BIOINFORMATICS ORIGINAL PAPER

Vol. 30 no. 1 2014, pages 104–114
doi:10.1093/bioinformatics/btt571

Data and text mining

Advance Access publication September 29, 2013

A user-oriented web crawler for selectively acquiring online content in e-health research

Songhua Xu*,†, Hong-Jun Yoon† and Georgia Tourassi

Biomedical Science and Engineering Center, Health Data Sciences Institute, Oak Ridge National Laboratory, One Bethel Valley Road, Oak Ridge, TN 37830, USA

Associate Editor: Igor Jurisica

ABSTRACT

Motivation: Life stories of diseased and healthy individuals are abundantly available on the Internet. Collecting and mining such online content can offer many valuable insights into patients' physical and emotional states throughout the pre-diagnosis, diagnosis, treatment and post-treatment stages of the disease compared with those of healthy subjects. However, such content is widely dispersed across the web. Using traditional query-based search engines to manually collect relevant materials is rather labor intensive and often incomplete due to resource constraints in terms of human query composition and result parsing efforts. The alternative option, blindly crawling the whole web, has proven inefficient and unaffordable for e-health researchers.

Results: We propose a user-oriented web crawler that adaptively acquires user-desired content on the Internet to meet the specific online data source acquisition needs of e-health researchers. Experimental results on two cancer-related case studies show that the new crawler can substantially accelerate the acquisition of highly relevant online content compared with the existing state-of-the-art adaptive web crawling technology. For the breast cancer case study using the full training set, the new method achieves a cumulative precision between 74.7 and 79.4% after 5 h of execution till the end of the 20-h long crawling session as compared with the cumulative precision between 32.8 and 37.0% using the peer method for the same time period. For the lung cancer case study using the full training set, the new method achieves a cumulative precision between 56.7 and 61.2% after 5 h of execution till the end of the 20-h long crawling session as compared with the cumulative precision between 29.3 and 32.4% using the peer method. Using the reduced training set in the breast cancer case study, the cumulative precision of our method is between 44.6 and 54.9%, whereas the cumulative precision of the peer method is between 24.3 and 26.3%; for the lung cancer case study using the reduced training set, the cumulative precisions of our method and the peer method are, respectively, between 35.7 and 46.7% versus between 24.1 and 29.6%. These numbers clearly show a consistently superior accuracy of our method in discovering and acquiring user-desired online content for e-health research.

Availability and implementation: The implementation of our user-oriented web crawler is freely available to non-commercial users via the following Web site: http://bsec.ornl.gov/AdaptiveCrawler.shtml. The Web site provides a step-by-step guide on how to execute the web crawler implementation. In addition, the Web site provides the two study datasets including manually labeled ground truth, initial seeds and the crawling results reported in this article.

Contact: xus1@ornl.gov

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on April 25, 2013; revised on September 25, 2013; accepted on September 26, 2013

1 INTRODUCTION

The Internet carries abundant and ever enriching user-generated content on a wide range of social, cultural, political and other topics. Life stories of patients are no exception to this trend. Collecting and mining such personal content can offer many valuable insights on patients' experiences with respect to disease symptoms and progression, treatment management, side effects and effectiveness, as well as many additional factors and aspects of a patient's physical and emotional states throughout the whole disease cycle. The breadth and depth of understanding attainable through mining this voluntarily contributed web content would be extremely expensive and time-consuming to capture via traditional data collection mechanisms used in clinical studies.

Despite the merits and rich availability of user-generated patient content on the Internet, collecting such information using conventional query-based web search is labor intensive for the following two reasons. First, it is not clear what are the right queries to use to retrieve the desired content accurately and comprehensively. For example, a general query such as 'breast cancer stories' would pull up over 182 million results using Google web search wherein only a selected portion, usually small (such as $<0.1\%$), of the whole search result set may meet the researcher's specific needs. Manually examining and selecting the qualified search results require extensive human effort. Second, clinical researchers have specific requirements regarding the user-generated disease content they need to collect. Query-based search engines cannot always support such requirements. Let us assume that a researcher wants to collect the personal stories of two groups of female breast cancer patients, those who have had children and those who have not. With much manual effort, the researcher might be able to obtain some stories of the first group, but so far no off-the-shelf general purpose search engine that we are aware of allows users to retrieve information that does not carry undesirable content (i.e. the support of negative queries). Given the steadily growing volume of patient-generated disease-specific online content, it is highly desirable to minimize

*To whom correspondence should be addressed.
†The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.

the manual intervention involved in source acquisition and subsequent mining processes. Although an extensive collection of automatic or largely automatic text mining algorithms and tools exists for analyzing social media content, limited efforts have been dedicated to developing automatic or largely automatic content acquisition tools and methods for obtaining online patient-generated content meeting certain e-health research needs and requirements. To meet this challenge in the e-health research community as well as the broader bioinformatics communities, we propose a user-oriented web crawler, which can acquire user-generated content satisfying particular content requirements with minimum intervention. We use cancer as the case study to demonstrate the value and impact of the proposed web crawling technology.

2 RELATED WORK

There is an extensive number of published studies reporting methods for adaptive web crawling (e.g. Aggarwal, 2002; Aggarwal *et al*., 2001; Almpanidis and Kotropoulos, 2005; Almpanidis *et al*., 2005; Badia *et al*., 2006; Chakrabarti *et al*., 2002; Chung and Clarke, 2002; Gao *et al*., 2006; Menczer *et al*., 2004; Pant and Srinivasan, 2005, 2006; Zhuang *et al*., 2005). In an early article, Aggarwal *et al*. (2001) contributed the pioneering idea of adaptively crawling the World Wide Web with the algorithmic guidance of using arbitrary predicates. In a separate article, the authors discussed the design of a learning crawler for topical resource discovery, which can apply learned knowledge in one crawling session for new crawling tasks (Aggarwal, 2002). Chakrabarti *et al*. (2002) proposed to accelerate focused web crawling by prioritizing unvisited uniform resource locators (URLs) in a crawler through simulating a human user's behaviors to identify and locate links meeting his/her information needs. Chung and Clarke (2002) proposed a topic-oriented collaborative crawling strategy that partitions the whole web into multiple subject topic-oriented areas and assigns one crawler for each partitioned area. Menczer *et al*. (2004) developed a framework for systematically evaluating topical specific crawling algorithms using multiple metrics. Pant and Srinivasan (2005) compared different classification schemes used for building adaptive crawlers that can learn from their past performance where the crawling process is simulated as a best-first graph search activity over the web. Almpanidis and Kotropoulos (2005) demonstrated the efficacy of combining text and link analysis for improving the web page collection productivity of focused crawlers. The same authors also adopted latent semantic indexing in a focused crawling effort to produce a vertical search engine. Zhuang *et al*. (2005) studied the problem of how to launch focused crawling efforts for harvesting missing documents in digital libraries.

Micarelli and Gasparetti (2007) overviewed methods for focused web crawling with an emphasis on approaches equipped with adaptive crawling capabilities. Babaria *et al*. (2007) proposed a method that tackles the focused web crawling problem as a large scale ordinal regression problem. Their crawler was subsequently supported by scalable ordinal regression solvers. Barbosa and Freire (2007) proposed an adaptive crawler that can efficiently discover hidden web entry points through web page content topic mining, link prioritization and exploration-based link visitation. De Assis *et al*. (2008) explored the impact of term selection in conducting genre-aware focused crawling efforts. Guan *et al*. (2008) explored how to use online topical importance estimation to efficiently and effectively guide the execution of focused crawlers. Chen *et al*. (2009) developed a cross-language focused crawling algorithm by applying multiple relevance prediction strategies. Batsakis *et al*. (2009) showed that by jointly analyzing web page content and link anchoring text, focused crawlers can better reach relevant pages. Ahlers and Boll (2009) introduced an adaptive geospatially focused crawler that can efficiently retrieve online documents relating to location information. Dey *et al*. (2010) introduced a focused web crawler for obtaining country-based financial data. Furuse *et al*. (2011) extended the Hyperlink-Induced Topic Search algorithm proposed by Kleinberg (1999) and introduced a new method to find related web pages with focused crawling techniques. The key feature of their algorithm was a mechanism to visit both forward and backward links from seed web pages for constructing an extended neighborhood graph to conduct focused web crawling. Liu and Milios (2012) introduced two probabilistic models for focused web crawling, one based on maximum entropy Markov model and the other based on linear-chain conditional random field. Fu *et al*. (2012) proposed to use opinion information to guide focused web crawling efforts for constructing a sentimentally aware web crawler.

