Sofia Search: A Tool for Automating Related-Work Search

Behzad Golshan
Boston University
behzad@cs.bu.edu

Theodoros Lappas
Boston University
tlappas@bu.edu

Evimaria Terzi
Boston University
evimaria@cs.bu.edu

ABSTRACT

When working on a new project, researchers need to devote a significant amount of time and effort to surveying the relevant literature. This is required in order to gain expertise, evaluate the significance of their work and gain useful insights about a particular scientific domain. While necessary, relevant-work search is also a time-consuming and arduous process, requiring the continuous participation of the user. In this work, we introduce Sofia Search, a tool that fully automates the search and retrieval of the literature related to a topic. Given a seed of papers submitted by the user, Sofia Search searches the Web for candidate related papers, evaluates their relevance to the seed and downloads them for the user. The tool also provides modules for the evaluation and ranking of authors and papers, in the context of the retrieved papers. In the demo, we will demonstrate the functionality of our tool, by allowing users to use it via a simple and intuitive interface.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Experimentation

Keywords

related-work search

1. INTRODUCTION

Searching for *related work* is an integral part of the research process. By identifying published papers that are relevant to their projects, researchers gather information on findings that have been previously presented in their field of interest. Such findings can provide motivation, insights and understanding of particular scientific field. Even though the Web and the various search engines have greatly facilitated the task of searching related work, it still remains a time-consuming and arduous process, which cannot be done automatically; it requires the continuous participation of the user. In this demo, we will refer to the task of searching related research literature as *related-work search.*

In the typical related-work search scenario, the user starts with a seed of one or more papers, which he knows to be relevant to his own project. She then looks for candidates by reading the available text and the cited references, following the citation graph, and submitting queries to search-engines and paper-repositories like Google Scholar, CiteSeer and DBLP. For each candidate paper, the user then needs to locate an openly available copy, download it, and manually verify its relevance by reading it. This is a recursive task, which terminates when the user is confident that all relevant items have been retrieved.

Our system, Sofia Search, aims to automate this process, thus equipping researchers with a valuable tool for handling a crucial and time-consuming task. At a high level, Sofia Search emulates the steps followed by the human user. Algorithm 1 presents these steps in the form of a pseudocode.

Algorithm 1 `Related-Work Search`

```
Input: Seed of papers $\mathcal{S}$, Lower bound $\ell$
Output: Set of relevant papers $\mathcal{R}$
$\mathcal{R} \leftarrow \emptyset$
for each paper $P \in \mathcal{S}$ do
    processPaper($P, \ell, \mathcal{S}, \mathcal{R}$)
return $\mathcal{R}$

Procedure processPaper()
Input: Paper $P$, Seed $\mathcal{S}$, Lower Bound $\ell$, $\mathcal{R}$
$\mathcal{C} \leftarrow inlinks(P) \cup outlinks(P)$
for each paper $P' \in \mathcal{C}$ do
    if Rel($P', \mathcal{S}$) $\geq \ell$ then
        $\mathcal{R} \leftarrow \mathcal{R} \cup \{P'\}$
        processPaper($P', \mathcal{S}, \ell, \mathcal{R}$)
```

The input to our system consists of a set of seed papers $\mathcal{S}$ that we know to be relevant to a particular field of study. Along with the papers, the user also inputs a lower bound ℓ on the *relevance* of every candidate related paper to the seed. Each paper in the seed is processed by the `processPaper()` routine. Given a seed paper P, the routine retrieves the set of papers that P cites (outlinks) and the set of papers that cite P (inlinks). These two sets compose the collection of candidates $\mathcal{C}$ to be considered. The relevance of each candidate to the seed is evaluated via the `Rel`() function; the higher the value of relevance to the seed, the more relevant a candidate is to the seed. If the relevance of candidate P' satisfies the lower bound, P' is added to the final result, and recursively processed by the routine. Clearly, the effectiveness of the the algorithm depends heavily on the way that function `Rel`() is evaluated. We discuss the details of this function in the next section.

