Surrogator: Enriching a Digital Library with Open Access Surrogate Resources

T. Y. S. S. Santosh
Indian Institute of Technology Kharagpur
West Bengal – 721302, India
santoshtyss@gmail.com

Debarshi Kumar Sanyal
Indian Institute of Technology Kharagpur
West Bengal – 721302, India
debarshisanyal@gmail.com

Plaban Kumar Bhowmick
Indian Institute of Technology Kharagpur
West Bengal – 721302, India
plaban@cet.iitkgp.ernet.in

ABSTRACT

Large digital libraries often index articles without curating their digital copies in their own repositories. Consider, for example, the National Digital Library of India (NDLI) which is a huge multi-disciplinary library acting as a single point of entry into a wide spectrum of digital repositories, national and international. Although NDLI allows free full text view for many of its articles, some – especially research publications – are hosted in libraries that impose access tolls. In these cases, NDLI displays the metadata and points to the repository where the full article is available. Similar situation occurs in many other indexing sites like the ACM Digital Library and Scopus. However, authors often keep a free version of their publications in their own institutional home pages or in preprint servers. It also happens sometimes that a conference paper behind a paywall has a closely resembling journal version freely available on the Web. These open access surrogates are valuable to researchers who cannot access the original publications. We design a tool called Surrogator to automatically identify open access surrogates of access-restricted scholarly articles present in a digital library. In this demo, we demonstrate the working of Surrogator on articles from Google Scholar and NDLI.

CCS CONCEPTS

• **Information systems** → *Search interfaces*; *Near-duplicate and plagiarism detection*; • **Applied computing** → **Digital libraries and archives**;

KEYWORDS

open access; surrogate; digital library; academic search engine

ACM Reference Format:
T. Y. S. S. Santosh, Debarshi Kumar Sanyal, and Plaban Kumar Bhowmick. 2018. Surrogator: Enriching a Digital Library with Open Access Surrogate Resources. In *Proceedings of ACM India Joint International Conference on Data Sciences and Management of Data, Demo Track (CoDS-COMAD'18).* ACM, New York, NY, USA, 5 pages. https://doi.org/XXX

1 INTRODUCTION

It is common to find a digital library that indexes articles without archiving their digital copies in its own repository. Examples include the ACM Digital Library[1], Scopus[2] and the newly developed National Digital Library of India (NDLI) [3]. Although NDLI allows free full text view for many of its articles, some – especially research publications – allow only restricted access typically because they are hosted in repositories that require access toll like subscription or pay-per-article or pay-per-view. In these cases, NDLI displays the metadata and points to the source where the full article is available. This creates a major hindrance to researchers who do not subscribe to expensive digital libraries. Such a situation is common in developing countries like India. Scientists traveling or otherwise outside their campus network often face similar hurdles. Not surprisingly, it has been observed that open access (OA) articles receive more downloads and citations vis-à-vis access-restricted ones [3]. By OA, we mean the full article is free to read and download. A silver lining, however, exists even when a publication is access-restricted: many authors cache a free version of their publications in their own institutional home pages or in preprint servers. It also happens sometimes that a conference paper archived behind a paywall has an extended journal version which is freely available on the Web. Given a scholarly article, we define its *surrogate* intuitively as an article that closely resembles it in content and is written by the same authors or the same group. These surrogates may not be identical to the actual publications but prove helpful to researchers who cannot afford the published titles.

This paper describes a tool for detecting OA surrogates for access-restricted scholarly publications in a digital library. In the rest of the paper, we will mean *OA surrogate* whenever we mention surrogate. Readers may argue that a search in an academic search engine like Google Scholar[4] (hereafter, called Scholar) can lead one to a surrogate, if any. Indeed, Scholar can find OA versions if they are available on the Web but it fails when there are mismatches between the titles or authors of a paper and its surrogate. Moreover, a search engine churns out hundreds of pages based on a keyword query and cannot understand if a user is looking for a surrogate. The reader has to manually separate the needle from the haystack. Our objective in this paper is to design a very lightweight tool to automate the task of surrogate identification.