Compared with all the crawling efforts surveyed earlier in the text and other similar pieces of work that cannot be included in this article due to space limitations, our proposed adaptive web crawler is characterized by three novel features. First, existing web crawlers rely heavily on the link structures of web graphs to determine the crawling priorities, under the assumption that relevant web pages are well interconnected. However, medical topic forums and blogs tend to be highly scattered with sparse or no links among them. Recognizing the challenge that this particular problem poses, we propose a new crawler that leverages a third-party search engine to massively and aggressively harvest candidate target crawling links, coupled with a parallel crawler navigation module that performs elaborate user-oriented crawling utility prediction and utility-driven crawling priority determination. Second, due to the critical importance of crawling utility prediction, our crawler carefully balances the time cost between repeatedly training a capable crawling utility predictor using a dynamically identified machine learning method and the actual time spent on crawling the web. Last but not least, our new crawler is equipped with an autonomous query composition and suggestion capability, built on content-based mining of exemplar search results. Compared with existing topic-based focused crawlers, the new crawler performs its function without a predefined topic ontology. Therefore, the crawler can be applied to efficiently and effectively acquire any content that matches the user's needs. This function enables users to harvest relevant content more comprehensively without the manual effort of composing explicit queries.

3 ESTIMATING WEB PAGE UTILITY SCORES AS FEEDBACKS FOR A WEB CRAWLER

To develop a self-adaptive web crawler, one key system module is a feedback component that is capable of estimating the utility

score of an arbitrary web page. The estimated web page utility scores can then guide the web crawler to make optimized crawling decisions. Below, we describe how we develop this predictive feedback module for web page utility score estimation. For easy reference and understanding, we list the key symbols and mathematical notations used in the algorithm description in the Appendix, which is available as a Supplementary File.

For an arbitrary web page wp and certain user information need Ω, we aim to develop a predictive model Φ that is capable of determining the utility score of wp according to Ω. The derived score is denoted as $\Phi(wp, \Omega) \in [0, 1]$, where the higher the score is, the more useful the web page is considered. To construct the predictive model Φ, we follow a supervised learning-based approach. Features extracted from wp as the input for Φ include words or word phrases detected from (i) the content words in the main body of an HTML file, (ii) words in the heading and subtitles of an HTML file and (iii) the anchor text embedded in an HTML file, including the URL(s) associated with the anchor text. To extract words in the main body of an HTML file, we use the Boilerpipe Java library introduced in Kohlschutter (2011). To obtain the heading and subtitles of an HTML file, we implement an HTML parser that extracts all the text enclosed in the HTML blocks of <*h*?*id*="…">…</*h*?> where ? stands for an integer number in the range of [1, 6]. For example, from the HTML block <*h*1*id*="*sectiontitle*">*Breast Cancer*</*h*1>, we can extract the heading text of 'Breast Cancer'. Similarly, to obtain the anchor text, we implement an HTML text parser that collects the annotation text associated with hypertext links embedded in an HTML file. Following the above procedure, we derive the aforementioned three sets of the text from wp, which are, respectively, denoted as $T_1(wp)$, $T_2(wp)$ and $T_3(wp)$.

We then apply the Rapid Automatic Keyword Extraction algorithm (RAKE) proposed by Rose *et al.* (2012) to identify a set of key words or phrases from each one of the text sets, $T_1(wp), \cdots, T_3(wp)$, prepared in the above. The results are, respectively, denoted as $kw_{i,j}(wp)$ $(i = 1, 2, 3; j = 1, \cdots, n_i)$, where n_i denotes the number of distinct key words extracted by the RAKE algorithm from the text set $T_i(wp)$ and the subscript j in the notation $kw_{i,j}(wp)$ indexes these key words individually. To train the web page utility estimator $\Phi(wp, \Omega)$ following a supervised learning-based procedure, our method also requires a set of manually labeled samples. For this purpose, we collect all the detected key words from web pages in **wp** and denote them as $\mathbf{kw} = \{kw^k_{i,j} | wp_k \in \mathbf{wp}\}$ where $kw^k_{i,j}$ is a short notation for $kw_{i,j}(wp_k)$. To train the utility estimator $\Phi(wp, \Omega)$, we present a selected collection of web pages $\mathbf{wp} = \{wp_\kappa\}$ and solicit human experts' manual ratings for these web pages according to the information quality measurement criterion Ω. Each human-labeled utility score, denoted as $\hat{\Phi}(wp_\kappa, \Omega)$, is a rational number in the range of [0, 1]. The higher the score value is, the better the quality of the web page is as considered by the human evaluator.

Given the substantial imbalance between the number of candidate key words that may be used as features for **wp** and the available human-labeled training samples, to train the utility estimator $\Phi(wp, \Omega)$, we first apply a feature selection procedure to reduce the amount of candidate key word features. This screening process consists of two steps. In the first step, we eliminate all the key words whose support values are below a certain empirically chosen threshold, which is set to five in all our experiments. After the infrequent key word filtering step, we denote the set of remaining key words as $\overline{\mathbf{kw}}$. In the second step of the key word reduction process, we examine the odd ratios of key words with respect to a human-labeled training set. Specifically, for each candidate key word $kw^k_{i,j} \in \overline{\mathbf{kw}}$ and a given threshold $\tau \in [0, 1]$, we first derive the key word's odd ratio $\psi(kw^k_{i,j}, \Omega, \tau, \mathbf{wp})$ with respect to the labeled training set as follows:

$$\psi(kw^k_{i,j}, \Omega, \tau, \mathbf{wp}) = \frac{p_{11}(kw^k_{i,j}, \mathbf{wp}) p_{00}(kw^k_{i,j}, \mathbf{wp})}{p_{01}(kw^k_{i,j}, \mathbf{wp}) p_{10}(kw^k_{i,j}, \mathbf{wp})}, \quad (1)$$

where $p_{11}(kw^k_{i,j}, \mathbf{wp})$ is the number of web pages in **wp** that contain the key word $kw^k_{i,j}$ and whose human-labeled utility score is above the threshold, i.e. $p_{11}(kw^k_{i,j}, \mathbf{wp}) = |\{wp_x \in \mathbf{wp} | kw^k_{i,j} \in wp_x, \hat{\Phi}(wp_x, \Omega) \geq \tau\}|$; $p_{10}(kw^k_{i,j}, \mathbf{wp})$ is the number of web pages in **wp** that contain the key word $kw^k_{i,j}$ and whose human-labeled utility score is below the threshold, i.e. $p_{10}(kw^k_{i,j}, \mathbf{wp}) = |\{wp_x \in \mathbf{wp} | kw^k_{i,j} \in wp_x, \hat{\Phi}(wp_x, \Omega) < \tau\}|$. Similarly, we define $p_{01}(kw^k_{i,j}, \mathbf{wp})$ and $p_{00}(kw^k_{i,j}, \mathbf{wp})$, which are counterparts for p_{11} and p_{10} with the only difference being that the web pages considered now do not contain the key word $kw^k_{i,j}$. That is, $p_{01}(kw^k_{i,j}, \mathbf{wp}) = |\{wp_x \in \mathbf{wp} | kw^k_{i,j} \notin wp_x, \hat{\Phi}(wp_x, \Omega) \geq \tau\}|$ and $p_{00}(kw^k_{i,j}, \mathbf{wp}) = |\{wp_x \in \mathbf{wp} | kw^k_{i,j} \notin wp_x, \hat{\Phi}(wp_x, \Omega) < \tau\}|$. We then rank all the candidate key words $kw^k_{i,j} \in \overline{\mathbf{kw}}$ in a descendant order according to their respective odd ratios derived from (1). When training the web page utility estimation model using a specific machine learning method, we progressively admit key words as features into the model one-by-one until the testing performance of the trained model as obtained through 10-fold cross-validation declines from the peak testing performance by >5%. We then retrospectively remove all the key word features admitted after the model achieves its peak performance moment.

In our experiments, we explored the following machine learning methods when developing the web page utility estimator $\Phi(wp_x, \Omega)$: Gaussian processes for regression, isotonic regression, least median squared linear regression, linear regression, radial basis function network, additive regression, bagging, regression by discretization and stacking. The implementations of all the above methods were provided by the Weka package (Hall *et al.*, 2009). According to our experimental results, we empirically found that the additive regression method achieves the best performance in all our experiments as measured by the 10-fold cross-validation scheme.

4 ADAPTIVE WEB CRAWLER FOR ACQUIRING USER-DESIRED ONLINE PATIENT CONTENT

Given a web page utility estimator $\Phi(wp_\kappa, \Omega)$ trained from a set of human-labeled example web pages $\{\hat{\Phi}(wp_\kappa, \Omega)\}$, we can then develop a user-oriented web crawler that is capable of adaptively acquiring relevant web pages that satisfy the user information requirement Ω. Figure 1a illustrates the overall architecture of our adaptive web crawler, the design of which will be discussed in this section. Figure 1b provides a companion computational data flow of the crawler design with more technical details.