While Sofia Search was developed to automate the related-work search, we extend its functionality to to include the following two natural applications:

- **Paper Ranking:** The set of retrieved papers $\mathcal{R}$ defines the relevant literature in the domain defined by the seed $\mathcal{S}$. However, not all reported papers in $\mathcal{R}$ are equally influential within this domain. In order to capture this intuition, we extend Sofia Search with a ranking function that ranks the papers in $\mathcal{R}$ based on their influence within this domain.
- **Author Ranking:** As for the papers, Sofia Search also ranks the authors of the papers in $\mathcal{R}$ in decreasing order of their influence. In this case, we assume that influential authors are those that have written influential papers. Therefore, the ranking of the papers also implies the ranking of the authors.

2. Sofia Search

In this section, we give a detail description of how the functionalities we introduced in the previous section have been implemented in Sofia Search. First, we identify and describe the components of the `ProcessPaper()` function shown in Algorithm 1. We then discuss how our tool incorporates paper- and author-ranking functionalities.

2.1 Related-work search

As shown in Algorithm 1, in order to process a candidate paper P two functionalities are required: **(a)** the evaluation of the relevance of P to the provided seed $\mathcal{S}$, and **(b)** the retrieval of the papers that are referenced by P' (outlinks), as well as those that reference P (inlinks).

2.1.1 Evaluation of relevance

The relevance of a paper P to a given seed of papers $\mathcal{S}$ is evaluated via the `Rel`() function. Intuitively, the relevance of P to $\mathcal{S}$, is defined by the pairwise relevance between P and any of the papers in $\mathcal{S}$.

Thus, we first need to define a method that evaluates the relevance between two papers P and P'. We do that by evaluating how related the two papers are accross a set of different dimensions; we call these dimensions *factors* and we use F to denote the total set of factors we consider. For each factor f, we implement a function that evaluates the relevance between P and P' with respect to f, denoted by $\texttt{Rel}(P, P' \mid f)$. Then, the relevance of paper P to seed $\mathcal{S}$ is defined as follows:

$$\texttt{Rel}(P, \mathcal{S}) = \max_{P' \in \mathcal{S}} \sum_{f \in F} w(f)\texttt{Rel}(P, P' \mid f). \tag{1}$$

Observe that the final relevance value is a linear combination of the the relevance scores computed for each of these factors. The weight $w(f)$ of each factor f in the linear function is learned via training on set of papers for which related papers have been manually identified.

The use of max in Equation (1) can be replaced by the average, median or any other appropriate aggregate function. Further, while this simple linear combination yields great results, Sofia Search can be extended to handle other combination functions. Next, we discuss the factors that we consider to form set F.

Content similarity: Intuitively, a high value of content similarity between two papers means that the two papers discuss similar (or the same) topics, and are thus closely related. We implement two alternative measures for content similarity: **(a)** cosine similarity under the popular vector-space model [2] and **(b)** topic-based similarity, as based on the topic distributions learned via LDA [3].

In the vector space model, a paper P is represented by vector $\vec{p}$ of numeric attributes. Each dimension in $\vec{p}$ corresponds to a term: if term t does not occur in P, then $\vec{p}(t) = 0$. Otherwise, $\vec{p}(t)$ takes a non-zero value that captures the importance of term t in the paper. The most popular measure of importance of terms in documents is *TF-IDF*. TF-IDF which considers the frequency of the term inside the given paper, as well as its frequency in the entire collection of available papers. Formally, given a corpus of papers $\mathcal{D}$ and a term t, the TF-IDF of t with respect to a particular paper $P \in \mathcal{D}$ is computed as follows:

$$\text{TF-IDF}(t, P, \mathcal{D}) = \mathit{tf}(t, P) \times \log \frac{|\mathcal{D}|}{\{P' : t \in P'\}} \tag{2}$$