Contribution: We design a simple graphical interface-based tool that can identify surrogates of access-restricted research publications. Since it can intelligently find a close match even when an OA replica of the original article is not available, it may prove more useful than conventional search engines to the academic community. Code and latest updates are available at `https://github.com/dksanyal/Surrogator`.

[1] `https://dl.acm.org/`
[2] `https://www.scopus.com/search/form.uri?display=basic`
[3] `https://ndl.iitkgp.ac.in/`
[4] `https://scholar.google.co.in/`

CoDS-COMAD'18, 11-13th January, 2018, Goa University, India
2018. ACM ISBN XXX.
https://doi.org/XXX

2 MOTIVATION AND PRIOR ART

Nowadays academic search engines like Scholar, Microsoft Academic[5], CiteSeerX[6], Semantic Scholar[7] and applications like Open Access Button[8] (that expect a more precise query like citation or DOI) are quite popular given the surge in research publications [10]. When queried with keywords or exact publication metadata, they discover articles from various sources on the Web including the publishers' sites, researchers' institutional homepage, repositories like arXiv [9] and academic social networks like ResearchGate[10]. Some of them might be OA. This makes them very useful to academicians especially early-career researchers [4], [5], [2].

A serious problem occurs when a publication being searched for does not have an OA copy on the Web which the user can retrieve. Academic search engines generally allow the user to input keywords (or facets) rather than the exact citation and thus, get a list of results matching the keywords. She can scan through them and select the article(s) closest to the one being searched. However, current applications do not consciously attempt to find *approximate* matches when they hit a paywall. We believe there are two main difficulties in doing so. First, defining an approximate match is itself difficult and there is substantial risk in misguiding the user. For example, many journals insist on at least 60% new content when inviting an extension of a conference paper. Thus, although the basic ideas of the conference paper are contained within the journal version, it is difficult to detect it automatically. Similarly, it is not trivial to identify an OA PhD dissertation that has integrated a number of publications by the PhD candidate (that are, however, copyrighted by the publishers and are access-restricted). Second, the issue of paywalls is not extremely serious in the West where publishers endeavor hard to procure subscription. Therefore, the problem remains largely unaddressed. We attempt to define the problem more precisely and take steps towards solving it at least partially.

3 WHAT IS A SURROGATE?

We give a functional definition here. Given an article d, we define its *surrogate* article d_s as an article that is open access and satisfies one of the following properties. The properties are sorted from exact match to most inexact match, in the sense that higher the position of a property that d_s satisfies, better is the expected *quality* of the surrogate.

1. d and d_s are the same articles.
2. d and d_s have the same authors, title and content but might be present in different locations.
3. d and d_s have the same authors and slight variations in title and/or content.
4. d and d_s have slight variations in authors and same title and content.
5. d and d_s have slight variations in authors and slight variations in title and/or content.

We intentionally keep the definition of *slight variation* open. It may be decided in an implementation depending on user requirements. It may also vary with instances; for example, some document pairs manually adjudged to be surrogates might show more variations in author and title than others. Moreover, it might be attribute dependent. For example, intuitively we would like far smaller variations in author list than in title or abstract. Our experiences show that Scholar uses criteria 1 and 2 when tagging an article with its freely downloadable copy. Criterion 5 is the most generic.