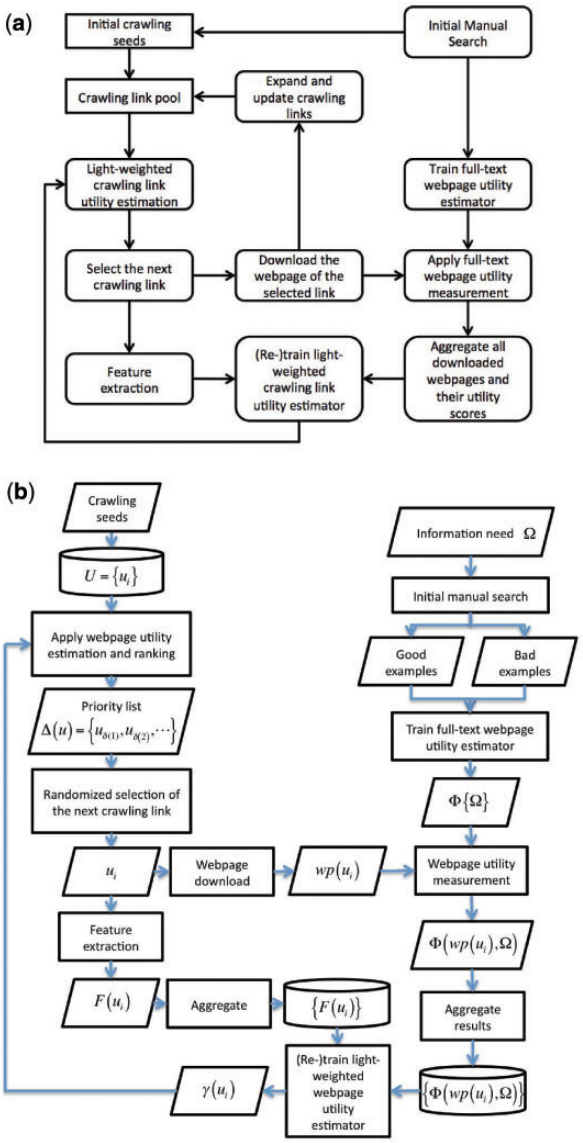

Fig. 1. The overall architecture (**a**) of our adaptive web crawler design and its companion computational data flow (**b**)

4.1 Crawler design I: a naive crawler design

A straightforward construction of a web crawler, which is denoted as $\mathcal{C}_1$, is to plug the trained utility estimator $\Phi(wp_\kappa, \Omega)$ at the end of the crawling pipeline. Using this simple design, the crawler expects a human user to supply a series of queries $\{Q_1, Q_2, \cdots, Q_n\}$, on which $\mathcal{C}_1$ then executes each query either sequentially or in parallel to obtain a collection of search result web pages $\{wp_x\}$. For each obtained search result page wp_x, $\mathcal{C}_1$ applies the trained estimator $\Phi(wp_x, \Omega)$ to derive the page's utility score. Only pages that carry utility scores above a user-specified threshold τ will be output as the filtered crawling results. One of the key shortcomings associated with the design of $\mathcal{C}_1$ is that $\mathcal{C}_1$ requires human users to provide a set of manually composed queries. As discussed in the beginning of this article, manual query composition poses non-trivial challenges for human end users. Preparing a set of queries that precisely expresses the user search intent and also ensures a comprehensive and diverse coverage over the user demanded topics are well beyond the reach of most human searchers. To overcome this limitation, we introduce a heuristic query generator, which is described later in this article. In addition, we further equip the crawler with an adaptive feature to speedup its execution. To differentiate from the first version of the crawler $\mathcal{C}_1$, we name the adaptive version of the crawler as $\mathcal{C}_2$, the design of which will be described in the next section.

4.2 Crawler design II: a user-oriented web crawler design

4.2.1 Problem Statement Given a list of search result URL links $\mathbf{u} = \{u_1, u_2, \cdots, u_z\}$, the goal of an adaptive crawler is to adaptively and dynamically determine a priority list $\Delta(\mathbf{u}) = \{u_{\delta(1)}, u_{\delta(2)}, \cdots, u_{\delta(z)}\}$ where these links shall be crawled. Here $\delta()$ is a ranking function over all the candidate web pages to be crawled and the notation of $u_{\delta(i)}$ represents the URL of the i-th web page that the crawler visits since the beginning of a crawling session. As discussed previously, exhaustively downloading all the links may take a long time wherein many links may not be relevant to the end user's need. Hence in practice, given a priority list $\Delta(\mathbf{u})$ and a certain amount of downloading time that a user can afford, the crawler will only download the header part of the prioritized results until all available time is used up. Determining a truly optimal ranking function $\delta()$ requires the full knowledge regarding the utilities of web pages pointed to by these links, which is well beyond the reach of any runtime algorithm. Therefore, we have to rely on some heuristics to construct the ranking function.

Given a web page's URL link u_i, once the crawler actually downloads the actual web page $wp(u_i)$ that is pointed to by the link, we can measure the utility of the individual search result page $\Phi(wp(u_i), \Omega)$ by applying the trained utility estimation function $\Phi(\cdot, \cdot)$. The problem now reduces to whether we can predict the value of $\Phi(wp(u_i), \Omega)$ based on all the available information regarding the link u_i. A typical search engine will return a highlight snippet about u_i, including u_i's URL, a running head text and a brief piece of selected text from the search result. Using text features $\mathbf{F}(u_i)$ extracted from the above three types of information, our method wants to predict the value of $\Phi(wp(u_i), \Omega)$. In our current implementation, $\mathbf{F}(u_i)$ includes all the individual non-stop words in the above snippet text of the web page $wp(u_i)$. The prediction function is stated as follows:

$$\Upsilon(wp(u_i), \Omega) : \mathbf{F}(u_i) \rightarrow \Phi(wp(u_i), \Omega) \tag{2}$$

The key difference between the function $\Upsilon(wp(u_i), \Omega)$ and $\Phi(wp(u_i), \Omega)$ is that $\Phi(wp(u_i), \Omega)$ is estimated using text features extracted from the full text of the web page $wp(u_i)$ after the web page is downloaded, whereas $\Upsilon(wp(u_i), \Omega)$ is estimated using text features extracted from the snippet text of the web page $wp(u_i)$ before the web page is downloaded. It should be noted that to obtain a sample pair of $(\mathbf{F}(u_i), \Phi(wp(u_i), \Omega))$ for training the prediction function Υ, there is a penalty in term of the link visitation time, which is non-trivial in many practical scenarios as analyzed at the beginning of this article. Therefore, from the runtime efficiency perspective, it is desirable to use $\Upsilon(wp(u_i), \Omega)$ instead of $\Phi(wp(u_i), \Omega)$ if the error can be tolerated. Let $\psi(\Upsilon, t)$ be the prediction error of the trained predictor Υ at a given time moment t.

With the progression of a crawler's execution in any crawling session, more training examples will be accumulated, which help train a more accurate predictor. Lastly, it is noted that the prediction accuracy of $\Upsilon(wp(u_i), \Omega)$ is affected by the amount of training samples available. Because in this work, we assume the time duration of deriving the value of $\Phi(wp(u_i), \Omega)$ is negligible once the web page $wp(u_i)$ is downloaded.

4.2.2 Objective formulation We can now formulate the optimal URL link visitation planning task as the following dynamic schedule optimization problem. Let us consider a pool of candidate URLs $\mathcal{U} = \{u_1, u_2, \cdots, \}$ to be crawled. The link u_i has the utility score of $\Phi(wp(u_i), \Omega)$, if its pointing destination web page has been downloaded, which costs time $T(u_i)$ to access. In this work, we do not consider the time of applying the trained estimator $\Phi(\cdot, \cdot)$ onto the web page with the URL u_i because all the candidate machine learning methods we considered for this predictive modeling task are able to yield the prediction values almost instantly once they are trained. For the time duration $T(u_i)$, it is primarily the response time of the Web site where the target link u_i points to. The link u_i has a lightweight utility prediction score $\Upsilon(wp(u_i), \Omega, t_j)$ at time t_j, which we assume will take an ignorable amount of time to assess, as evaluating the function of $\Upsilon(wp(u_i), \Omega, t_j)$ does not require downloading the web page $wp(u_i)$. The above prediction has its relative prediction error interval of $[-\psi_k(wp(u_i), t_j), \psi_k(wp(u_i), t_j)]$. In our current implementation, we derive multiple error intervals for each estimation, which are differentiated by the subscript k. The estimation confidence for the relative error interval of $[-\psi_k(wp(u_i), t_j), \psi_k(wp(u_i), t_j)]$ is denoted as $\eta_k(wp(u_i), t_j) \in [0, 1]$, the higher the score is, the more confident the estimation is. To measure the relative error intervals and their corresponding confidence, each time when a candidate predictive model is trained as allowed by the available training data, we then test the model on the testing dataset using the leave-one-out evaluation scheme. Instead of aggregating all the testing errors into some overall error metric, we compute the $90\%, 80\%, \cdots, 10\%, 5\%$ percentile error intervals $[-\psi_k(wp(u_i), t_j), \psi_k(wp(u_i), t_j)](k = 1, \cdots, 10)$ where $\psi_k(wp(u_i), t_j)$ is assigned as the relative error of the k-th percentile error value derived in the aforementioned procedure. Its corresponding confidence value $\eta_k(wp(u_i), t_j)$ is assigned as the corresponding percentile value, which indicates the likelihood that the true error will indeed fall into the estimated error interval.

