COST-SENSITIVE TOPICAL DATA ACQUISITION FROM THE WEB

Mahdi Naghibi[1], Reza Anvari[1], Ali Forghani[1] and Behrouz Minaei[2]

[1]Malek-Ashtar University of Technology, Lavizan, Tehran, Iran
[2]Departement of Computer Engineering, Iran University of Science and Technology, Narmak, Tehran, Iran

ABSTRACT

The cost of acquiring training data instances for induction of data mining models is one of the main concerns in real-world problems. The web is a comprehensive source for many types of data which can be used for data mining tasks. But the distributed and dynamic nature of web dictates the use of solutions which can handle these characteristics. In this paper, we introduce an automatic method for topical data acquisition from the web. We propose a new type of topical crawlers that use a hybrid link context extraction method for topical crawling to acquire on-topic web pages with minimum bandwidth usage and with the lowest cost. The new link context extraction method which is called Block Text Window (BTW), combines a text window method with a block-based method and overcomes challenges of each of these methods using the advantages of the other one. Experimental results show the predominance of BTW in comparison with state of the art automatic topical web data acquisition methods based on standard metrics.

KEYWORDS

Cost-Sensitive Learning, Data acquisition, Topical Crawler, Link Context, Web Data Mining

1. INTRODUCTION

Real-world data mining problems have various challenges during their processes, and different types of cost are associated with each step of solutions suggested for these processes from the start to the end[1]–[3]. Utility or cost based data mining considers these distinct costs and compare learning methods based on more comprehensive metrics. This approach considers three main steps, particularly for the classification task and each step is associated with its related cost during the process. These steps are data acquisition, model induction, and application of the induced model to classify new data [4]. The cost of data acquisition has been neglected more than the others in many cost-sensitive data mining and classification researches. We will consider the cost of data acquisition from the web as efficient use of available bandwidth for topical crawlers. The web is one of the most comprehensive sources of information for many data mining tasks such as classification and clustering. It contains various kinds of data including text, image, video, etc. However, for acquiring these data from the big, distributed, heterogeneous and dynamic web, we need methods that automatically surf the web pages with efficient use of available bandwidth and collect desired data regarding predefined target topics. Topical web crawlers are strong tools to cope with this challenge. They start from some seed URLs, download their target pages and extract the links inside them; then they assign scores to URLs according to the likelihood of the links to access pages in the target subject. Fig. 1 shows the process of topical data acquisition from the web using topical crawlers.

The main concentration of proposed methods for topical crawlers is on making them able to predict the relevancy of target pages of given links. Link context is the words that exist around a link inside the page [5]. It has very good clues for guiding of crawlers. However, determining the link context of a hyperlink is a hard task for a crawler, because finding the barriers between link contexts of different links needs consideration of many visual cues inside the web page, which are designed for the human user to find the related parts of the page, not for machine agents.

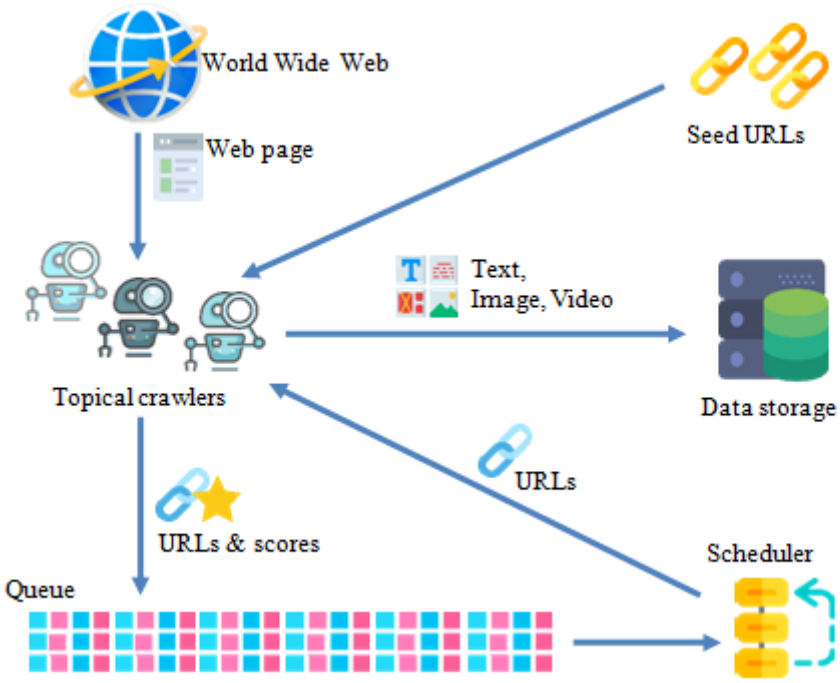


Figure 1. Illustration of topical data acquisition process form the web

We propose Block Text Window (BTW), a hybrid link context extraction method for topical web crawling. This method is based on the assumption that utilizing the Vision-Based Page Segmentation (VIPS) algorithm [6] for page segmentation can improve the accuracy of link context extraction and since this algorithm has some shortages in extracting page blocks accurately, BTW uses text window method [5] on the text of page blocks to extract link contexts more correctly. We have done empirical studies on the performance of the proposed method and compared it with the most effective existing approaches using standard evaluation metrics. The organization of this paper is as follows: section 2 discuses about the related fields of research, section 3 introduces the proposed method, in section 4 the experimental study is presented, and section 5 belongs to the conclusions.

2. RELATED WORKS

Based on the scope of this paper we investigate three interrelated fields: cost-sensitive data acquisition, topical crawling, and link context extraction methods.

2.1. Cost-Sensitive Data Acquisition

A considerable amount of literature has been published on cost-sensitive data acquisition in related domains such as active learning[7], cost-sensitive feature selection, and feature extraction[1], [8]. The active learning method in [9] considers the cost of labelling instances for the proposed recommender system. The authors of [10] used a combination of deep and active learning for image classification and tried to minimize the cost of assigning labels to the

instances. Recently in [11] the researchers proposed a combination of classifier chains and penalized logistic regression which takes into account features cost. Liu et al. proposed a cost-sensitive feature selection method for imbalanced class problems [12].But there are a few numbers of researches that consider the cost of collecting cases. Weiss et al. [13] proposed a cost and utility-based evaluation framework that considers all steps of a data mining process. They refer to the cost of cases as the cost associated with acquiring complete training examples. Based on the definitions of [13], the induced model$A$ has more utility than the induced model B if and only if:

$$Cost_{total}(A) < Cost_{total}(B) \tag{1}$$

The $Cost_{total}$ is the sum of all costs during different stages of classification problem and can be computed by:

$$Cost_{total}(M) = Cost_{data\_asquisition}(M) + Cost_{model\_induction}(M) + Cost_{misclassification\&model\_application}(M) \tag{2}$$

Where the cost of data acquisition includes the cost of collecting instances, features (tests) and labels. Cost of model induction includes computational costs. The last cost in (2) describes the misclassification errors and computational cost during the utilization process of the models. In the current research, we focus on the cost of collecting web page instances form the web which can be considered as effective bandwidth usage by topical crawlers.