The term *tf(t,P)* returns the frequency of t in P. We use an extended corpus of papers to compute the can be used to compute the global frequency factor (i.e. the second factor in the equation). Using TF-IDF, we generate a a numeric vector for any given paper. Vectors $\vec{p}$ and $\vec{p}'$ that correspond to papers P and P' can then be compared using the vector cosine similarity [2] as follows:

$$\text{C-SIM}(\vec{p}, \vec{p}') = \frac{\vec{p} \cdot \vec{p}'}{||\vec{p}|| \times ||\vec{p}'||}. \tag{3}$$

In Sofia Search, we have also implemented an alternative measure of similarity between two papers, based on Latent Dirichlet Allocation (LDA) [3]. In this case, every paper is expressed as a mixutre of a fixed number of topics. Given a training corpus and a specified number of topics, LDA learns these distributions for each paper. We refer the reader to the original paper for more information on the technique [3]. Again we use a large corpus of papers is used to learn the topic models offline. Then, we express new documents as mixtures of the extracted topics; for this purpose, we use the functionality provided for the MALLET toolkit [6]. We then compare two documents by comparing their components with respect to their distribution of topics.

Note that both for cosine similarity as well as the LDA method, we use training corpora that are from same general domain as the papers to be evaluated (e.g. “medicine” or “computer science”).

Frequency and placement: This factor takes into account two papers P and P', such that P cites P'. Then the factor assumes that the relevance (or importance) of P' to P is encoded in: **(a)** the number of times P' is explicitely referenced in the text of P, and **(b)** the placement of these references i.e., the actual sections within P where they occur. For **(a)**, clearly an important reference is referenced repeatedly in the text. For **(b)**, the correlation between the section P' is referenced and its importance depends on the scientific domain. In our current implementation, we have assumed that the most important references appear in the "abstract" or in the actual technical sections of a paper, while the least important ones appear in the "related-work section"; afterall, the related-work section often contains papers only remotely related papers. For paper P that cites P' and for f =*(frequency and placement)* we combine **(a)** and **(b)** in the following relevance measure

$$\texttt{Rel}(P, P' \mid f) = \sum_{r \in \text{Refs}(P,P')} W(S_r), \tag{4}$$

where Refs(P, P') returns the references of P' that appear in P and for every reference r, S_r is the section in which this reference appears. Finally, $W(S_r)$ is a weighted scheme that assigns different importance to different sections, following the intuition we described above.

Coauthorship similarity: The intuition behind this method is that researchers are more likely to read and be influenced by papers written by people they collaborate with, i.e., their own co-authors. In order to capture this intuition in a factor we proceed as follows: For every paper P, we extract its set of authors $A(P)$. For each author in $A(P)$, we query DBLP and extract his/her most-frequent co-authors. Taking the union of $A(P)$ with these latter sets of authors, we construct the extended neighborhood of authors $N(P)$ for paper P. Given two papers P and P' we consider their relevance with respect to coauthorship similarity, by computing the Jaccard coefficient [2] of sets $N(P)$ and $N(P')$. That is, for f =*(coauthorship similarity)*, we have that

$$\texttt{Rel}(P, P' \mid f) = \frac{|N(P) \cap N(P')|}{|N(P) \cup N(P')|}. \tag{5}$$

Temporal distance: The relevance between two papers can be partially encoded in their distance on the temporal axis. The assumption made by this measure is that a paper published in 2011 is more likely to be strongly related (e.g., in terms of motivation and problem settings) to a contemporary paper, rather than a paper written in the 1980s. The distance or similarity between two papers is this encoded via the difference (in years) between their publication dates.

Citation neighborhood similarity: This factor quantifies the relevance between two papers P and P' by comparing the set of papers that are cited by them or cite these papers. More specifically, the citation neighborhood of P, denoted by $C(P)$, is extraced by retrieving the papers that cite and are cited by P and recursively by the references of P, where the recursion proceeds until a pre-specified depth. Given the citation neighborhoods of P and P' we quantify the relevance of P and P' with respect to this factor by computing the Jaccard coefficient between $C(P)$ and $C(P')$.