4 PROPOSED ALGORITHM

We describe our proposed method for surrogate identification in Algorithm 1. The algorithm takes as input a citation c (in any suitable format like MLA, Chicago, etc.) and outputs a set $\mathcal{S}_c$ (possibly empty) of surrogate objects of c. Each surrogate object is a 3-tuple (c_i, l_i, oa_i) where c_i is a citation and l_i is a hyperlink pointing to a copy in a repository where it is hosted (as determined by the metadata in the citation and it might be in a library with access toll) and oa_i is a hyperlink from where its OA full text may be downloaded. We rely heavily on Scholar to identify surrogates. A query string given to Scholar returns a set of articles $R = \{r_i\}$ where $r_i = (titleLink, oaLink, authors, venue, excerpt, citedByLink, relatedArticlesLink, allNVersionsLink)$ referring to the title with hyperlink to its source (as indicated by the citation), hyperlink to its OA copy (if any), the list of authors of the article, the venue of publication (including year of publication wherever present), an excerpt (normally part of the abstract), a hyperlink to articles citing it, a hyperlink to related articles and a hyperlink to different versions of this article respectively as shown in Figure 1. We keep only the first page of results in R (i.e., set $K = 10$) and locate the result $rmatch$ in it that best matches the input c. If $rmatch.oaLink$ points to an OA copy, we are done. Otherwise, we define a search space R for locating its surrogates as the union $R = R_1 \cup R_2$ where the set R_1 is the set of articles cited by $rmatch$ and R_2 is the set of related articles of $rmatch$. Here, we implicitly assume that surrogates of c are strongly related to c and hence, should be clustered with it in the results. Moreover, it is highly likely that a surrogate of c (like an extended journal version of a conference paper) will cite c if the surrogate is published after c. This heuristic suffices for most practical purposes although the exact ranking algorithm used by Scholar is not known publicly [1]. Both R_1 and R_2 can be very large sets. So we again restrict to the first page of results for each of them. To identify surrogates of c in R, we follow algorithm 2. We consider only the subset of articles published within a predefined length of time ($YMAX$ years) apart from the publication date of c. Specifically, we set $YMAX = 3$. Among these results, we use predefined thresholds on author and title similarities to identify surrogates. Intuitively, if there is a very high overlap in author lists of two articles, we allow the overlap in title to be low and vice versa. This also means we use criteria 5 of Section 3 with different thresholds. Similarity between author lists (author name in 'initials lastname' format) is estimated using Jaccard index. Similarity of titles is computed by first removing stop words from them, stemming the remaining portions (using Porter stemmer) and finally taking their cosine similarity. The thresholds are chosen heuristically using a test bench. In particular, the author similarity thresholds are

[5] http://academic.research.microsoft.com/
[6] http://citeseerx.ist.psu.edu
[7] https://www.semanticscholar.org/
[8] https://openaccessbutton.org/
[9] https://arxiv.org/
[10] https://www.researchgate.net/

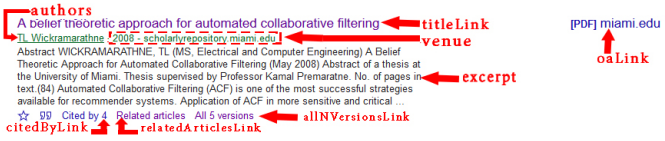

Figure 1: An article in Scholar with different components annotated

$AS1 = 0.9, AS2 = 0.5, AS3 = 0.1$ and the title similarity thresholds are $TS1 = 0.7, TS2 = 0.5, TS3 = 0.3$. Some user-defined functions are used in these algorithms but they have self-explanatory names and hence, are not discussed further.

Input : Citation c
Output: Set of surrogates $\mathcal{S}_c$
Function ***findSurrogate***

```
S_c ← φ;
;                                          /* Set K */
; /* Get first K Scholar articles for query c */
R ← getScholarArticles(c, K);
if |R| == 0 then
    return S_c;
else
    rmatch ← result in R closest to c;
    if rmatch.oaLink ≠ NULL then
        S_c ← {(makeCitation(rmatch),
        rmatch.titleLink, rmatch.oaLink)};
        return S_c;
    else
        ;      /* Get first K Scholar articles in
         cited-by list of rmatch */
        R1 ←
         getScholarArticles(rmatch.citedByLink, K);
        ;      /* Get first K Scholar articles in
         related-articles list of rmatch */
        R2 ←
         getScholarArticles(rmatch.relatedArticlesLink, K);

        R = R1 ∪ R2;
        forall result r in R do
            bSurr ← isSurrogate(c, r);
            if bSurr then
                S_c ← S_c ∪
                 {(makeCitation(r), r.titleLink, r.oaLink)};

        end
        return S_c
end
```