2.2. Topical Crawling Methods

The authors of [14] evaluated the feasibility of utilizing topical crawlers for building event collections on the existing web archives to reduce the time of the crawling process. They reported good results mostly for events that happened in the past. AbdulNabi and Premchand[15] proposed EPOW (Effective Performance of WebCrawler) architecture for efficient acquisition of user needed information. They considered parallelization policy and made an optimized system which can download a large number of pages per second. The authors of [16] proposed a statistical hypothesis based learning mechanism for learning the topical crawling speed in different network environments. The method utilized to maintain the politeness of crawling on a fluctuated network bandwidth.

Researchers of [17] used a query generator to utilize the previously indexed pages by Google to reduce the cost of crawling and building a cost-efficient topical crawler for collecting academic contents. In [18] the authors presented a crawling and information extraction system for e-commerce sites. This system crawls many e-commerce sites and accumulate their content. It displays the extracted data in a site which is called site summary. The authors of [19]proposed an adaptive topical crawler that uses interactive and automatic adaption approaches for online learning of hyperlink selection policy, form previously crawled pages. Han et al. utilized reinforcement learning for topical web crawling [20]. They formulated the problem as a Markov decision process and proposed a new representation of states and actions which considers both content information and the link structure. The Hidden Markov Model (HMM) is used in [21] to model the probability of guiding links to on-topic pages. To make this model, the user must supply the appropriate feedbacks for the crawled pages.

In a recent paper, Farag et al. [22] proposed a topical crawler for automatic event tracking and archiving. An architecture for effective migrating parallel web crawling approach with topical and incremental crawling strategy is introduced in[23].The main advantage of this migration is that the analysis portion of the crawling process is done locally at the residence of data rather than

inside the search engine repository. This reduces network traffic and improves the efficiency of the data acquisition process. In the next part, we categorize topical crawling methods according to the strategy they use for link context extraction.

2.3. Link Context Extraction Methods

We divide link context extraction methods into four categories: considering page text and link text as the link context, text window method, DOM-based methods, and block-based methods. Fig. 2 illustrates the link context extraction methods by typical examples. In the following, we explain these methods.

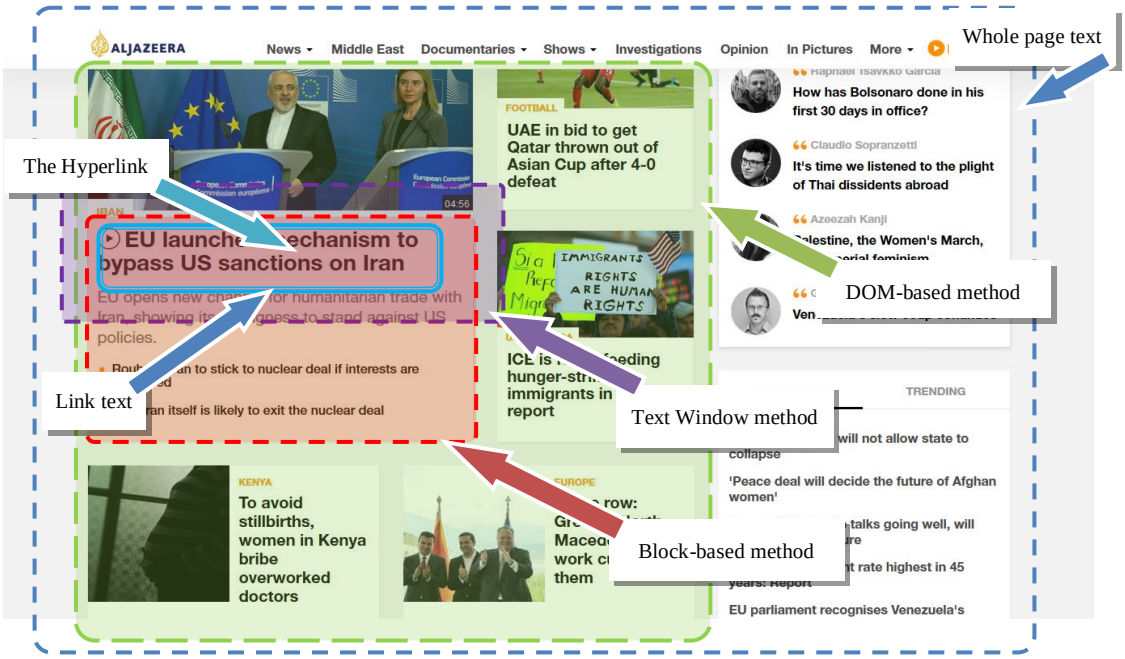


Figure 2. Link context extraction methods including page text, link text, a DOM-based method, a Text Window method, and an appropriate block-based method

2.3.1. Page and Link Text

The easiest way to extract the link context is to consider the entire text of a web page as the link context of all page links. The proposed crawler in [24] which is called Fish Search, utilized this method to evaluate the links of a web page and thus all the links on the page have an equal score. The best-first method which is a modified version of this algorithm combines the page text score with link text score.This combination can be done based on the following formula:

$$link\_score = \beta \times Relevancy(page\_text) + (1 - \beta) \times Relevancy(link\_context) \quad (3)$$

Where $link\_score$ is the score of the link within the page, $page_text$is the entire page's text, and $link_context$is the extracted link context of the link which is the link text, in a typical best-first method. $Relevancy$ Function is responsible for computing the score of a given text, based on the target topic.

2.3.2. Text Window Method

This effective method, considers a number of W words in the vicinity of a link as the link context [5]. It uses a symmetric window around the link and extracts a number of $W/2$ words before and $W/2$ after the text of the link. Also, the text of the link is put inside the window. This method has an unresolved challenge: It does not know the optimal either near-optimal number of terms of link context around a hyperlink.

2.3.3. DOM Methods

Document Object Model (DOM), considers the page as a tree based on the hierarchy of its HTML tags. This model is used for the link context extraction in some topical crawling methods. In [25] DOM of the page is utilized for scoring the words in different parts of the DOM tree, based on their distances from the link. They assumed that closer words to link could make a better link context. DOM is also used in [5] for link context extraction of links which have few words in their link text. All of these methods have a severe deficiency because the HTML which is the base of DOM, must be interpreted by a web browser to determine the visual structure of the page and its raw format is not supposed to be used for tasks like link context extraction.

2.3.4. Block Methods

Page block consists of related partitions of the web page, and a page segmentation process extracts these blocks from the page. The block-based methods consider the text of the block as the link context of the links exist inside the block. Based on the results of [26] we expect block-based methods extract link context of the links more accurately than the other methods. The VIPS algorithm [6] performs some kind of rendering on the HTML of the page, similar to web browsers, and utilizes many visual cues inside the page including colors, font sizes, and other styles that make the visual structure of the page and guides the algorithm to locate the distinct blocks of the page. This algorithm extracts the vision-based structure of the web page using the combination of DOM and the visual cues of the page. The process of segmentation is performed in three phases which includes extraction of the block, detection of the separator, and construction of the content structure. These three continuous phases make a round in the process. The VIPS works in a top-down fashion. First, the algorithm segments the page into many big blocks and make a hierarchical structure from these blocks. Then the same segmentation procedure is performed recursively inside big blocks until enough small page blocks with an appropriate degree of coherence (DoC) values which are above a predefined threshold PDoC are constructed. After the segmentation process, all the leaf nodes are extracted as blocks. The VIPS is one of the best algorithms for extraction of page blocks. We utilized this algorithm in our hybrid method for link context extraction.