Number of citations: This factor does not consider the relevance between a pairs of papers. Instead, it evaluates the general impact of a paper via the number of the citations it has accumulated. We include this factor to boost popular papers. An issue that emerges here is that older papers have more time to accumulate citations. To account for this, we evaluate papers based on the average number of citations that they have accumulated per year, following their publication year.

2.1.2 Paper identification and retrieval

Apart from computing the relevance between papers, Sofia Search also implements modules for the retrieval of the all the papers that cite and are cited by a given paper P. For papers cited by P, our tool parses the given file (typically in pdf format) and extracts the titles of the references. This is done via a rule-based parsing mechanism that utilizes regular expressions. For papers that cite P, we submit a query to Google Scholar, which maintains the titles of the papers that cite P.

After retrieving the titles of all papers that cite P, Sofia Search next retrieves the actual (usually .pdf) files. While numerous versions of a paper may exist on the web, not all of them are openly available. Therefore, Sofia Search checks various potential sources. This is done by automatically submitting a query Google Scholar, which maintains a list of different web locations for each paper (i.e. "versions"). If the list is exhausted without finding a usable link, we submit a query to DBLP, which also keeps a link for every paper in the DBLP database.

2.2 Paper and author ranking

In addition to locating and retrieving the set of relevant papers, Sofia Search also provides functionalities for the ranking and visualization of top authors and papers, in the context defined by the set of retrieved papers. Initially, papers are ranked based on their individual relevance scores to the seed, as defined in Equation (1). Then, the score of each author is computed as the the sum of the relevance scores of the papers that have been retrieved and he has co-authored. The ranking of papers can be used to recommend to the user the order in which to the user the order in which he should process (i.e., read) the retrieved papers. Similarly, author-ranking provides information on who are the authorities on the topic captured in the seed.

3. THE Sofia Search INTERFACE

In this section, we discuss the interface of Sofia Search and describe how users can interact with our system and utilize its functionalities. We have designed the Sofia Search interface so that it is simple and intuitive. A screenshot can be seen in Figure 1.

The "Input" tab – placed at the top part of the interface – is used to input the seed of papers to be considered. The user has three different ways to specify the papers of interest. The first way is to type a link (url) to an online version of the paper in the textbox in the top. Given this url Sofia Search automatically downloads the specified paper. The second way is to simply type the title of the paper in the textbox. Our tool then searches different online repositories (e.g., CiteSeer, Google Scholar, DBLP) in order to locate and download the paper. Finally, if the user already has a local copy of the paper, he can load it by clicking the "Browse" button and locating the file on the disk. After a paper has been specified by using any of the above three options, the user clicks the "Add" button to verify the insertion of the file to the seed. The list of added files is shown in the text area below. In the example shown in the figure, the user has added the highly cited paper "Fast Algorithms for Mining Association Rules", by Rakesh Agrawal and Ramakrishnan Srikant. The paper was loaded by typing a link to an online version.