Algorithm 1: Identifying surrogates for a given citation

5 SYSTEM DESCRIPTION

We had two alternatives regarding when to tag the scholarly documents in a digital library with their surrogates, *offline* or *online*

Input : Citation c, Scholar article r
Output: Boolean indicating if r is a surrogate of c
Function ***isSurrogate***

```
;   /* Set YMAX, AS1, AS2, AS3, TS1, TS2, TS3 */
if r.oaLink ≠ NULL then
    yc ← getYear(c);
    yr ← getYear(r);
    if yr ≠ NULL and yc ≠ NULL then
        yearDiff = |yc − yr|;
        if yearDiff > YMAX then
            ;                          /* too far apart */
            return False;
    authorSim ←
     findAuthorSimilarity(getAuthors(c), getAuthors(r));

    titleSim ←
     findTextSimilarity(getTitle(c), getTitle(r));
    if authorSim ≥ AS1 then
        if titleSim ≥ TS3 then
            return True;
        else
            return False;
    else if authorSim ≥ AS2 then
        if titleSim ≥ TS2 then
            return True;
        else
            return False;
    else if authorSim ≥ AS3 then
        if titleSim ≥ TS1 then
            return True;
        else
            return False;
    else
        return False;
else
    return False;
end
```

Algorithm 2: Checking if a Scholar article is a surrogate of a given citation

(when user submits query). Offline operation allows complex computation and probably higher accuracy. However, given the increasing volume of publications on the Web or in the library, the process has to run frequently. Online operation must rely on simpler processing at the cost of lower accuracy. Currently, we have chosen to implement an online method. Our system called Surrogator consists of a GUI as a front-end through which a user can enter a query made of keywords or a citation. Additionally she must choose a source from the list {`NDLI, Google Scholar`}. It directs the query to the source via the https protocol. The results retrieved from the source are presented to the user. Surrogator simply acts as an intermediary here, except that it also checks if a retrieved result is access-restricted or OA. For NDLI, it consists in checking the *access restriction* field of the metadata while for Scholar, it needs to verify if a result has an associated link to an OA version. Thus, it

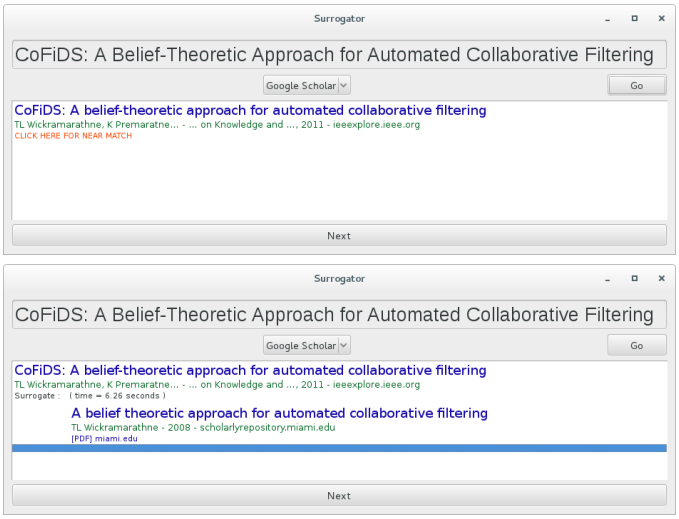

Figure 2: Surrogator showing Scholar's response to a query (top) and its surrogate (bottom)

understands for which articles the user might require surrogates and accordingly tags them with a hyperlink `CLICK HERE FOR NEAR MATCH`. If the user clicks on it, Surrogator retrieves the surrogates (if any) using Algorithm 1 and shows them alongside the article. Note that search for surrogate resources uses Scholar irrespective of the source chosen by the user in the GUI. The tool is written as a Python script and comprises around 1.3 KLOC. It uses several Python libraries like PyQt4, BeautifulSoup, numpy, webbrowser and nltk. We have executed it using Python 2.7.5 on Linux platform.