3. COST-SENSITIVE WEB DATA ACQUISITION

The concentration of this research is on proposing a method capable of collecting page instances form the web with minimum cost. To achieve this goal, we use a novel topical crawling method and will evaluate its performance based on standard metrics. The proposed method is inspired from [27]. Their experimental studies in the field of web information retrieval show that best results are achieved when they used a combination of VIPS algorithm and text window method for page segmentation. In the following of this section, we explain our incentive to use this hybrid method for link context extraction based on empirical studies of [27] and reported results in the field of topical web crawling. Also, the proposed hybrid link context extraction method is discussed in details.

3.1. Challenges of Solely Using Text Window and VIPS for Link Context Extraction

In this subsection, we investigate challenges of solely using text window method and VIPS algorithm for link context extraction and clarify the intent of proposing the hybrid method which combines both.

3.1.1. Challenges of Text Window Method

Text window method has two major challenges which can't resolve solely:

- The position of link context can't be determined efficiently.
- Optimal (even near-optimal) size of the text window is unknown.

The process of determining link context in regular text window methods begins by extraction of the whole page text from its DOM, based on a depth-first manner. Then the method specifies the closest words to the link. But this algorithm has two problems that prevent it from locating right link contexts. The first problem is originated from the reliance of this method solely on the raw DOM of the page. The HTML of the page need to be rendered to show the predefined visual view of it, and since naïve depth-first strategies are not conscious of this visual layout, they can't guide the method to correct link contexts. The second problem comes from symmetry of text window in this method which adds wrong and redundant words to link context. Because in many cases the correct link context appears only after the text of the link or before it and doesn't exist in both positions.

The second major challenge in text window method is the unclear size of a text window for appropriate link context extraction. This challenge comes from the diversity of layout and structure design of web pages since web has an open nature and layout designer use elements of a page based on the taste of their users which results in various page styles on the web. For example, a news page has a different structurein comparison witha shopping page.

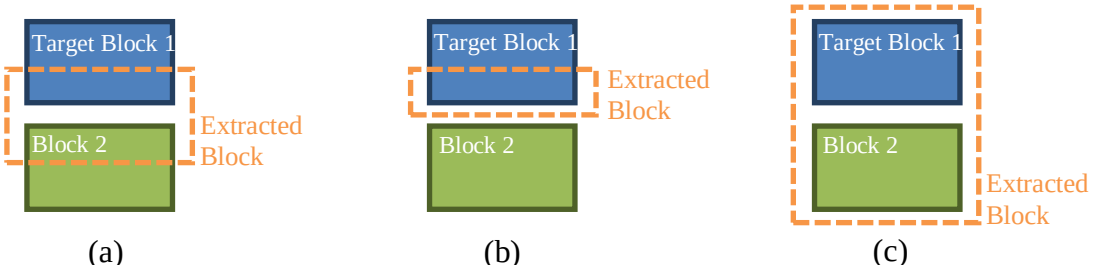


Figure 3. Illustration of three kinds of wrongly extracted blocks. The higher boxes are true target blocks and the dashed lines show the extracted blocks. (a) The target block segmented more than it should and contains parts that belong to block 2. (b) The target block segmented more than it should and all of its parts belong to it. (c) The target blocks segmented less than it should and contains parts that belongs to block 2.

3.1.2. Challenges of VIPS Algorithm

When VIPS algorithm is utilized in a link context extraction method, the method will face two main challenges:

- Wrongly extracted blocks may contain noisy words.
- Blocks with long texts may involve noisy words.

These two challenges can result in the wrong prediction of the link's target page relevancy. Although VIPS has many advantages, in some situations it may work wrong and may produce three kinds of wrongly extracted blocks. Fig. 3 illustrates these kinds of blocks which include:

- Blocks segmented more than they should and contain some parts that truly belong to other blocks (Fig. 3.a).
- Blocks segmented more than they should and all of their parts truly belong to them (Fig. 3.b).
- Blocks segmented less than they should and contain some parts that truly belong to other blocks (Fig. 3.c).

If a block is segmented more than it should, it is rare that it contains parts that truly belong to other blocks. Since its DoC with its real parts which it doesn't contain now, likely to be much higher than its current DoC. In other words, it is more likely that a block which is partitioned more than it should, contains its own parts rather than the parts of the others. In cases that the block is segmented more than it should but doesn't contain parts of other blocks, then it doesn't contain noisy words and the extracted link context based on its content is fragmental but still clean and doesn't mislead us. However, our experimental observations show that third kind of wrongly extracted blocks that segmented less than they should and contain parts that truly belong to other blocks produced many times by VIPS algorithm. This kind of blocks contains a considerable number of noisy words and if they get included in link contexts of the block hyperlinks can result in the wrong prediction of target pages relevancies.

When we use VIPS algorithm, In addition to wrongly extracted blocks problems, we encounter a set of blocks with highly variable lengths. The reported statistic in [27] shows that 19% of blocks are larger than 200 words. If the whole text of long blocks is considered as its contained hyperlinks' contexts, then extracted link contexts involve noisy words probably. Intuitively, we know most of the link contexts are not that much long. Thus the variable length problem still exists even if we extract link contexts of hyperlinks based on the text of blocks that containing them.

3.2. Block Text Window Method

We propose a hybrid link context extraction method that combines the VIPS page segmentation algorithm with Text window method. This hybrid method which is called block text window (BTW) employs advantages of each of these combined methods to overcome challenges of the other one. Like CombPS [27] our BTW method has two steps:

Step 1: block extraction from web pages using VIPS algorithm. In this step, the given page is passed to the VIPS algorithm as its input. VIPS constructs vision-based content structure and all of the leaf nodes of this structure are considered as the page's blocks and will be sent to the next step as inputs.

Step 2: link context extraction from blocks using text window method. The text window method is applied to hyperlinks contained in each input block, and link contexts are extracted from the text of blocks. Unlike CombPS we do not segment a block based on a set of overlapping windows. For each hyperlink, text windows are considered as $W/2$words before and $W/2$ the position of hyperlink's text in its surrounding block.

Algorithm: Vision Based Page Segmentation (VIPS)

```
input:root: root of DOM tree
output: cStructure: hierarchical vision-based content structure of the page
method: Vision-Based-Page-Segmentator(root)
pool ← Raw-Blocks-Extractor(root);
separators ← Separators-Detector(pool);
cStructure ← Content-Structure-Constructor(pool, separators);
returncStructure;
end method
```

Figure 4. Pseudocode of the VIPS algorithm

Algorithm: Block Text Window (BTW)

```
input:link: a hyperlink of the page
cStructure: hierarchical vision-based content structure
T: the size of the text window
output:linkContext: link context of the input hyperlink
method: BTW- Link-Context-Extractor(link,cStructure, T)
for eachleaf∈cStructure.leafsdo //leaf is a block
iflink∈ leaf.linksthen
        (textWinBefore, textWinAfter) ← Text-Win(link .text,leaf.text ,T)
linkContext←textWinBefore⊕link.text⊕textWinAfter
break for
end if
end for
retrunlinkContext
end method
```

Figure 5. Pseudocode of the proposed block text window (BTW) method

Figures4 and 5 show the pseudocode of the VIPS algorithm and the proposed BTW method. The first step is the base of our hybrid link context extraction method. Extracted blocks by VIPS algorithm approximately determine the appropriate position of link contexts, where for each hyperlink is the text of its containing block. Also, the length of the majority of blocks is less than 50 words [27]that means the VIPS algorithm can determine the size of link context in addition to its position for many hyperlinks.