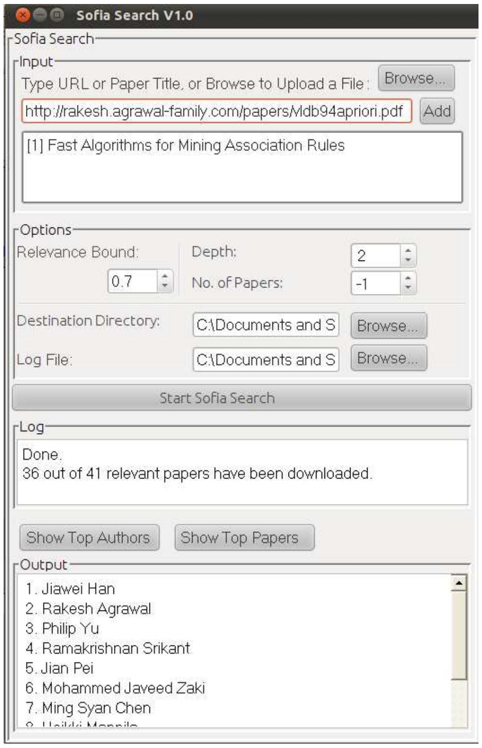
The "Options" tab is used to specify the parameters required as part of Sofia Search's input. The user can input four parameters, which regulate the size of the retrieved set of relevant papers. These parameters are: the "Relevance Bound", the "Depth Parameter", the "Destination Directory" and the "Log File". The "Relevance Bound" parameter represents the minimum requirement of relevance (Equation (1)) that is required in order for the paper to be included in the output. The value of this parameter is in $[0, 1]$; the higher the value, the more difficult it becomes for a paper to be included in the reported related papers. The "Depth Parameter" represents the maximum allowed hop distance between a candidate paper and any paper in the seed. The hop distance is computed on the citation graph. For example, a candidate that directly cites or is cited by a seed paper has a hop distance of 1. Finally, the number of papers is simply the maximum number of papers to retrieve. The user can specify any combination of these three parameters. The "Destination Directory" parameter represents the local folder that will store the downloaded papers. Finally, the "Log File" parameter points to a file that logs the various actions of Sofia Search.

After specifying the desired parameters, the user can click on the "Start Sofia Search" button to initiate the search and retrieval process. The progress of the tool is recored in the "Log" tab, which mirrors the output written to the specified log file. In the example shown in Figure 1, the process has been completed. The message in the Log area states that 36 out of 41 relevant papers were downloaded. The missing papers are due to the inability to locate a freely downloadable version. These are recorded in the log, which the user can consult if he wants to manually retrieve these papers.

The final tab, placed at the bottom of the interface, is the "Output" tab. This is used to display the top authors and top papers in the context of the retrieved relevant literatures. Authors and papers are ranked as described in Section 2.2. The user can toggle between the author and paper rankings by pressing the respective buttons on the top of the tab.

4. RELATED SYSTEMS

A number of tools have been built to assist researchers and students to effectively search for relevant literature. Google Scholar [5] Microsoft Academic Search [7], ArnetMiner [1] and DBLife [4] are some of the existing websites which provide detailed information on a paper, such as the year of publication, the number of citations and links to downloadable versions of the paper. Some of these sites also provide author information, such as the number of publications, the h-index [1], and the total number of received citations.



Type URL or Paper Title, or Browse to Upload a File : http://rakesh.agrawal-family.com/papers/vldb94apriori.pdf

[1] Fast Algorithms for Mining Association Rules

Relevance Bound: 0.7

Depth: 2

No. of Papers: -1

Destination Directory: C:\Documents and S

Log File: C:\Documents and S

Done.
36 out of 41 relevant papers have been downloaded.

1. Jiawei Han
2. Rakesh Agrawal
3. Philip Yu
4. Ramakrishnan Srikant
5. Jian Pei
6. Mohammed Javeed Zaki
7. Ming Syan Chen
8. Heikki Mannila

Figure 1: A Screenshot of Sofia Search.

The focus of these websites is on hosting and providing information on papers and authors. Users interact with these websites by manually submitting queries and evaluating the produced output. While all existing sites provide useful and relevant functionality, the motivation and contribution of Sofia Search is different: our tool provides an end-to-end, fully automated solution which, given a seed of papers, identifies the relevant previous works, downloads them and makes them available to the user. The influence of authors and papers is then evaluated within the context of this particular search and not in terms of their global contribution to science.

Acknowledgments

This work was supported by NSF grant #1017529 and gifts from Microsoft and Yahoo!

5. REFERENCES

[1] Arnetminer. `http://arnetminer.org/`.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press Books, 2011.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.

[4] Dblife. `http://dblife.cs.wisc.edu/`.

[5] Google scholar. `http://scholar.google.com/`.

[6] Mallet. `http://mallet.cs.umass.edu/`.

[7] Microsoft academic search. `http://academic.research.microsoft.com/`.

[1] http://en.wikipedia.org/wiki/H-index