Given the above notations, we can quantitatively state the goal of our optimization problem as follows: given a certain amount of time t_x permissible for a crawler, the problem goal is to find an optimal URL visitation trajectory $\mathcal{V}(t_0, t_x) = (u_{\delta(1)}, u_{\delta(2)}, \cdots, u_{\delta(n_x)})$ that maximizes the total utility score of all the URLs visited since the beginning moment of a crawling session t_0 and ends by the time period t_x wherein the URL sequence $\{u_{\delta(1)}, u_{\delta(2)}, \cdots, u_{\delta(n_x)}\}$ encompasses all the web pages the crawler manages to visit under the visitation trajectory $\mathcal{V}(t_x)$ within the given time duration of $[t_0, t_x]$. One final touch regarding the problem statement is that the URL pool $\mathcal{U}$ itself is a dynamically growing set. Because each time when the web page $wp(u_i)$ pointed to by the URL u_i is visited, the crawler may discover new URLs from $wp(u_i)$, which will then be extracted and added into the pool $\mathcal{U}$. For reference, we denote the snapshot of the pool of candidate URLs awaiting to be crawled at the time moment of t_i as $\mathcal{U}(t_i)$. At the beginning of a crawling session, i.e. at the initial time moment t_0, no web pages have been crawled. The corresponding candidate URL pool $\mathcal{U}(t_0)$ is the pool of seed web page URLs for launching the crawler. Formally, we can express the optimization objective as follows:

$$\text{maximize}_{\mathcal{V}(t_0, t_x) = (u_{\delta(1)}, \cdots, u_{\delta(n_x)})} \mathcal{G}(t_0, t_x, \mathcal{V}(t_0, t_x)) \tag{3}$$

in which the objective function is defined as follows:

$$\begin{aligned} &\mathcal{G}(t_0, t_x, \mathcal{V}(t_0, t_x)) = \sum_{i=1}^{n_x} \Phi(wp(u_{\delta(i)}), \Omega) \text{ subject to :} \\ &\sum_{i=1}^{n} T(u_{\delta(i)}) \le t_x - t_0; \\ &u_{\delta(i)} \in \mathcal{U}\left(\sum_{j=1}^{i-1} T(u_{\delta(j)})\right) (i = 1, \cdots, n) \end{aligned} \tag{4}$$

Note that in (4), the first constraint, $\sum_{i=1}^{n} T(u_{\delta(i)}) \le t_x - t_0$, ensures that visiting all the URLs along the visitation trajectory $\mathcal{V}(t_0, t_x)$ will not take the total length of the allocated time. The second constraint $u_{\delta(i)} \in \mathcal{U}(\sum_{j=1}^{i-1} T(u_{\delta(j)}))$ guarantees that at any moment when the crawler executes the visitation trajector $\mathcal{V}(t_0, t_x)$, which we assume to be the moment after the $(i-1)$-th URL in $\mathcal{V}(t_0, t_x)$ is visited but before the i-th link is to be visited, the next URL the crawler is going to visit shall only come from the current candidate URL pool $\mathcal{U}(\sum_{j=1}^{i-1} T(u_{\delta(j)}))$ wherein $\sum_{j=1}^{i-1} T(u_{\delta(j)})$ is the corresponding time stamp for the moment. Also, by definition, $\delta()$ is a ranking function, which implies that $i \neq j \Rightarrow \delta(i) \neq \delta(j)$.

It shall be noted that the target function $\mathcal{G}(t_0, t_x, \mathcal{V}(t_0, t_x))$ defined in (4) cannot be directly used in the actual optimization process during runtime because as mentioned earlier, to obtain the information $\Phi(wp(u_{\delta(i)}), \Omega)$, the crawler first needs to visit the URL $u_{\delta(i)}$, which would incur the cost of link visitation time $T(u_{\delta(i)})$. Expecting to have the full knowledge of $\Phi(wp(u_{\delta(i)}), \Omega)$ for all $u_{\delta(i)}$s involved in the optimal planning process is impractical because this would require the crawler to visit every link in the candidate URL pool, which is highly undesirable. Taking into account the considerable amount of time cost for 'knowledge acquisition' in terms of the time required for downloading the web page $wp(u_{\delta(i)})$ to derive the value of $\Phi(wp(u_{\delta(i)}), \Omega)$, we revise the objective function $\mathcal{G}(t_0, t_x, \mathcal{V}(t_0, t_x))$ in (4) and formulate a new objective function $\widehat{\mathcal{G}}(t_0, t_x, \mathcal{V}(t_0, t_x))$ that can be evaluated computationally on the fly. For simplicity, we use the short notation of T_i to denote $\sum_{j=1}^{i-1} T(u_{\delta(j)})$, which indicates the time moment immediately after the first $i-1$ URLs have been visited by the crawler in a crawling session:

$$\begin{aligned} &\text{maximize}_{\mathcal{V}(t_0, t_x) = (u_{\delta(1)}, \cdots, u_{\delta(n_x)})} \widehat{\mathcal{G}}(t_0, t_x, \mathcal{V}(t_0, t_x)) = \\ &\sum_{i=1}^{n_x} \frac{\sum_{k=1}^{10} \eta_k(wp(u_{\delta(i)}), T_i)(1 - \psi_k(wp(u_{\delta(i)}), T_i)) \Upsilon(wp(u_{\delta(i)}), \Omega, T_i)}{10} \\ &\text{subject to :} \\ &\sum_{i=1}^{n} T(u_{\delta(i)}) \le t_x - t_0; u_{\delta(i)} \in \mathcal{U}(T_i) (i = 1, \cdots, n). \end{aligned} \tag{5}$$

To understand the design of (5), for a given utility estimate for a web page $\Upsilon(wp(u_{\delta(i)}), \Omega, T_i)$, for its k-th relative error interval $[-\psi_k(wp(u_{\delta(i)}), T_i), \psi_k(wp(u_{\delta(i)}), T_i)]$, the corresponding lowest utility estimate is $(1 - \psi_k(wp(u_{\delta(i)}), T_i))\Upsilon(wp(u_{\delta(i)}), \Omega, T_i)$ with the estimate confidence being $\eta_k(wp(u_{\delta(i)}), T_i)$. Please note the above estimate gives a conservative measure regarding the utility scores harvested from crawled web page as the actual relative error may not be as high as the maximum value, $\psi_k(wp(u_{\delta(i)}), T_i)$, in the error interval. By averaging such estimates for all 10 error intervals, we derive a conservative estimation of the confidence-modulated utility score for the i-th web page crawled. Adding up all n_x web pages the crawler acquires along the visitation trajectory, we derive the overall confidence-modulated utility score for all the web pages acquired by the crawler during the period of $[t_0, t_x]$ under the most conservative estimate.

4.2.3 Problem resolution The above formulated optimization problem is apparently computationally intractable because the three terms involved in the objective function $\eta_k(wp(u_{\delta(i)}), T_i)$, $\psi_k(wp(u_{\delta(i)}), T_i)$ and $\Upsilon(wp(u_{\delta(i)}), \Omega, T_i)$ as well as the candidate URL pool $\mathcal{U}(T_i)$ are all functions of the time consumed up to a certain intermediate step, i.e. T_i, in the simultaneous crawler online execution and dynamic planning process. Others may want to use dynamic programing to derive an optimal solution for the problem. However, given the typically large number of steps involved in the planning process it is commonly expected for a crawler to encounter and gather tens of thousands or even millions of web pages during one run. Consequently, the cost of executing a dynamic programing procedure can quickly grow computationally unaffordable.