In the second step,a fine-tuning happens. We intend in this step to remove noisy words which truly belong to blocks other than the ones they are contained. We are aware of the disadvantages of applying text window to block text for link context extraction which drops some truly related parts of blocks and makes imperfect some of the link contexts. But we believe that in a trade-off between keeping noisy words and dropping some related parts of blocks, the second one has more benefits and can improve the quality of extracted link contexts and topical crawling performance. Experimental results of this paper are witnesses for this assertion. It should be pointed out that the second step does not change the link context of hyperlinks that contained in blocks with shorter text length than the applied text window size.

For conclusion we can say that the proposed hybrid method by utilizing the VIPS algorithm for page segmentation has the following advantages in the first step:

- It can locate the proper position of link contexts.
- For a majority of page blocks that are smaller than the suitable size of a text window (which is about 40 words [5]), the proper size of link contexts is determined.

In the second step, applying a text window on extracted blocks has the following benefits:

- Noisy words of many wrongly extracted blocks are prevented from inclusion in link contexts.
- Link context sizes are normalized and noisy words which likely exist in blocks with long texts are dropped from link contexts.

In the next section, we empirically evaluate our hybrid link context extraction approach and compare its performance with other methods.

4. EXPERIMENTAL STUDY

In this part first, we explain the metrics used for comparison of the methods. Second, the settings of experiments are described. Then the results and their analysis is explained based on the metrics.

4.1. Evaluation Metrics

Since the main cost in our topical data acquisition problem comes from the bandwidth usage, we use standard metrics in topical crawling that can evaluate different methods based on their success on collecting on-topic page with minimum waste of this resource. Harvest Rate and Target Recall which are two standard metrics have been used to evaluate topical crawling methods and their link context extraction strategy. Many papers such as [21] and[28] have used these metrics for evaluation of topical crawling methods.

4.1.1. Harvest Rate

From the beginning of the process to the current moment for t crawled web pages the harvest rate $H(t)$ is calculated using (4):

$$H(t) = \frac{1}{t}\sum_{i=1}^{t} r_i \tag{4}$$

Where r_i the relevancy of is crawled page i and is calculated using an evaluator classifier. The settings of evaluator classifiers are explained in the next section.

4.1.2. Target Recall

For t crawled web pages from the start of the process until now, the target recall $R(t)$ is computed using (5):

$$R(t) = \frac{|C(t) \cap T|}{|T|} \tag{5}$$

Where $C(t)$the set of is fetched web pages from the start to page t,T is the set of target pages and $|T|$indicates the number of pages in this set.

4.2. Experimental Settings

4.2.1. Evaluated Methods

We used[29]codes for implementation of topical crawlers. Four distinct methods have been compared in this paper which includes: best first method, text window method, a block-based method, and our proposed hybrid method. The modified versions of text window and best first methods that use page text relevancy in addition to link context relevancy for calculation of final scores of links have higher evaluation results in comparison with their regular versions which don't use page text, based on [21] and [5]. Thus these modified versions of methods are compared with the proposed method. Equation (3) is utilized for the combination of scores. For the best first method, link text is used as $link_context$in (3), and the text of the window is used as this variable for the text window method.

In the block-based method, VIPS algorithm is used for page segmentation and performance of two variants of this algorithm is evaluated. In one variant, only block texts is used for link context extraction and scoring the links. In the other variant, the combination of link context relevancy and page text relevancy is used for calculation of link scores. Based on the reported experiments in [30] the block-based methods which uses VIPS algorithm have the state of the art results in comparison with other best-first methods. The proposed hybrid method extracts link contexts as described in section 3. Same as the block-based method we evaluated two versions of BTW. Equation (3) is utilized for the combination of the relevancy of page text and link context in corresponding BTW and block-based methods. The value 0.25 is used for the β parameter in (3) based on reports of [5], For combinational variants of methods which use link text and page text for scoring the links. The size of the window in the text window and BTW methods is considered 10, 20 and 40 words similar to sizes used in [5]. In VIPS the PDoC parameter is considered 0.6 based on [27].The codes in[31]are used for implementation of VIPS algorithm.

Table 1. Descriptions of selected topics from DMOZ

Topic	**Depth**	**URLs**	**Root URL**
England football	1	250	dmoz-odp.org/Sports/Soccer/UEFA/England/
algorithm	1	120	dmoz-odp.org/Computers/Algorithms/
graph theory	2	80	dmoz-odp.org/Science/Math/Combinatorics/Graph_Theory/
java	1	410	dmoz-odp.org/Computers/Programming/Languages/Java/
Olympics	3	330	dmoz-odp.org/Sports/Events/Olympics/

Table 2. Seed URLs and number of target pages for topics

Topic	Target Pages	Seed URL
England football	80	dir.yahoo.com/Regional/Countries/United_Kingdom/Recreation_and_Sports/Sports/Soccer/ (from archive.org) www.ask.com/web?q=soccer+england
algorithm	40	www.ask.com/web?q=algorithms www.google.com/search?q=algorithms en.wikipedia.org/wiki/Algorithm
graph theory	30	www.ask.com/web?q=graph+theory www.google.com/search?q=graph+theory en.wikipedia.org/wiki/Graph_theory
java	140	dir.yahoo.com/Computers_and_Internet/Programming_and_Development/Languages/Java/(from archive.org) www.ask.com/web?q=java+programming
Olympics	110	dir.yahoo.com/Recreation/Sports/Events/International_Games/Olympic_Games/(from archive.org) www.ask.com/web?q=olympic+games

4.2.2. Selected Topics

We have selected five topics from DMOZ [32] for evaluation and comparison of methods. DMOZ is a manually edited open content directory of the web links which use a hierarchical structure for organizing URLs in many topics. Chosen target topics include England football, algorithm, graph theory, java, and Olympics. The URLs of these topics are downloaded based on the hierarchical structure of DMOZ and sets of URLs are built for topics. Table 1 shows the root URLs and number of downloaded URLs for selected topics. The depth parameter shows the depth of subcategories of a selected topic for inclusion in the set of URLs; For example, depth 2 means that in addition to URLs of selected topic which we consider as level 0 in hierarchical structure of DMOZ, the URLs of subcategories with 1 and 2 higher levels are included in URL set,.

In each set, some URLs are selected and the target set is made for calculating target recall of the methods. The average results of methods are computed for selected topics based on the evaluation metrics. Table 2. Shows the seed URLs and number of target pages for selected topics.

