A Collaborative Approach to Building Evaluated Web Pages Datasets

Ricardo Barros¹, José A. Rodrigues Nt.¹, Heraldo J. A. Carneiro Filho¹, Fabrício R. S. Ferreira¹, Olivério C. Fernandes¹, Carlos Eduardo P. Silva², André L. G. Ribeiro¹, Geraldo B. Xexéº¹,², Jano M. de Souza¹

¹COPPE, Graduate School of Engineering ²Computer Science Dept. – Institute of Mathematics UFRJ - Federal University of Rio de Janeiro, Rio de Janeiro, Brazil
{rbarros, rneto, heraldo, fabriciorsf, oliverio, kdu, aribeiro, xexo, jano}@cos.ufrj.br

Abstract

In order to evaluate information retrieval algorithms it is imperative to use a dataset as a test database. However, access to such datasets is often difficult and expensive, since building them is a time-consuming and costly task. This paper presents a collaborative approach to dataset creation that uses a data quality evaluation technique based on fuzzy theory, to assist users in selecting suitable web documents for their datasets. These documents are automatically captured by a crawler and assessed on information derived from their metadata.

Keywords: Cooperative Work, Dataset Building, Data Quality, Web Document Metadata, Fuzzy Theory, Information Retrieval.

1. Introduction

Evaluation of most researches in information retrieval is commonly done by calculating precision and recall curves for two or more algorithms on the same set of data. When performing a search, the user wants to find all relevant documents, in a particular collection, for his query [1].

Determining which documents are relevant for a specific query is a task that can only be done by the user himself. Therefore, to calculate measures such as precision and recall for a specific algorithm, it is necessary to have, beforehand, a collection of documents, queries, and associations between documents and queries, indicating which documents are relevant for each query. Having such datasets at hand makes it possible to compare different algorithms precisely and independently.

However, access to these datasets is usually difficult and expensive, since building them is a time-consuming and costly task. This is especially true for large datasets: dozens of queries must be defined, hundreds or thousands of documents must be collected, and, for each query, every single document must be evaluated by a user to determine whether it is relevant in answering that particular query.

We developed FoxSet, a set of tools to help researchers in collecting and evaluating the several documents that will make up these datasets, in a collaborative approach, to facilitate this task. Collaboration is made possible by an extension to the open source Mozilla Firefox web browser [2], through which most user interaction is carried out. Whenever more significant or complex user interaction is required, the extension invokes a web application.

One of the tasks that can be done cooperatively is the collection of web documents. Users can add suitable documents to a particular dataset as they normally browse the web, by clicking a button. The document will then be immediately available for other users to work on. Different roles define what actions each user is allowed to perform in the process of building a dataset. These roles are dataset-specific, not system-wide.

The task of finding appropriate documents for a dataset can also be done semi-automatically, by a backend service. The service is able to crawl the web from a set of starting web documents or through major search engines. Our approach attempts to avoid outdated, imprecise or invalid documents by performing an automatic document evaluation. This evaluation uses a novel data quality assessment technique, to filter documents according to user requirements. The technique is based on deriving different indicators of data quality from web document metadata and combining these indicators using fuzzy theory methods [21].

In our approach, cooperation is also the cornerstone of manual document evaluation. By allowing several users to share the task of evaluating every combination of documents and queries, the process of building datasets is made easier.

2. Background

2.1. Fuzzy theory

By modeling the uncertainties of natural language through concepts of partial truth, i.e. truth-values falling somewhere between absolutely true and absolutely false [3], fuzzy logic is able to manipulate real-world objects that possess imprecise limits. We should regard the process of “fuzzification” as a methodology to generalize any specific theory from a crisp (discrete) to a continuous (fuzzy) form. Just as there is a strong relationship between Boolean logic and set theory, there
is a similar strong relationship between fuzzy logic and fuzzy set theory.

A fuzzy subset A of a set X can be defined as a set of ordered pairs, each with the first element from X and the second element from the interval [0,1], with exactly one ordered pair present for each element of X. This defines a mapping – a membership function – between elements of the set X and values in [0,1]. The set X is referred to as the universe of discourse for the fuzzy subset A.