6 PRELIMINARY EVALUATION

Currently the tool has been tested on ad hoc queries only. We report a few examples for illustration purposes.

Example 6.1. When we search with the string 'CoFiDS: A belief-theoretic approach for automated collaborative filtering' in Scholar, it shows a link to the article [9] in IEEE Xplore with no hint if its OA surrogate exists. Surrogator with `source='Google Scholar'` also shows the same link to IEEE Xplore but allows the user to `CLICK HERE FOR NEAR MATCH`; clicking on it discovers the OA Master of Science thesis [8] available freely on the Web. See Figure 2. The same query in NDLI fetches many results and Surrogator can be used to fetch their surrogates as shown in Figure 3.

Example 6.2. The query 'Shen, Shigen, et al. "Evolutionary game based dynamics of trust decision in WSNs." Sensor Network Security Technology and Privacy Communication System (SNS & PCS), 2013 International Conference on. IEEE, 2013.' in Scholar finds [7] but does not identify any OA version. However, Surrogator identifies [6] as its surrogate. Interestingly, [6] mentions it is an expanded version of the above conference paper. It is thus an acceptable surrogate.

Sometimes our tool reports false positives and misses OA articles but we observed it to work correctly for a number of cases.

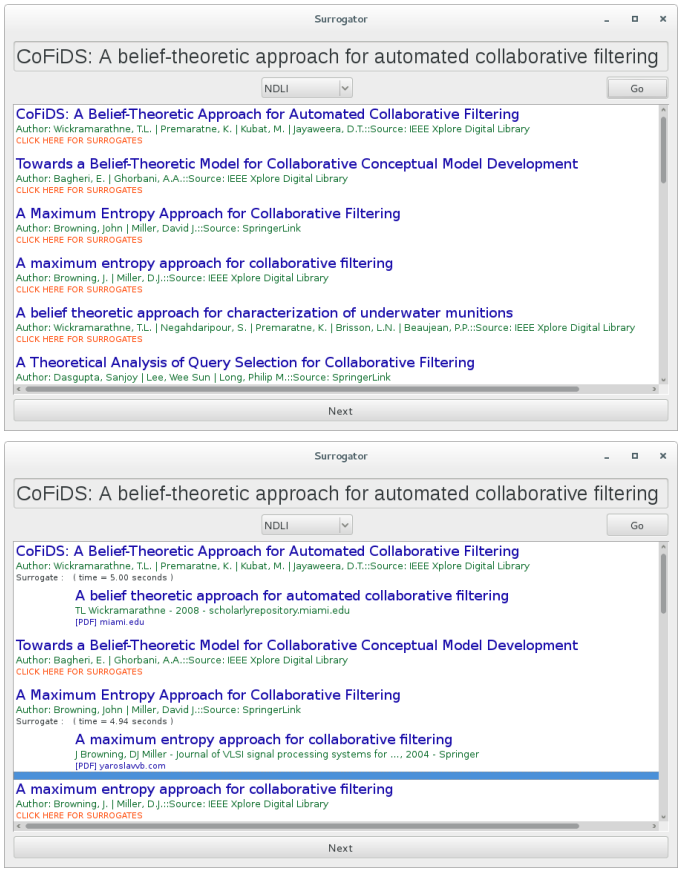

Figure 3: Surrogator showing NDLI's response to a query (top) and surrogates of the first and the third articles (bottom)

7 CONCLUSION

We presented a simple application to identify surrogates for access-restricted research publications and retrieve links to them with the usual search results. We believe it will be a valuable addition to an academic search engine or a digital library that has a large user base without subscription to the indexed libraries. Currently, we are exploring how to make the search more precise by comparing abstracts along with titles and authors (respecting the time constraints), take a more principled approach towards selection of thresholds and incorporate machine learning algorithms to improve the results. That said, the best way to overcome paywalls is to pull them down completely by encouraging researchers, institutes and funding agencies to opt for OA publications as much as possible.