4.2.3. Conductor and Evaluator Classifiers

We have used two kinds of classifiers during the topical crawling and evaluation process. The evaluator classifier calculates the relevancy of downloaded pages during the crawling. It computes the r_i parameter in (4). An evaluator classifier is trained for each selected topic. The conductor classifiers is utilized by topical crawlers to compute the relevancy of extracted link contexts to the target topic. The URLs of target sets of topics are excluded from the training set of conductor classifiers to make a fair process.

Multi-layer perceptron neural networks[33] are used as conductor and evaluator classifiers. The Vector Space Model (VSM) [34] is used for representation of pages. To represent pages in VSM, first the text of pages is extracted using HTML parser[35]. Then the stop words are eliminate from the text, and other words are rooted in Porter algorithm[36]. The weight of words is also determined by the TFIDF method[37]. Each of the dimensions of vector space is one of the rooted words and is considered as a feature. Due to the high number of words, it is not practicable to train MLP classifiers with these dimensions of the feature space. One solution is to use feature selection methods. Also, if the feature selection is done in an appropriate manner, it will also increase the efficiency of the classification[38].

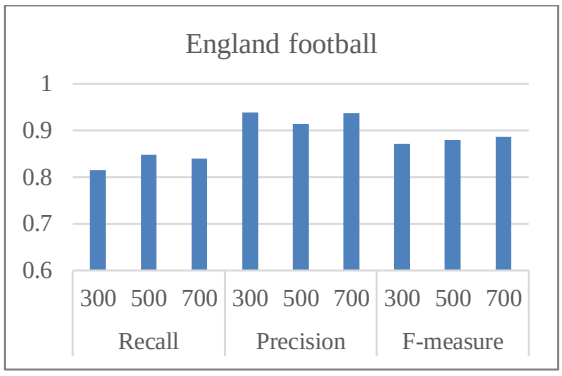
According to [38], the information gain (IG) method has been used to reduce the dimensions of the feature space. If $\{c_i\}_{i=1}^{m}$ be the set of topics in the target space then the information gain of word$w$ will be computed using (6):

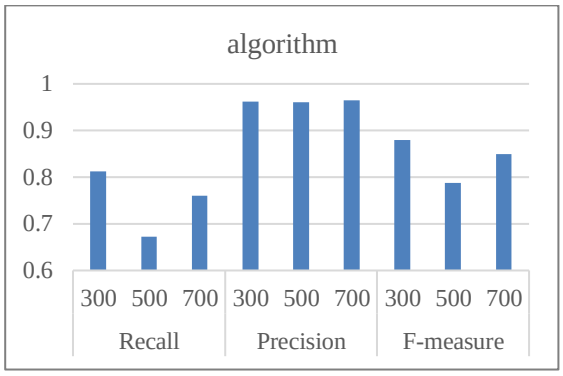
$$IG(w) = -\sum_{i=1}^{m} P_r(c_i)\log P_r(c_i) + P_r(w)\sum_{i=1}^{m} P_r(c_i|w)\log P_r(c_i|w) + P_r(\overline{w})\sum_{i=1}^{m} P_r(c_i|\overline{w})\log P_r(c_i|\overline{w}) \quad (6)$$

Where P_r is the probability function.Because of the significant impact of the number of selected features on the efficiency of classification, a set of experiments has been carried out to determine the appropriate number of features for classifiers of each topic. Figure 6 shows the efficiency of trained classifiers based on recall, precision and F-measure criteria. These criteria are calculated based on the 10-fold cross validation.

All MLP classifiers have three layers (with a single hidden layer). The number of hidden layer neurons is 20% of the number of input layer neurons. The target pages of the set of URLs for a selected topic are used as positive learning samples for classifiers and twice the number of these instances the randomly downloaded pages from other topics of DMOZ are used as negative learning instances. The classifiers are Implemented using some of the open source weak packages [39] and default settings of weaker used for training.

As charts show, all of these classifiers have high precision. F-measure is a combination of recall and precision, which alone represents the actual performance of a classifier.For 3 topics, the use of 500 words as inputs to classifiers leads to optimal performance. However, for graph theory, the use of 700 features has a remarkable advantage over the use of 500 attributes for the classification. In algorithm topic, the increase in the number of attributes reduces the efficiency of classifier, and the use of 300 input features has the highest F-measure for classification.





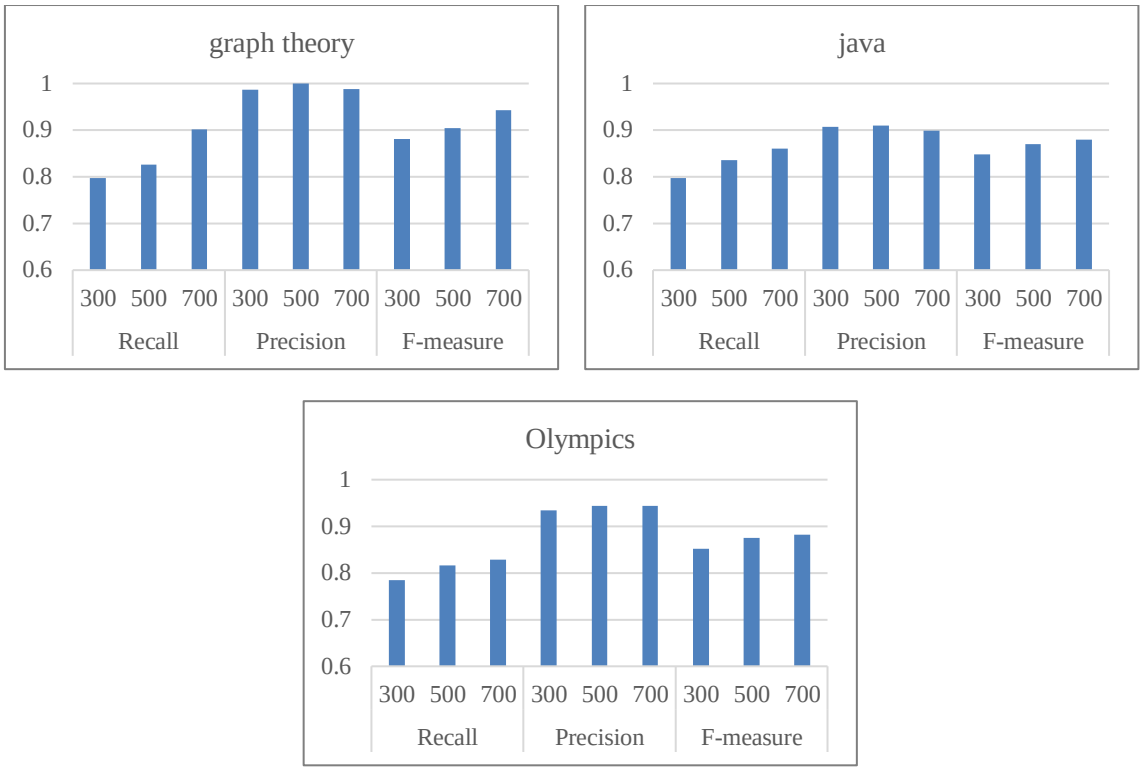


Figure 6. The efficiency of conductor classifiers based on different criteria for selected topics. The horizontal axis shows the number of selected features

Table 3. The specifications of conductor classifiers for selected topics

Topic	Positive Samples	Total Samples	Input Features	Recall	Precision	F-measure
England football	250	750	500	0.85	0.91	0.88
algorithm	120	350	300	0.81	0.96	0.88
graph theory	80	250	700	0.90	0.99	0.94
java	410	1200	500	0.84	0.91	0.87
Olympics	330	1000	500	0.82	0.94	0.88

Table 3 shows the specifications of conductor classifiers for topics. Criteria represent the high efficiency of conductor classifiers. Thus, the decisions made by the conductor classifiers will be trusted.