To derive the solution in a computationally affordable way, we hence turn to an approximation algorithm-based approach, which can be executed efficiently in practice. Let t_z be a moment when the algorithm needs to probabilistically choose a crawling target. At this moment, the algorithm chooses from the then candidate URL pool $\mathcal{U}(t_z)$ a link u_i with the probability of $p_i(t_z) = \frac{q_i(t_z)}{\sum_{u_j \in \mathcal{U}(t_z)} q_j(t_z)}$ where $q_i(t_z)$ is defined as follows:

$$q_i(t_z) = \sum_{k=1}^{10} \eta_k(wp(u_i), t_z)(1 - \psi_k(wp(u_i), t_z))\Upsilon(wp(u_i), \Omega, t_z). \quad (6)$$

Note that before we acquire a sufficient number of samples of $\Phi(wp(u_i, \Omega))$, we cannot train a reliable model to serve as $\Upsilon(wp(u_i, \Omega))$ because the training data required of $\Upsilon(wp(u_i, \Omega))$ is in the form of pairs of $(\mathbf{F}(u_i), \Phi(wp(u_i), \Omega))$ (see the defintion of (2) in Section 4.2.1). Therefore, we assign a uniformly random distribution over all candidate URLs that are currently available. That is, we set $p_i(t_z) = \frac{1}{|\mathcal{U}(t_z)|}$. In our current implementation, we use (6) to assign probability distributions of $\{p_i(t_z)\}$ when the crawler has acquired >1000 web pages in a crawling session.

4.3 Workflow of prototype web crawler

Putting all the algorithmic modules together, we constructed a user-oriented adaptive web crawler. The overall operating procedure of the pipeline is as follows. Given a specific user web crawling need Ω, the user first conducts a few brief web query efforts wherein he/she would compose a few queries to search the web. Within the returned search results, the user then selectively identifies a few 'good' search result web pages $\{wp_i^{\text{good}}\}$ that satisfy the need Ω as well as a few 'bad' search result web pages $\{wp_i^{\text{bad}}\}$ that fail to satisfy the need. Given such initial set of positive and negative examples, the method then trains the predictive model $\Phi(wp_x, \Omega)$ for determining the utility score of an arbitrary search result web page wp_x using the method introduced earlier in Section 3. Given the trained web page utility assessment model $\Phi(wp_x, \Omega)$, we then launch our advaptive web crawler following the design introduced at Section 4.2.

During the online execution of the web crawler, according to the utility scores $\Phi(wp_x, \Omega)$ derived from the currently downloaded web pages $\{wp_x\}$, we can then train the lightweighted web page utility estimation function $\Upsilon(wp(u_i), \Omega, t_z)$, which can be deployed to estimate the utility of a web page $wp(u_i)$ without requiring the web crawler to first download the page, thus offering a tremendous time saving for the crawling procedure. To balance two alternative options, (i) to derive a most accurate estimation model $\Upsilon(wp(u_i), \Omega, t_z)$ by making use of all available training samples through timely model retraining and (ii) to save some model training time for the actual web crawling process, we adopt the tradeoff solution of only retraining the model of $\Upsilon(wp(u_i), \Omega, t_z)$ when there are 10% additional training samples available. Note that after each round of retraining, we will acquire a new version of the functions $\eta_k(wp(u_{\delta(i)}), T_i)$, $\psi_k(wp(u_{\delta(i)}), T_i)$ and $\Upsilon(wp(u_{\delta(i)}), \Omega, T_i)$, which allows the crawler to update its crawling strategy according to (6).

It shall also be noted that during any moment of the crawler's execution process, an end user can always label additional search results as being satisfactory or not with respect to the information acquisition need Ω. Such interactive user labeling process allows an end user to dynamically expand the training sets of $\{wp_i^{\text{good}}\}$ and $\{wp_i^{\text{bad}}\}$. Once these two training sets are expanded, the model of $\Phi(wp_x, \Omega)$ will be updated, which in turn will lead to an updated model of $\Upsilon(wp(u_i), \Omega, t_z)$.

5 EXPERIMENTAL RESULTS

To explore the potential of our proposed web crawler, we conducted two crawling tasks where we, respectively, collected patient-generated online content regarding two cancer research topics: one on breast cancer patients and their lifestyle choices and the other on lung cancer patients with history of smoking. Both crawling tasks are relevant for addressing epidemiological type questions by analyzing online personal stories of cancer patients who meet the specific selection criteria imposed by the cancer researcher. For the breast cancer study, 133 positive and 875 negative exemplar search results were manually collected by a human researcher to initialize the crawling system. These sample results were mostly collected by manually searching established cancer-related forums such as the American Cancer Society's cancer survivor network forum ACS (2013). To assess the performance advantage of the new crawler with respect to the state-of-the-art, we compared our crawler with one of the leading adaptive web crawlers proposed by Barbosa and Freire (2007) as a peer crawling method. We implemented the prototype system of the peer method according to the design and all technical details disclosed in their original publication. In our comparative

experiments, the same set of example search results is used to train and initialize the peer crawling method.

The runtime performance of both crawlers is reported in Figure 2 for both case studies. The figure shows the total amount of web pages crawled (Raw Volume), the amount of web crawling results obtained by a crawler that are estimated relevant to the current crawling objective wherein the relevance estimation is performed by the crawler's built-in self-assessment capability (Gross Volume) and the amount of satisfactory search results obtained as judged by the human end user (Net Volume). In addition, we also show the temporal precision (Precision) and cumulative precision (Cumulative Precision) of either crawler throughout the whole crawling process as measured via a sampling-based human evaluation procedure. Owing to the large volume of crawling results produced, it is prohibitively expensive to ask a cancer researcher to manually evaluate the quality of each individual result for deriving the precision of either crawler. Therefore, we used a sampling-based manual evaluation procedure wherein the researcher manually evaluated the quality of every other 50 crawling results using binary labels (1: relevant, 0: irrelevant) regarding the relevance of the sampled result. Comparing between (a) and (b) in Figure 2, we can see that the peer method obtains a slightly larger raw crawling volume than our crawler, which is possibly caused by the more sophisticated decision logic used by our method. This qualitative difference is weakly reversed when it comes to the gross crawling volume, suggesting that our method works better in guiding itself to encounter more useful web pages than the peer approach. More importantly, the precision attained by the new crawler, both in terms of its temporal precision and cumulative precision as evaluated by the human searcher, is consistently superior to that of the peer method. Consequently, the net crawling amount obtained by our proposed crawler is substantially more abundant than that of the peer method. Table 1 reports the quantitative difference between the two methods in terms of their net crawling volumes.

To further assess the performance advantage of our web crawler, we performed a secondary experiment for the breast cancer case study by asking our adaptive crawler to repeat the search using a substantially reduced number of exemplar search results. Specifically, the crawler was initialized using only 67 positive and 438 negative search results. The results are also shown in the figure, which demonstrate a similar qualitative performance difference between the two methods in comparison: (i) the peer method encounters more web pages than our crawler in that it obtains a larger raw crawling volume than the new method; (ii) the gross volume of the two methods is comparable, with a slight advantage achieved by the new crawler; (iii) the precision of the new crawler is substantially higher than that of the peer method, leading to a substantially larger net volume of results crawled by our approach. Table 1 shows a quantitative comparison between the two adaptive web crawling methods in terms of their effectiveness and time efficiency of harvesting user-desired online content for two cancer case studies—one on breast cancer and the other on lung cancer research using both an enriched set and a reduced set of user-labeled exemplar search results to train each crawler, respectively. This table reports the number of distinct web pages crawled that are relevant to the specific information