This can be better understood with an example. Let us talk about people and “age.” In this case, the set X = {10,20,30,40,60,70} is the set of “ages”. Let us define fuzzy subsets “child”, “young”, “adult”, and “old” which will answer the question “to what membership degree is person x “child”, “young”, “adult”, and “old”? Zadeh [4] describes “age” as a linguistic variable, which represents our cognitive category “age group.” To each person in the universe of discourse, we have to assign a degree of membership in the fuzzy subsets. The easiest way to do this is with a membership function based on the person’s age. The fuzzy set A, young, could be described as: A={(10,1), (20,0.8), (30,0.5), (40,0.2), (50,0.1), (60,0), (70,0)}.

The literature offers families of parameterized membership functions such as triangular, exponential, and Gauss functions. Defuzzification is the inverse process – the conversion of a fuzzy set into crisp values [5]. There are many defuzzification methods (at least 30), but the most common techniques are the centroid and maximum methods. The main concepts of our proposal and how these concepts have been extended by fuzzy logic theory will be shown later.

2.2. Quality and information quality

In spite of the existence of different efforts to create a definition of quality, “no single definition or standard of quality exists” [6]. Pipino et al. [7] state that quality is a multidimensional concept, since users must deal with both subjective perceptions of the individuals involved with the data, and objective measurements based on the dataset under evaluation. Previous initiatives, exemplified in [8], tried to define, organize, and prioritize the required information, improving its quality to the end user.

Quality and information quality have attracted the interest of researchers in a large number of disciplines, including management of information systems. A strong commercial interest in the latter exists, with emphasis on the costs and the impact for the organizations in consequence of low data quality [9].

2.3. Quality Dimensions

Despite the frequent use of some terms to indicate data quality, there is not a rigorously defined or standardized set of data quality dimensions. A comprehensive set of dimensions, able to represent users’ quality expectations, such as timeliness, reputation and completeness, can be found in [7]. To carry out information quality evaluation, first we need to identify the set of quality dimensions. The most appropriate set depends on the user application, the selection of metrics, and the implementation of the evaluation algorithms that measure or estimate such quality dimensions [10]. Wand and Wang [11] state that the choice of these dimensions is primarily based on intuitive understanding, industrial experience or literature review. Tillman [12] emphasizes the need to have in mind the current state of the Internet, to adopt generic criteria for information quality evaluation.

2.4. Metadata

In some approaches presented in [8], different metadata can be associated with data, including metadata to improve or restrict quality, according to a set of dimensions [13,14]. As an example of metadata adopted in this work we can refer to Kleinberg’s proposal [15]. He developed a set of algorithms, called HITS (Hyperlink Induced Topic Search), for extracting information from hyperlink structures. He states that annotation on the Internet “almost says something about the way the Web has evolved.” The goal of HITS is to rank pages on the Web through the discovery of related authoritative information sources. HITS introduced two concepts: authorities and hubs. Kleinberg’s method says that the best authorities will be those that point to the best hubs, and the best hubs will be the ones that point to the best authorities. HITS operates on focused subgraphs of the Web that are constructed from the output of a text-based Web search engine, like Google or Alta Vista. From there on, text is ignored, and the application only looks at the way pages in the expanded set are linked to one another. For an assessment of several link analysis algorithms (algorithms that explore hyperlink structure) and how they can help in predicting quality ratings of web documents, refer to [16].

2.5. Collaborative Evaluation

Link analysis algorithms have been successfully used to evaluate information on web pages by exploiting the link structure of the web [16] – created through independent activities of many web page authors. Search engines that employ this approach, such as Google, rely on an implicit rating scheme, where a link to a document is considered a vote for it.

An alternative to this is explicit rating, where a user assigns a rating or a positive (or negative) vote to some document. Explicit ratings are used in many commercial collaborative filtering applications, to recommend new products to users [17], e.g., Amazon and Netflix. This approach has also been proved successful by social news aggregators, like Digg, which became popular in part because they rely on the distributed opinions of many independent voters.

When applying explicit rating, several issues must be considered. For one, a rating scale must be chosen.
Stevens [18] proposed four levels of measurement: nominal, ordinal, interval and ratio. The choice of level of measurement determines what operations and relations are defined for a set of collected ratings, e.g., a mean is only defined for interval and ratio.

Another issue that should be addressed is how differently users perceive the given scale values. The meaning of each scale value must be made clear and unambiguous, to minimize misunderstandings.