4.3. Implementation Results

Long term crawling has been done to achieve more stable results for more accurate comparison of topical crawling methods. Thirty thousand of web pages have been crawled for each method regarding different topics.

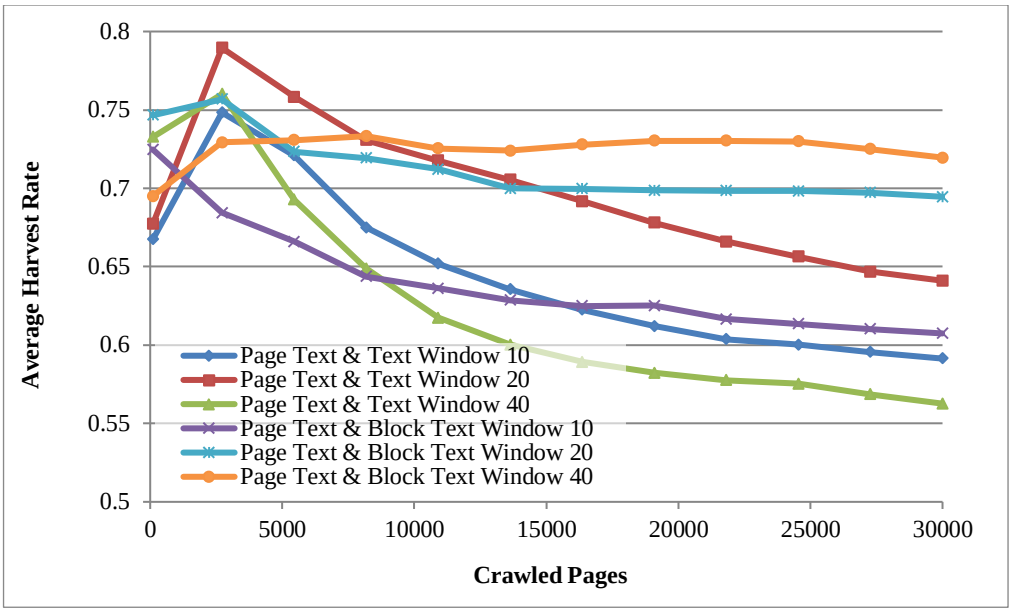


Figure 7. Harvest rate of BTW and text window methods

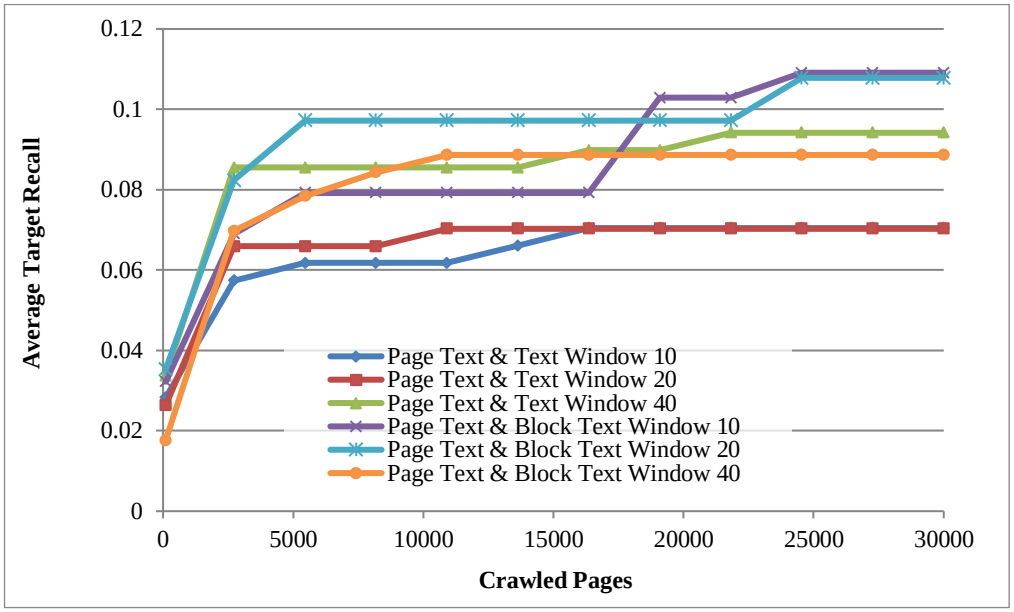


Figure 8. Target recall of BTW and text window methods

Figs. 7 and 8 illustrate the harvest rate and target recall of text window and BTW methods for different sizes of windows. As it is seen in Fig. 7 harvest rate of BTW methods is more than the harvest rate of text window methods with the same size of window at the end of the crawling process; BTW 10 has better harvest rate than text window 10, BTW 20 has better harvest rate than text window 20, and BTW 40 has much better harvest rate than text window 40. The Superiority of BTW over text window method is increased for larger window sizes. Same results are hold based on target recall metric except for window size 40 where target recall of BTW method is less than the text window method but is very close to it. These results show that using

BTW method can result in better topical crawling performance in comparison with the text window method. The harvest rate metric alone can represent the performance of a topical crawler. This metric has some advantages over target recall because harvest rate evaluates every step of topical crawling using the evaluator classifiers, but target recall doesn't assign any positive score to relevant web pages which are crawled but doesn't exist in target set *T*.