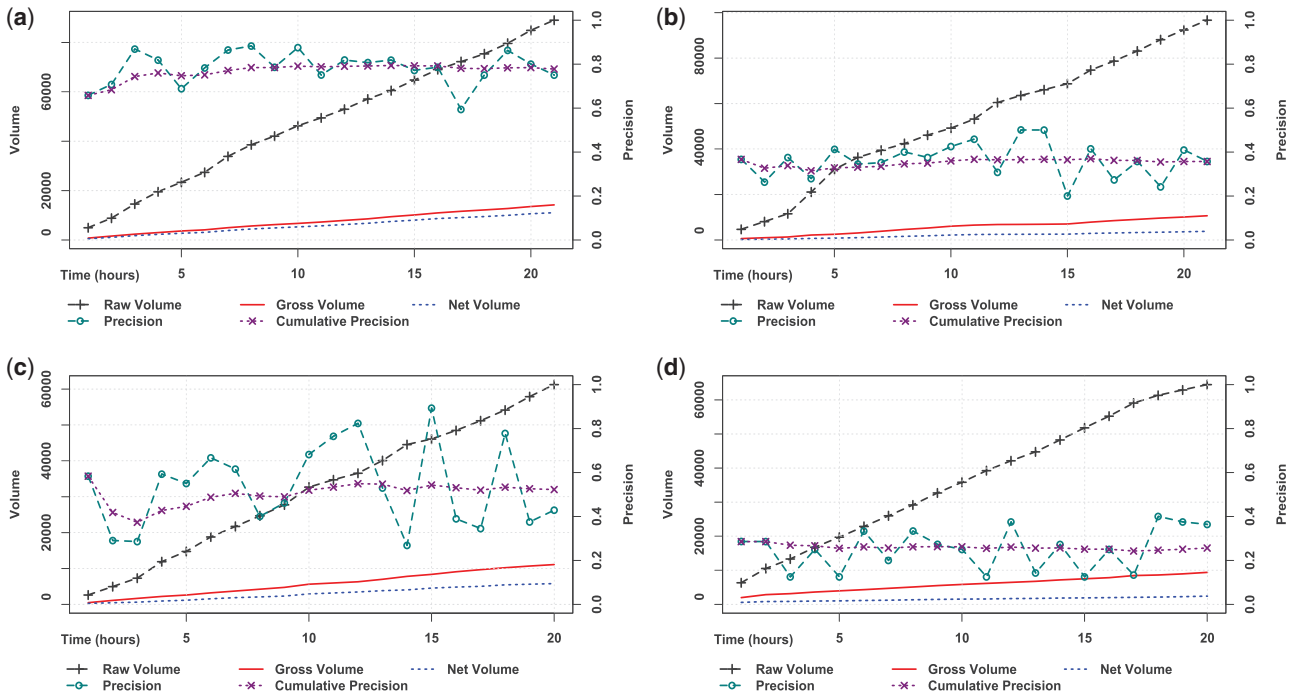

Fig. 2. Performance comparison between our new crawling method (**a**) and the peer method by Barbosa and Freire (2007) (**b**) for collecting life stories of breast cancer patients. Both crawling methods were initialized using the same set of labeled samples. To explore the influence of the training sample size, we randomly selected 50% of the available positive and negative training samples and repeated the aforementioned comparable analysis (**c** and **d**)

Table 1. Quantitative comparison between the performance of our crawler and that of the peer method (Barbosa and Freire, 2007) in terms of the net volumes of user-desired online content harvested by each crawler for progressively extended periods of crawling time

Comparison	Cumulative crawling time (hour)				
Method	1	6	11	16	20
(a) Crawling for the breast cancer study (enriched training set)					
Peer crawler	225	1021	2403	2904	3607
Our crawler	506	3085	5732	8691	10628
Rate (our/peer)	2.25	3.02	2.39	2.99	2.95
(b) Crawling for the breast cancer study (reduced training set)					
Peer crawler	562	1125	1565	1964	2403
Our crawler	284	1569	3184	4830	5800
Rate (our/peer)	0.51	1.39	2.03	2.46	2.41
(c) Crawling for the lung cancer study (enriched training set)					
Peer crawler	166	883	1620	2325	3263
Our crawler	404	3124	5286	6810	8130
Rate (our/peer)	2.43	3.54	3.26	2.93	2.49
(d) Crawling for the lung cancer study (reduced training set)					
Peer crawler	107	1104	2021	2643	3404
Our crawler	168	1409	2285	3302	4881
Rate (our/peer)	1.57	1.28	1.13	1.25	1.43

Note: See more detailed explanations about the comparative experiments for performance benchmarking in the text.

needs and requirements of either study, referred to as 'net volumes', after executing each adaptive crawling process for progressively extended periods of time, namely after 1, 6, 11, 16 and 20 h of crawling. To derive the end user-evaluated net volumes of online content obtained up to each snapshot moment of a crawling process, we adopt the aforementioned selective sampling-based manual evaluation strategy. As mentioned earlier in the text, both our adaptive web crawler and the state-of-the-art peer crawler were trained and initialized using the same set of seed URLs and user-labeled exemplar search results for capturing and understanding the type and scope of online content desired by e-health researchers in either crawling experiment. Instead of reporting the raw volumes of web content acquired by the two crawlers, respectively, we choose to compare the net volumes of selectively acquired online content because the latter volumes more truthfully indicate the amount of acquired web content relevant and useful for e-health researchers in either study. The last row of each subtable also reports the rate of harvesting user-desired online content by our crawler with respect to the harvesting rate of the peer crawler at each crawling snapshot moment. In both comparative studies, the adaptive crawler is consistently superior to the peer method. In addition, the comparative study using a reduced set of user-labeled sample web search results further supports the advantage of the new crawler. Namely, our new crawler can be initialized with a small set of exemplar search results for quick launch, which is still capable of obtaining superior crawling performance.

Similarly, for the lung cancer case study, 73 positive and 700 negative web pages from sites such as the Lung Cancer Support Community were manually collected as exemplar search results to initialize the system. For comparison purposes, we further conducted a peer crawling session by using a reduced set of human labels consisting of 50 positive and 400 negative sample web pages. The runtime performance of our crawler for the new crawling task under the two initialization conditions is reported in Figure 3 and Table 1. Similar to the breast cancer case study, the new crawler demonstrates clear advantage over the peer method for the lung cancer case study as well.

As illustrated in Figure 3, for the lung cancer study, the precision attained by the proposed new crawler, both in terms of its temporal precision and cumulative precision according to the evaluation by the human end user, is consistently superior to that of the peer method. Benefited by this prevailing advantage of the new crawler in more precisely locating and acquiring online content relevant to the specific crawling needs, the net crawling amount by the new crawler consistently surpasses that of the peer method for both crawling sessions using the enriched and reduced training sets of user-labeled exemplary search results. This conclusion can also be quantitatively verified by the comparative rate of harvesting speeds between the two crawling methods as reported in the last rows of the subtables (b) and (d) in Table 1.

We further observe that when trained using the enriched set of user-labeled exemplary search results, our crawler obtains a roughly comparable rate of raw crawling volumes as the peer method. Yet, when trained using the reduced set of user-labeled search results, for the initial 10 h of crawling, our crawler exhibits a slower rate of harvesting the raw crawling volume than the peer method. However, as the crawling time keeps increasing, our crawler gradually catches up with the peer crawler in terms of the raw crawling volume. At the end of the 21 h of crawling, the raw volumes of results obtained by both methods become highly comparable.

For the initial slower rate of acquiring the raw crawling volume, recalling the early conclusion that the new crawler consistently sustains a superior crawling precision than the peer method, it seems that the proposed new crawler favors precision rather than the raw crawling volume as compared with the peer crawler. That is, we hypothesize that the new crawler may choose to spend more time in executing its adaptive web crawling logics to determine on-the-fly a web crawling plan for the current crawling task rather than the alternative strategy of performing more operations of raw web crawling with a less deliberated adaptive crawling plan. Consequently, the new crawler may exhibit a slower rate of web crawling than the peer method. As verified by the performance evaluation results reported in Figure 3 and Table 1, such prioritized execution of the planning process for adaptive web crawling turns out to yield more effective overall return in terms of the net volume of crawling results obtained. Such tactical crawling strategy determination may not be necessary or evident when the amount of available user-labeled exemplary training samples is abundant; yet when the samples are scarce or less informative, extra planning in the adaptive crawling process may be more important for the proposed crawler compared with the peer method.