If evaluation is to be made by more than one user for a given object, the issue of how to aggregate ratings also arises. When using interval or ratio levels of measurement, it is common to calculate means by averaging ratings. This, however, does not bear in mind that the opinion of a particular user might be more important than of another one, since the former might be more knowledgeable than the latter on the specific subject matter at hand.

Approaches to distinguish among users opinions are usually based on reputation (the user has a reputation reflecting his prior collaborations or his knowledge as viewed by his peers) or self-evaluation (the user rates his expertise on the subject matter). When using means to aggregate ratings, reputation or self-evaluation can be taken into account by employing a weighted average.

Finally, if collaborative evaluation is to be done on a public or uncontrolled environment – i.e., where users are not known in advance and might act as they please – the issue of malicious user behavior is of great importance. In such environments, it is common to have users trying to game the system by intentionally providing false or erroneous information.

3. Our approach

As we stated before, our approach relies on collaboration among users and automatic quality evaluation of web documents, to speed up and ease the process of building datasets. It is composed of six core processes, performed by users with distinct roles.

3.1. Permissions and Roles

Permissions are system-wide, but roles are dataset-specific. Users are called collaborators, since we use the term “user” for a dataset role. To avoid confusion, we will refer to people who directly interact with our system as collaborators from now on.

We only have two permissions: administrator rights and coordinator rights. Administrator rights give a collaborator the ability to create, modify and remove collaborators’ profiles. Coordinator rights give a collaborator the ability to design a dataset, i.e., to initiate the process of building a dataset. On a dataset, we have three roles: coordinator, evaluator and user. Coordinators are responsible for the managerial activities of building datasets, such as creation, role assignment and finalization. They are also the only ones who can perform tasks such as choosing the context of the dataset, defining how automatic evaluation will be done, selecting an evaluation scale, defining queries, and removing unsuitable web documents from the collection.

Evaluators are collaborators who will be cooperatively collecting web documents (if automatic collection and evaluation is not being used) and, afterwards, evaluating those documents.

Finally, users are collaborators who are interested in using a particular finalized dataset. They are able to create different subsets of a dataset, based on some criteria, and can export these subsets in different formats, such as XML.

3.2. Process

The six processes we use in our approach are dataset creation, dataset feeding, automatic evaluation, manual evaluation, dataset finalization and dataset usage. The whole dataset lifecycle is depicted in Figure 1.

3.2.1. Dataset creation. The coordinator is responsible for this process. In order to create a dataset, it is necessary to define its context. The context can be seen as the subject of the dataset. The coordinator must decide if the dataset will be manually populated or automatically populated, using a crawler. He must choose which quality dimensions the backend service will evaluate in that process. In our first version, we contemplate the following quality dimensions: completeness, reputation and timeliness. After selecting a subset of the available quality dimensions, the coordinator assigns an importance degree to each quality dimension. We adopted a scale of importance degrees from 0 (no importance), to 4 (essential).

The definition of queries for the dataset is also included in this process, but in our implementation it can be done at any time between dataset creation and manual evaluation.

![Figure 1. Dataset lifecycle with processes and roles](image-url)

3.2.2. Dataset feeding. This process can be done manually by evaluators or automatically by the FoxSet crawler. The crawler was adapted from [20], to retrieve web document (and their metadata) related to the dataset context.

Metadata extracted from each web document are forward links, backward links and last modification date. They are used to derive information used to
evaluate web documents automatically. **Error! Not a valid bookmark self-reference.** shows the original metadata and derivation functions.

### Table 1. Original and derived metadata

<table>
<thead>
<tr>
<th>Original Metadata</th>
<th>Functions to obtain Derived Metadata (fD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$te_i$ – update date of a Web document $i$</td>
<td>(1) $(UT) = ut_i = t_e - at_i$</td>
</tr>
<tr>
<td>$qt_i$ – query date of a Web document $i$</td>
<td>(2) $Authority = c_i - \sum h_j$, where: $j \in BL_i$</td>
</tr>
<tr>
<td>$BL_i$ – number of links which points to the Web document $i$ on a context</td>
<td>(3) $Hub = h_i = \sum c_i j$, where: $j \in FL_i$</td>
</tr>
<tr>
<td>$FL_i$ – number of links going out of the Web document $i$ on a context</td>
<td>The calculation of Authorities and hubs considers a set $S$ of documents on a context. It is an iterative process where all of weights initialize on 1. Afterwards, hubs and authorities weights are calculated and the results are normalized. This process is repeated until the convergence of values $a$ and $h$ of all documents. We adopt JUNG to obtain the values of hubs and authorities.</td>
</tr>
</tbody>
</table>