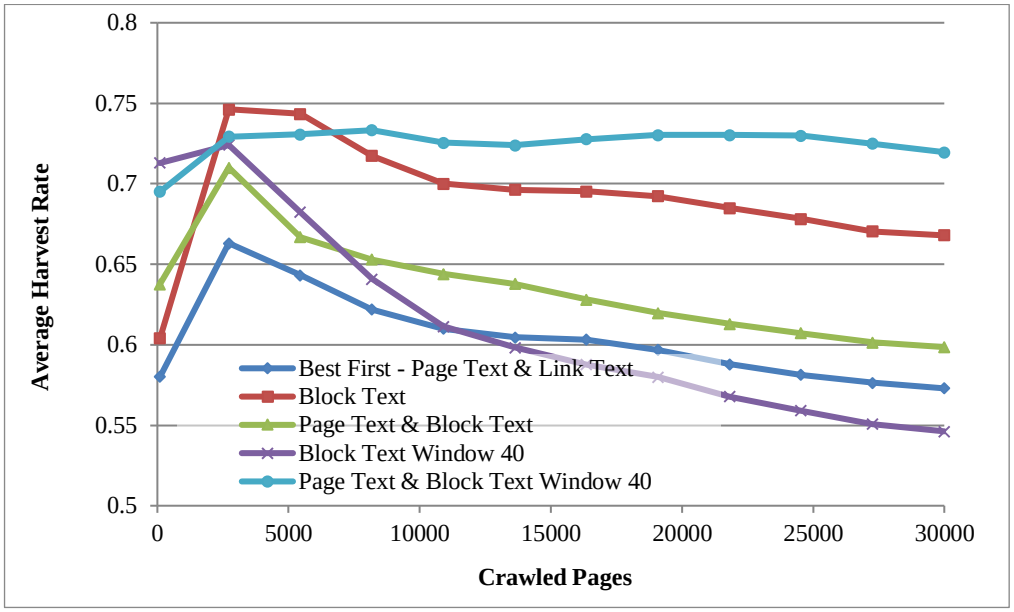


Figure 9. Harvest rate of BTW, best first and the block-based methods

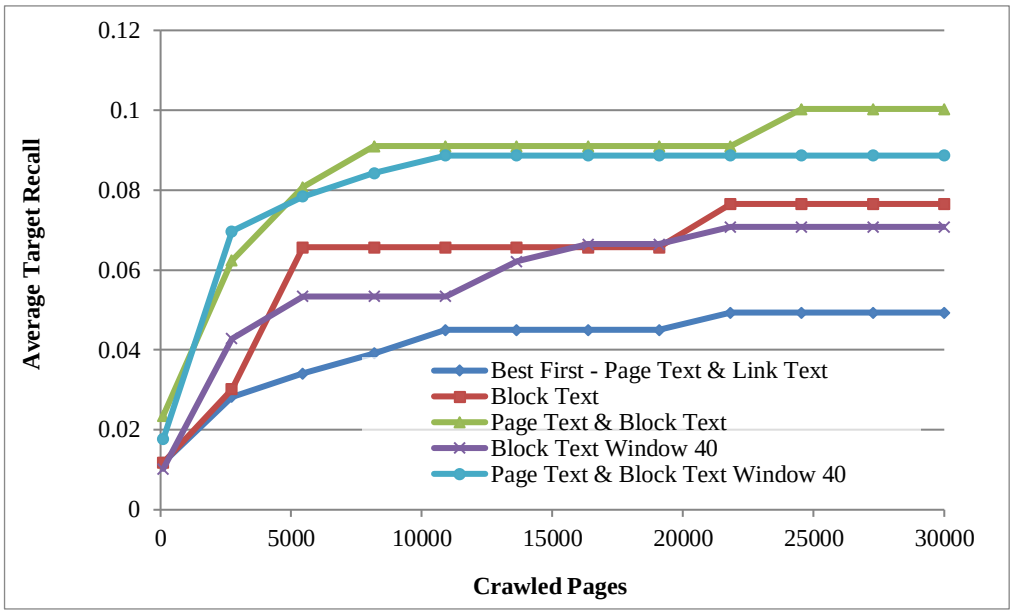


Figure 10. Target recall of BTW, best first and the block-based methods

In Figs. 9 and 10 we compared two versions of BTW with one version of the best first method and two versions of the block-based method as they are described in previous sections. We used window size 40 for BTW method as found to be effective based on the results of the previous experiment. Fig. 9 shows the harvest rate and Fig. 10 shows the target recall of methods. As we can see in Fig. 9 the version of BTW that combines page text relevancy with extracted link context relevancy (Page Text & Block Text Window 40) has better harvest rate than other methods. Also Fig. 9 illustrates that the block-based method that uses page text (Page Text & Block Text) has lower harvest rate than the variant of this method that doesn't use it. This situation doesn't occur for evaluated versions of BTW method; The BTW that uses page text (Page Text & Block Text Window 40) has much better harvest rate than the other version of this method (Block Text Window 40).One reason for this situation can be that in BTW method the text window limits the size of the text, and to improve the efficiency of the BTW method, it is necessary to combine the score of BTW with the score of page text. But in the block-based method, the text of the block is not limited by any windows and there is no need for the page text to improve the performance of the block-based method. As Fig. 10 shows the target rates of BTW methods and block-based methods are close to each other.

Dramatic changes of method's performance even after crawling ten thousand pages especially based on harvest rate metric shows the necessity of evaluating topical crawling methods based on longer crawling terms as done in this paper.

Table 4. Comparison of methods after crawling 30000 pages

Link Context Extraction Methods	**Harvest Rate**		**Target Recall**	
	AVG	***STDV***	***AVG***	***STDV***
Best First - Page Text & Link Text	0.57	0.33	0.05	0.06
Page Text & Text Window 10	0.59	0.30	0.07	0.06
Page Text & Text Window 20	0.64	0.23	0.07	0.05
Page Text & Text Window 40	0.56	0.28	0.09	0.08
Page Text & Block Text	0.60	0.27	0.10	0.10
Block Text	0.67	0.21	0.08	0.08
Page Text & BTW 10	0.61	0.24	0.11	0.09
Page Text & BTW 20	**0.69**	**0.13**	**0.11**	**0.12**
Page Text & BTW 40	**0.72**	**0.17**	**0.09**	**0.08**
BTW 40	0.55	0.23	0.07	0.05

Table 4 shows the averages and standard deviations of harvest rate and target recall at the end of crawling thirty thousand pages. Heuristically it is clear that using a short window size of 10 makes BTW method equivalent to text window method in some aspects and their close performance in table 4empirically shows it. The combinatorial versions of BTW that have used 20 and 40 words as their window sizes have better average harvest rate than other methods and their standard deviations are lower. Also, these two methods have an appropriate average target recall in comparison with other evaluated methods. Reported standard deviations in table 4 are consistent with reported standard errors in [28].

5. CONCLUSION

We investigated the cost-sensitive data acquisition problem with the focus on the cost of collecting pages as the learning instances from the web. We used novel topical crawlers as an efficient tool for effective use of available bandwidth to reduce the cost of collecting instances. These crawlers use the new proposed BTW method which is a hybrid link context extraction method for topical web crawling. The method combines a text window method with a block-

based method that uses VIPS algorithm for page segmentation and overcomes challenges of each of these methods using the advantages of the other one. BTW can find the position and size of link context accurately. Experimental results show the predominance of BTW in comparison with the state of the art topical crawling methods.

REFERENCES

[1] M. Kachuee, K. Karkkainen, O. Goldstein, D. Zamanzadeh, and M. Sarrafzadeh, "Nutrition and Health Data for Cost-Sensitive Learning," Feb. 2019.

[2] Z. Guan, J. Li, L. Wu, Y. Zhang, J. Wu, and X. Du, "Achieving Efficient and Secure Data Acquisition for Cloud-Supported Internet of Things in Smart Grid," IEEE Internet Things J., vol. 4, no. 6, pp. 1934–1944, Dec. 2017.

[3] L. Litman, J. Robinson, and T. Abberbock, "TurkPrime.com: A versatile crowdsourcing data acquisition platform for the behavioral sciences," Behav. Res. Methods, vol. 49, no. 2, pp. 433–442, Apr. 2017.

[4] G. M. Weiss and Y. Tian, "Maximizing classifier utility when there are data acquisition and modeling costs," Data Min. Knowl. Discov., vol. 17, no. 2, pp. 253–282, 2008.