For the second phase of the gradual speedup of the proposed adaptive crawler, a potential reason is that when more online content has been crawled and thus becomes available for retraining the lightweighted crawler navigation model $\Upsilon(wp(u_i), \Omega, t_z)$, the utility of frequent model retraining declines. Thus, by shifting

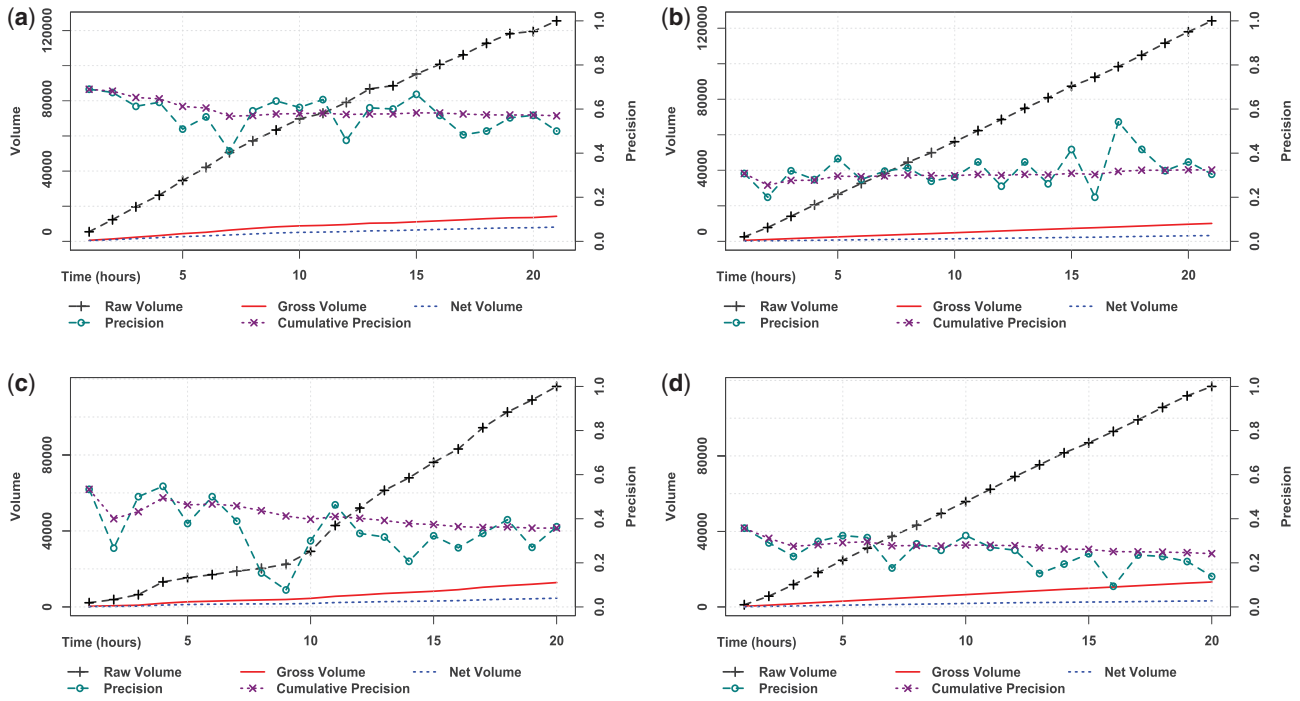

Fig. 3. Performance comparison between our new crawling method (**a**) and the peer method by Barbosa and Freire (2007) (**b**) for collecting personal stories of lung cancer patients with history of smoking. Both crawling methods were initialized using the same set of labeled samples. To explore the influence of the training sample size, we randomly selected a subset of the available positive and negative training samples and repeated the aforementioned comparable analysis (**c** and **d**)

more time from the model retraining to actual web crawling operations, the crawler can obtain the raw volume faster. A second potential reason for the accelerated rate of the raw crawling volume is that after the crawler has accumulated a critical mass amount of online content, such an intermediate result set may lead to a more effective $\Upsilon(wp(u_i), \Omega, t_z)$ for guiding the crawler to visit web pages with fast link visitation time.

6 DISCUSSION

Comparing the design of our newly proposed adaptive crawler with that of the peer method presented in Barbosa and Freire (2007), there exist four key aspects of similarities between the two approaches as follows: (i) both methods are designed with an online learning capability, which automatically learns on-the-fly to capture characteristics of promising crawling paths or destination web regions that lead to the most rewarding discoveries of online content relevant to the current crawling needs,; (ii) both methods are equipped with a randomized link visitation strategy where the likelihood of selecting a certain link from the candidate URL pool for visit in the next crawling step is probabilistically modulated by the estimated reward that the link can lead to the discovery of relevant content, (iii) both methods are equipped with some self-assessment and self-critic module that can autonomously determine the relevance of any harvested web page with respect to the crawling needs for selective output of crawling results and (iv) both methods extract text features and select the most reliable subset of candidate features to construct the predictive estimator on the crawling utility of a link.

The key differences between the two methods include (i) to forecast the utility of a link u_i, our method introduces a lightweighted learner $\Upsilon(wp(u_i), \Omega)$ that predicts the utility score before crawling the web content pointed to by u_i according to the information provided by the snippet associated with u_i. Such snippet is always available for search results returned by a typical search engine (such as Google and Yahoo). The snippet includes u_i's URL, a running head text of u_i and a brief piece of selected text from the search result associated with u_i. In comparison, the adaptive link learner introduced in the peer method only examines the text information encoded in a link's URL when performing the link utility estimation. The extended scope of textual information available from the snippet of a link allows our predicting function to be able to estimate the link utility more accurately (as confirmed by the experimental results), which results in better accuracy in targeted web crawling. (ii) In addition to the lightweighted learner $\Upsilon(wp(u_i), \Omega)$, our method also carries a more powerful web page utility assessment function $\Phi(wp, \Omega)$ that measures the utility of a web page wp with respect to the information need Ω after the web page is crawled. Based on the output of the function $\Phi(wp, \Omega)$, we dynamically retrain the function of $\Upsilon(wp(u_i), \Omega)$ so that the particular machine learning model selection and configuration are optimized on-the-fly according to all the cumulative quality assessment scores produced by the function $\Phi(wp, \Omega)$ since the beginning of the current crawling session. This novel two-tier online learning framework, which was not present in the peer method, allows our method to be able to train a more tailored and task-optimized link utility predictor $\Upsilon(wp(u_i), \Omega)$ in an autonomous fashion. (iii) The peer

method uses a specialized classifier for recognizing online searchable forms as a critic to judge the relevance of their online crawling results. This feedback mechanism is only applicable for the particular crawling needs to discover web pages of searchable forms as hidden entries to online databases. In contrast, when assessing the utility of a crawled online content web page, our method comprehensively considers the content words in the main body of an HTML file, words in the heading and subtitles of an HTML file and the anchor text embedded in an HTML file. Benefited by this more generic design of the self-assessment mechanism, which is coupled by a corresponding more advanced text feature extraction, selection and predictive modeling implementation, our adaptive crawler is able to detect online content meeting a much wider spectrum of users' needs for a more generic scope of applications. (iv) Our new crawler design explicitly models the confidence and reliability of its link utility prediction performance during the execution of online web crawling and considers such uncertainty during its planning of adaptive strategies for the current crawling task. This uncertainty modeling feature is missing from the design of the peer method. (v) Our crawler design explicitly models the time required to access a given web page for balancing the time spent between developing more carefully planned crawling strategies versus executing more operations of web page harvesting with less deliberated crawling strategies. Such feature is also absent from the design of the peer method.

To better understand the behavior characteristics of the two crawling methods, we conducted some further investigative analysis to comparatively examine crawling results obtained by each crawler for the two case studies reported in this article. First, we examined the overlap between crawling results harvested by the two crawlers using three metrics, which assess the amount of common content between two crawling result sets S_1 and S_2 on the levels of key words, documents and sites, respectively. For the key word-level overlap between S_1 and S_2, we first extracted the key words from each result set using the RAKE algorithm (Rose *et al*., 2012). Next, we ranked the two lists of key words according to each key word's term frequency-inverse document frequency value among its corresponding crawling result set. We then counted the number of common key words included in the top-ranked key word lists of S_1 and S_2 as the key word-level overlap between S_1 and S_2. Figure 4 reports results of the estimated key word-level overlap between the two sets of crawling results where both crawlers are trained with the full set of example search results. Figure 4a shows that the key word-level overlap remains roughly in the same value range (between 60 and 80%) even when estimated using different sizes of top-ranked key words. We then chose a representative window size (300) for the top-ranked key words and estimated the key word-level overlap throughout the entire crawling sessions. The results are shown in Figure 4b, according to which we can see the overlap between the two crawling result sets also stably remains within the same value range of 60 and 80% during the progression of the crawling processes. For the document-level overlap, we use the cosine distance to measure pairwise document similarity. We then estimated the overlap between S_1 and S_2 as the number of documents in S_1 and S_2 that can find at least one counterpart document in the other crawling result set with which their pairwise document similarity is no <90%, divided by the

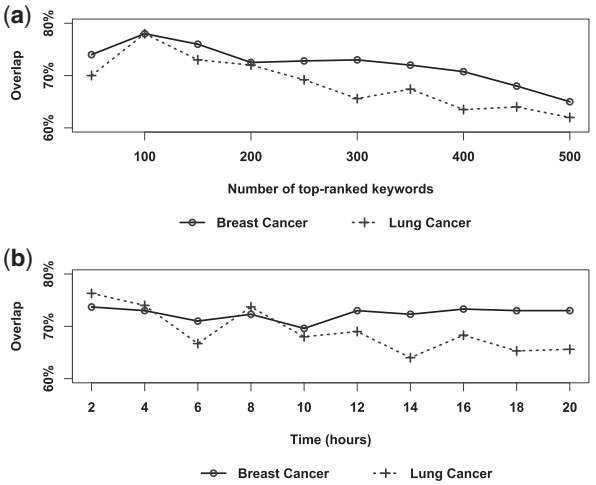

Fig. 4. Estimated key word-level overlap between crawling results obtained by our adaptive crawler and the peer crawler for the two case studies (using the full set of training examples in each crawling session)

total number of documents in S_1 and S_2 (without duplicate document removal). For the breast cancer study, the estimated document-level overlap is 69.5%; for the lung cancer case study, the estimated document-level overlap is 66.2%. Lastly, we also estimate the site-level overlap between the two crawling result sets. For the breast cancer study, the estimated site-level overlap is 33.7%; for the lung cancer case study, the estimated site-level overlap is 27.3%. We suspect the reason for the reduced numbers in the estimated overlap scores is because the site-level overlap estimate focuses on the physical URL positions of the acquired web pages rather than the actual content embedded in these web pages. Therefore, site-level overlap estimate presumably underestimates the actual content overlap between the two crawling result sets. Overall, according to the above analysis of overlapping content between the two crawling result sets, we acknowledge the benefit of simultaneously running both crawlers to acquire online content in a complementary fashion, as only a partial body of the harvested materials is commonly acquired by both crawling methods.

