have defined for ourselves. Because without an application(our Bonus application for Chrome), users will not be able to have the complete experience we wanted them to have.

The last risk lies in the overfitting issue for the training set used for Bullseye algorithm. We use 200 software engineering research papers as we, as authors, will create gold true passages. In future work, we can include other general science papers in the training set and ask students from other domains to create gold true passage sets.

## 4. REFERENCES


[1] Azzah Al-Maskari and Mark Sanderson. A review of factors influencing user satisfaction in information retrieval. *Journal of the American Society for Information Science and Technology*, 61(5):859–868, 2010.

[2] Bruce Croft and Jinxi Xu. *Corpus-specific stemming using word form co-occurence*. Citeseer, 1994.

[3] Hang Cui, Ji-Rong Wen, and Tat-Seng Chua. Hierarchical indexing and flexible element retrieval for structured document. In *Advances in Information Retrieval*, pages 73–87. Springer, 2003.

[4] Joshua C Denny, Anderson Spickard III, Kevin B Johnson, Neeraja B Peterson, Josh F Peterson, and Randolph A Miller. Evaluation of a method to identify and categorize section headers in clinical documents. *Journal of the American Medical Informatics Association*, 16(6):806–815, 2009.

[5] Eibe Frank, Chang Chui, and Ian H Witten. Text categorization using compression models. 2000.

[6] Charles R Hildreth. Accounting for users' inflated assessments of on-line catalogue search performance and usefulness: an experimental study. *Information research*, 6(2):6–2, 2001.

[7] Jaap Kamps, Maarten Marx, Maarten De Rijke, and Börkur Sigurbjörnsson. Xml retrieval: What to retrieve? In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 409–410. ACM, 2003.

[8] Mounia Lalmas and Anastasios Tombros. Evaluating xml retrieval effectiveness at inex. In *ACM SIGIR Forum*, volume 41, pages 40–57. ACM, 2007.

[9] Marc Light, Gideon S Mann, Ellen Riloff, and Eric Breck. Analyses for elucidating current question answering technology. *Natural Language Engineering*, 7(04):325–342, 2001.

[10] PDFBox. `http://pdfbox.apache.org/index.html`.

[11] Amit Singhal, Gerard Salton, Mandar Mitra, and Chris Buckley. Document length normalization. *Information Processing & Management*, 32(5):619–633, 1996.

[12] Stefanie Tellex, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 41–47. ACM, 2003.