[5] G. Pant and P. Srinivasan, "Link contexts in classifier-guided topical crawlers," IEEE Trans. Knowl. Data Eng., vol. 18, no. 1, pp. 107–122, Jan. 2006.

[6] D. Cai, S. Yu, J. R. Wen, and W. Y. Ma, "VIPS: a visionbased page segmentation algorithm." Microsoft Technical Report, MSR-TR-2003-79, 2003.

[7] F. Min, F.-L. Liu, L.-Y. Wen, and Z.-H. Zhang, "Tri-partition cost-sensitive active learning through kNN," Soft Comput., vol. 23, no. 5, pp. 1557–1572, Mar. 2019.

[8] M. Kachuee, O. Goldstein, K. Karkkainen, S. Darabi, and M. Sarrafzadeh, "Opportunistic Learning: Budgeted Cost-Sensitive Learning from Data Streams," Jan. 2019.

[9] N. Rubens, M. Elahi, M. Sugiyama, and D. Kaplan, "Active Learning in Recommender Systems," in Recommender Systems Handbook, Boston, MA: Springer US, 2015, pp. 809–846.

[10] Y. Gal, R. Islam, and Z. Ghahramani, "Deep Bayesian Active Learning with Image Data," in Proceedings of the 34th International Conference on Machine Learning, 2017, vol. 70, pp. 1183–1192.

[11] P. Teisseyre, D. Zufferey, and M. Słomka, "Cost-sensitive classifier chains: Selecting low-cost features in multi-label classification," Pattern Recognit., vol. 86, pp. 290–319, Feb. 2019.

[12] M. Liu, C. Xu, Y. Luo, C. Xu, Y. Wen, and D. Tao, "Cost-Sensitive Feature Selection by Optimizing F-Measures," IEEE Trans. Image Process., vol. 27, no. 3, pp. 1323–1335, Mar. 2018.

[13] G. M. Weiss and Y. Tian, "Maximizing classifier utility when training data is costly," ACM SIGKDD Explor. Newsl., vol. 8, no. 2, pp. 31–38, 2006.

[14] M. Klein, L. Balakireva, and H. Van de Sompel, "Focused Crawl of Web Archives to Build Event Collections," in Proceedings of the 10th ACM Conference on Web Science - WebSci '18, 2018, pp. 333–342.

[15] S. AbdulNabi and P. Premchand, "Effective performance of information retrieval on web by using web crawling," International J. Web Semant. Technol., vol. 3, no. 2, May 2012.

[16] S. Nandy, P. P. Sarkar, and A. Das, "Analysis of a Statistical Hypothesis Based Learning Mechanism for Faster crawling," Int. J. Artif. Intell. Appl., vol. 3, no. 2, Aug. 2012.

[17] J.-G. Lee, D. Bae, S. Kim, J. Kim, and M. Y. Yi, "An effective approach to enhancing a focused crawler using Google," J. Supercomput., pp. 1–18, Feb. 2019.

[18] H. Simanjuntak, N. Sibarani, B. Sinaga, and N. Hutabarat, "Web Mining On Indonesia E-Commerce Site: Lazada And Rakuten," Int. J. Database Manag. Syst., vol. 7, no. 1, p. 1, 2015.

[19] C. Iliou, T. Tsikrika, G. Kalpakis, S. Vrochidis, and I. Kompatsiaris, "Adaptive Focused Crawling Using Online Learning," Springer, Cham, 2018, pp. 40–53.

[20] M. Han, P.-H. Wuillemin, and P. Senellart, "Focused Crawling Through Reinforcement Learning," in ICWE 2018: Web Engineering, 2018, pp. 261–278.

[21] S. Batsakis, E. G. M. Petrakis, and E. Milios, "Improving the performance of focused web crawlers," Data Knowl. Eng., vol. 68, no. 10, pp. 1001–1013, Oct. 2009.

[22] M. M. G. Farag, S. Lee, and E. A. Fox, "Focused crawler for events," Int. J. Digit. Libr., vol. 19, no. 1, pp. 3–19, Mar. 2018.

[23] M. F. Farooqui, M. R. Beg, and M. Q. Rafiq, "An extended model for effective migrating parallel web crawling with domain specific and incremental crawling," Int. J. Web Serv. Comput., vol. 3, no. 3, p. 85, 2012.

[24] P. M. E. De Bra and R. D. J. Post, "Information retrieval in the World Wide Web: Making client-based searching feasible," Comput. Networks ISDN Syst., vol. 27, no. 2, pp. 183–192, 1994.

[25] S. Chakrabarti, K. Punera, and M. Subramanyam, "Accelerated focused crawling through online relevance feedback," in Proceedings of the eleventh international conference on World Wide Web - WWW '02, 2002, pp. 148–159.

[26] T. Peng, C. Zhang, and W. Zuo, "Tunneling enhanced by web page content block partition for focused crawling," Concurr. Comput. Pract. Exp., vol. 20, no. 1, pp. 61–74, 2008.

[27] D. Cai, S. Yu, J. R. Wen, and W. Y. Ma, "Block-based web search," in Proceedings of the 27th ACM SIGIR conference, 2004, pp. 456–463.

[28] C. Wang, Z. Guan, C. Chen, J. Bu, J. Wang, and H. Lin, "On-line topical importance estimation: an effective focused crawling algorithm combining link and content analysis," J. Zhejiang Univ. Sci. A, vol. 10, no. 8, pp. 1114–1124, Aug. 2009.

[29] "http://carl.cs.indiana.edu/fil/IS/JavaCrawlers/," Accessed May 2019.

[30] Y. Yu, S. Huang, N. Tashi, H. Zhang, F. Lei, L. Wu, Y. Yu, S. Huang, N. Tashi, H. Zhang, F. Lei, and L. Wu, "A Survey about Algorithms Utilized by Focused Web Crawler," J. Electron. Sci. Technol., vol. 16, no. 2, pp. 129–138, 2018.

[31] "http://www.cad.zju.edu.cn/home/dengcai/VIPS/VIPS.html," Accessed May 2019.

[32] "http://dmoz-odp.org," Accessed April 2019.

[33] G. Tewary, "Effective Data Mining for Proper Mining Classification Using Neural Networks," Int. J. Data Min. Knowl. Manag. Process, vol. 5, no. 2, 2015.

[34] R. Baeza-Yates, B. Ribeiro-Neto, and others, Modern information retrieval, vol. 463. ACM Press, 1999.

[35] "http://htmlparser.sourceforge.net/," Accessed May 2019.

[36] K. S. Jones and P. Willett, Readings in information retrieval. Morgan Kaufmann, 1997.

[37] J. Ifeanyi-Reuben Nkechi and E. Benson-Emenike Mercy, "An Efficient Feature Selection Model for IGBO Text," Int. J. Data Min. Knowl. Manag. Process, vol. 8, no. 6, 2018.

[38] Y. Yang and J. O. Pedersen, "A Comparative Study on Feature Selection in Text Categorization," in Proceedings of the Fourteenth International Conference on Machine Learning, 1997, pp. 412–420.

[39] "http://www.cs.waikato.ac.nz/ml/weka/index.html," Accessed May 2019.