To understand the content similarities and differences between online materials harvested by the two crawlers, we further generated for each case study three lists of top-ranked key words, including key words that appear in both sets of crawling results, referred to as 'common keywords' and key words that appear only in crawling results acquired by one of the two crawlers, referred to as 'unique keywords'. By manually reading through these three lists of top-ranked key words saliently embodied in the crawling results, we empirically notice that the common key words cover overwhelmingly generic non-technical terms associated with cancer, such as 'cancer', 'therapy', 'surgery', 'treatment', 'tumor' and 'diagnosis'. For key words unique to the peer crawling method, they primarily include technical terms associated with cancer, such as 'invasive breast cancer', 'ultrasound', 'discharge', 'vivo', 'ducts', 'lobules', 'mortality', 'ductal carcinoma' and 'inhibitors'. Finally, key words unique to our adaptive crawler are mostly associated with lifestyle aspects of a patient, and fewer key words describing the disease itself. In summary,

the peer method appears to collect online content richer in medical terms, whereas the adaptive crawler appears to be able to harvest cancer patient stories that expose more abundant details in lifestyle and emotionally related matters. It is noted the latter type of crawling results acquired by our crawler agrees better with the true information collection needs of e-health researchers in both case studies—one about gathering life stories of breast cancer patients and the other about smoking of lung cancer patients. We speculate this better alignment of crawling results with researchers' information acquisition needs is achieved by the more content-sensitive adaptive crawling mechanism of our new crawler, benefited by its more advanced selective online content detection and acquisition algorithm proposed in this article.

7 CONCLUSION

We propose a user-oriented web crawler that can adaptively acquire social media content on the Internet to meet the specific online data source acquisition needs of the end user. We evaluated the new crawler in the context of cancer epidemiological research with two case studies. Experimental results show that the new crawler can substantially accelerate the online user-generated content acquisition efforts for cancer researchers than using the existing state-of-the-art adaptive web crawling technology.

ACKNOWLEDGEMENTS

This manuscript has been authored by UT-Battelle, LLC, under Contract No. DE-AC05 00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. This study was approved by the Oak Ridge site-wide Internal Review Board.

Funding: National Cancer Institute (1R01CA170508-01).

Conflict of Interest: none declared.

REFERENCES

ACS. (2013). American Cancer Society: Cancer Survivors Network, Atlanta, GA, USA.

Aggarwal,C.C. (2002) Collaborative crawling: mining user experiences for topical resource discovery. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD'02, pp. 423–428.

Aggarwal,C.C. *et al.* (2001) Intelligent crawling on the World Wide Web with arbitrary predicates. In: *Proceedings of the 10th International Conference on World Wide Web*. WWW'01. pp. 96–105.

Ahlers,D. and Boll,S. (2009) Adaptive geospatially focused crawling. In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management*. CIKM'09. pp. 445–454.

Almpanidis,G. and Kotropoulos,C. (2005) Combining text and link analysis for focused crawling. In: *Proceedings of the Third International Conference on Advances in Pattern Recognition - Volume Part I*. ICAPR'05. Springer-Verlag, Berlin, Heidelberg, pp. 278–287.

Almpanidis,G. *et al.* (2005) Focused crawling using latent semantic indexing: an application for vertical search engines. In: *Proceedings of the 9th European Conference on Research and Advanced Technology for Digital Libraries*. ECDL'05. Springer-Verlag, Berlin, Heidelberg, pp. 402–413.

Babaria,R. *et al.* (2007) Focused crawling with scalable ordinal regression solvers. In: *Proceedings of the 24th international conference on Machine learning*. ICML'07. pp. 57–64.

Badia,A. *et al.* (2006) Focused crawling: experiences in a real world project. In: *Proceedings of the 15th International Conference on World Wide Web*. WWW'06. pp. 1043–1044.

Barbosa,L. and Freire,J. (2007) An adaptive crawler for locating hidden web entry points. In: *Proceedings of the 16th International Conference on World Wide Web*. WWW'07. ACM, New York, NY, pp. 441–450.

Batsakis,S. *et al.* (2009) Improving the performance of focused web crawlers. *Data Knowl. Eng.*, **68**, 1001–1013.

Chakrabarti,S. *et al.* (2002) Accelerated focused crawling through online relevance feedback. In: *Proceedings of the 11th international conference on World Wide Web*. WWW'02. pp. 148–159.

Chen,Z. *et al.* (2009) A cross-language focused crawling algorithm based on multiple relevance prediction strategies. *Comput. Math. Appl.*, **57**, 1057–1072.

Chung,C. and Clarke,C.L.A. (2002) Topic-oriented collaborative crawling. In: *Proceedings of the Eleventh International Conference on Information and Knowledge Management*. CIKM'02. pp. 34–42.

de Assis,G.T. *et al.* (2008) The impact of term selection in genre-aware focused crawling. In: *Proceedings of the 2008 ACM symposium on Applied Computing*. SAC'08. pp. 1158–1163.

Dey,M. *et al.* (2010) Focused web crawling: a framework for crawling of country based financial data. In: *Proc. IEEE International Conference on Information and Financial Engineering (ICIFE)*. pp. 409–412.

Fu,T. *et al.* (2012) Sentimental spidering: leveraging opinion information in focused crawlers. *ACM Trans. Inf. Syst.*, **30**, 24:1–24:30.

Furuse,K. *et al.* (2011) An extended method for finding related web pages with focused crawling techniques. In: *Proceedings of the 15th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems - Volume Part II*. KES'11. Springer-Verlag, Berlin, Heidelberg, pp. 21–30.

Gao,W. *et al.* (2006) Geographically focused collaborative crawling. In: *Proceedings of the 15th International Conference on World Wide Web*. WWW'06. pp. 287–296.

Guan,Z. *et al.* (2008) Guide focused crawler efficiently and effectively using on-line topical importance estimation. In: *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR'08. pp. 757–758.

Hall,M. *et al.* (2009) The weka data mining software: an update. *ACM SIGKDD Exp. Newslett.*, **11**, 10–18.

Kleinberg,J.M. (1999) Authoritative sources in a hyperlinked environment. *J. ACM*, **46**, 604–632.

Kohlschutter,C. (2011) The Boilerpipe library: boilerplate removal and fulltext extraction from html pages. In: *Google Code Base*.

Liu,H. and Milios,E. (2012) Probabilistic models for focused web crawling. *Comput. Intell.*, **28**, 289–328.

Menczer,F. *et al.* (2004) Topical web crawlers: evaluating adaptive algorithms. *ACM Trans. Internet Technol.*, **4**, 378–419.

Micarelli,A. and Gasparetti,F. (2007) *The Adaptive Web: Adaptive Focused Crawling*. Springer-Verlag, Berlin, Heidelberg, pp. 231–262.

Pant,G. and Srinivasan,P. (2005) Learning to crawl: comparing classification schemes. *ACM Trans. Inf. Syst.*, **23**, 430–462.

Pant,G. and Srinivasan,P. (2006) Link contexts in classifier-guided topical crawlers. *IEEE Trans. Knowl. Data Eng.*, **18**, 107–122.

Rose,S.J. *et al.* (2012) Rapid automatic keyword extraction for information retrieval and analysis. US patent **8,131,735 B2**. July 2, 2009.

Zhuang,Z. *et al.* (2005) What's there and what's not?: focused crawling for missing documents in digital libraries. In: *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital libraries*. JCDL'05. pp. 301–310.