#### 3.2.3. Automatic evaluation.

The objective of this process is to provide an automatic evaluation of each web document, based on the quality aspects that are important to the dataset context. This will help the coordinator in selecting which documents ought to belong to the dataset. It is done using the quality dimensions described previously, and is made up of six steps, which were originally conceived in [21]: metadata derivation, fuzzification, definition of SQER (Single Quality Evaluation Results), definition of CQD, calculation of CQER (Composed Quality Evaluation Results) and defuzzification.

Metadata derivation consists of deriving the original metadata into a number that can be manipulated in the fuzzification step. For timeliness, we extract the last modification date of the web document and we subtract it from the current date. For completeness and reputation, two scores are calculated, hub and authority, as defined in Table 1.

Fuzzification is better shown by means of an example. Figure 2 shows how the membership functions map the update time into membership degrees of the timeliness linguistic variable, regarding each linguistic term.

Our implementation uses triangular membership functions to distribute the domain values in each linguistic term, as shown in Figure 3.

**Figure 3. Membership functions for linguistic terms**

The definition of CQD step consists of forming a context weight vector, composed of the importance degrees assigned by the coordinator to each quality dimension in the dataset context. This vector is defined by the formula $CQD = \langle w(c,qd_1), w(c,qd_2), ..., w(c,qd_n) \rangle$, where $w$ is the weight of quality dimension $qd_i$ in context $c$.

The definition of SQER step consists in forming a matrix which lines represent linguistic terms and columns represent membership degrees of linguistic variables, for each linguistic term. Table 2 demonstrates one such matrix.

**Table 2. Example of SQER matrix**

<table>
<thead>
<tr>
<th>SQER</th>
<th>Timeliness</th>
<th>Completeness</th>
<th>Reputation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poor</td>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>Bad</td>
<td>0.2</td>
<td>0.6</td>
<td>0.3</td>
</tr>
<tr>
<td>Regular</td>
<td>0.5</td>
<td>1</td>
<td>0.4</td>
</tr>
<tr>
<td>Good</td>
<td>0.8</td>
<td>0.6</td>
<td>0.5</td>
</tr>
<tr>
<td>Excellent</td>
<td>1</td>
<td>0.3</td>
<td>0.6</td>
</tr>
</tbody>
</table>

The definition of CQER step combines the CQD vector and the SQER matrix and calculates composed results to each web document. This is done by the following formula:

$$CQER_j = \frac{\sum_{i=1}^{n} (cqdi \times SQER_{ij})}{\sum_{i=1}^{n} cqdi}$$

$SQER$ represents the input fuzzy sets and $CQER$ represents the output fuzzy sets, where $cqdi$ is the importance degree of each $qd_i$ in context $c$; $j$ is a specific linguistic term and $i$ is a linguistic variable.

The defuzzification step provides a single score for the membership degrees of each linguistic term of each linguistic variable. A final score for a document, as determined by the centroid method, can be seen in Figure 4.
3.2.4. Manual evaluation. The process of manual evaluation is done cooperatively by the coordinator and evaluators. It starts with the coordinator reviewing the collected web documents. If automatic evaluation was performed, the coordinator is able to filter documents by their scores.

After that, collaborative evaluation starts, with each evaluator receiving a set of web documents and a set of queries, and evaluating each document with respect to each query.

The definition of the evaluation scale in our approach is different from the usual definition of rating scales. We want two types of information: a binary variable (“yes” or “no”) and a nominal variable (which might be composed of an arbitrary number of scale values defined by the coordinator, such as “does not answer”, “partially answers”, “fully answers”), both indicating whether a document is relevant or not for a given query. To minimize redundant work by the evaluators, we made it the job of the coordinator, to decide which scale values indicate that a document is relevant for a particular query. Thus, while one coordinator might say that “partially answers” and “fully answers” both indicate that a document is relevant, another might only accept “fully answers” ratings as indicative of document relevancy.

Although our approach provides freedom in defining the evaluation scale, we enforce that any scale defined by the coordinator uses the ordinal level of measurement, i.e. can always be ordered, which allows us to calculate median ratings, but not mean ratings. Since each combination of document and query can be evaluated by more than one evaluator (this is a dataset-specific setting, chosen by the coordinator), our collaborative evaluation uses medians to aggregate these different ratings.