ACKNOWLEDGMENTS

We thank Professor Partha Pratim Das, Department of Computer Science and Engineering, IIT Kharagpur and Joint-PI, NDLI Project for suggesting this problem. This work is supported by *Development of National Digital Library of India as a National Knowledge Asset of the Nation* sponsored by Ministry of Human Resource Development, Government of India.

REFERENCES

[1] Jöran Beel and Bela Gipp. 2009. Google Scholar's ranking algorithm: An introductory overview. In *Proceedings of the 12th International Conference on Scientometrics and Informetrics (ISSI'09)*, Vol. 1. Rio de Janeiro (Brazil), 230–241.

[2] Dalmeet Singh Chawla. 04 April 2017. Unpaywall finds free versions of paywalled papers. *Nature News* (04 April 2017). https://doi.org/10.1038/nature.2017.21765

[3] Chawki Hajjem, Stevan Harnad, and Yves Gingras. 2006. Ten-year cross-disciplinary comparison of the growth of open access and how it increases research citation impact. *arXiv preprint cs/0606079* (2006). (Retrieved November 3, 2017).

[4] Gali Halevi, Henk Moed, and Judit Bar-Ilan. 2017. Suitability of Google Scholar as a source of scientific information and as a source of data for scientific evaluation – Review of the Literature. *Journal of Informetrics* 11, 3 (2017), 823–834. https://doi.org/10.1016/j.joi.2017.06.005

[5] David Nicholas, Cherifa Boukacem-Zeghmouri, Blanca Rodríguez-Bravo, Jie Xu, Anthony Watkinson, Abdullah Abrizah, Eti Herman, and Marzena Świgoń. 2017. Where and how early career researchers find scholarly information. *Learned Publishing* 30, 1 (2017), 19–29. https://doi.org/10.1002/leap.1087

[6] Shigen Shen, Longjun Huang, En Fan, Keli Hu, Jianhua Liu, and Qiying Cao. 2016. Trust dynamics in WSNs: an evolutionary game-theoretic approach. *Journal of Sensors* 2016 (2016). https://doi.org/10.1155/2016/4254701

[7] Shigen Shen, Changyuan Jiang, Hua Jiang, Lizheng Guo, and Qiying Cao. 2013. Evolutionary game based dynamics of trust decision in WSNs. In *Proceedings of the 2013 International Conference on Sensor Network Security Technology and Privacy Communication System (SNS & PCS)*. IEEE, 1–4. https://doi.org/10.1109/SNS-PCS.2013.6553823

[8] Thanuka Lakmal Wickramarathne. 2003. *A belief theoretic approach for automated collaborative filtering.* Master's thesis. Electrical and Computer Engineering, University of Miami, USA. Open Access Theses. 182. http://scholarlyrepository.miami.edu/cgi/viewcontent.cgi?article=1181&context=oa_theses (Retrieved November 3, 2017).

[9] Thanuka L Wickramarathne, Kamal Premaratne, Miroslav Kubat, and Dushyantha Jayaweera. 2011. CoFiDS: A belief-theoretic approach for automated collaborative filtering. *IEEE Transactions on Knowledge and Data Engineering* 23, 2 (2011), 175–189. https://doi.org/10.1109/TKDE.2010.88

[10] Feng Xia, Wei Wang, Teshome Megersa Bekele, and Huan Liu. 2017. Big scholarly data: A survey. *IEEE Transactions on Big Data* 3, 1 (2017), 18–35. https://doi.org/10.1109/TBDATA.2016.2641460