However, we do not address the previously mentioned issue of distinguishing among different opinions. We also do not provide any means to identify, nor to avoid, malicious user behavior. For now, we only envision FoxSet being used in controlled environments (such as research teams). As FoxSet matures, the need for such mechanisms will surely arise.

3.2.5. Dataset finalization. The process of dataset finalization is done by the coordinator when all queries have been defined and all documents have been evaluated for each query. Once finalized, the dataset cannot be modified.

3.2.6. Dataset usage. In this process, the user is able to refine the original dataset by removing queries or documents, or by filtering them based on some criteria.

4. Results

Since the first complete experiment is not yet concluded, we used the tool to create a case study, demonstrating FoxSet use.

First, the coordinator set the context to be “economy”, selected three quality dimensions, and assigned to each of them an importance degree – 3 for timeliness, 4 for reputation and 2 for completeness. He decided for automatic collection and provided five web documents as seeds. After that, he defined the evaluation scale to be “No info”, “Mentions”, “Partially explains” and “Fully explains”, selecting the last two as indicating document relevancy. Finally, he defined query Q1 as “What are the impacts of the current financial crisis?” Upon dataset creation, the backend service started to crawl the web. When the number of documents reached the limit defined, in this case 500, automatic evaluation begun. The resulting scores ranged from 0.01244 to 0.43126.

<table>
<thead>
<tr>
<th>ID</th>
<th>Score</th>
<th>Document URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>0.43126</td>
<td><a href="http://www.economywatch.com/">http://www.economywatch.com/</a></td>
</tr>
<tr>
<td>D2</td>
<td>0.32707</td>
<td><a href="http://www.brazzil.com/">http://www.brazzil.com/</a></td>
</tr>
<tr>
<td>D3</td>
<td>0.32507</td>
<td><a href="http://www.scps.nyu.edu/areas-of-study/global-affairs/">http://www.scps.nyu.edu/areas-of-study/global-affairs/</a></td>
</tr>
<tr>
<td>D4</td>
<td>0.32469</td>
<td><a href="http://www.hindustantimes.com/">http://www.hindustantimes.com/</a></td>
</tr>
<tr>
<td>D5</td>
<td>0.32466</td>
<td><a href="http://www.csmonitor.com/world/">http://www.csmonitor.com/world/</a></td>
</tr>
</tbody>
</table>

When manual evaluation initiated, the coordinator reviewed the collected documents and filtered documents with scores of 0.324 or less, to keep only the five top-scoring documents, shown in Table 3.

During collaborative evaluation, the evaluators rated these documents, for our only query Q1.

Table 4 reports on how many evaluators assigned each scale value to each document in respect to whether it answered query Q1 or not. It also shows the median ratings for each document.

<table>
<thead>
<tr>
<th>Scale value</th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
<th>D5</th>
</tr>
</thead>
<tbody>
<tr>
<td>No info</td>
<td>0</td>
<td>4</td>
<td>10</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Mentions</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Partially explains</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>Fully explains</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

After manual evaluation, the coordinator made final adjustments to the dataset and finalized it. A resulting subset from the above example is shown in Figure 5.
Figure 5. A resulting subset of the example dataset

At this point, we are unable to attest to what extent our automatic quality evaluation reduces the subsequent manual evaluation workload. A comparative analysis with 20 students is ongoing at the time of this writing.

5. Conclusion

In this paper, we presented a collaborative approach to building evaluated datasets that relies on fuzzy theory, data quality, metadata and collaborative evaluation concepts and techniques, to speed up and ease the burdensome process of building a dataset. Some aspects of the problems we have tackled in this paper have been deliberately simplified for ease of understanding. Nevertheless, we hope that our work is able to assist researchers, especially those in the field of information retrieval, in assessing their research.

Our research efforts are now aimed at conducting experiments to measure how much our automatic quality evaluation reduces evaluation workload.

For future work, we intend to investigate and implement self-evaluation and reputation mechanisms, as in [22]. This will allow for a better balance among evaluators of different skill levels and help detecting and preventing malicious user behavior in collaborative document evaluation.

Acknowledgement

This work was supported by CAPES, CNPq and FAPERJ.

References
