Domain Specific Search in Indian Languages

by

Nikhil Priyatam, Srikanth Reddy Vaddepally, Vasudeva Varma

in

The 21st ACM International Conference on Information and Knowledge Management (CIKM 2012)

sheraton, Maui, USA

Report No: IIIT/TR/2012/-1

Centre for Search and Information Extraction Lab
International Institute of Information Technology
Hyderabad - 500 032, INDIA
October 2012

Domain Specific Search in Indian Languages

Pattisapu Nikhil Priyatam
Search and Information Extraction Lab
IIIT-Hyderabad
AP, India
nikhil.priyatam@research.iiit.ac.in

Srikanth Vaddepally
Search and Information Extraction Lab
IIIT-Hyderabad
AP, India
v.srikanth961@gmail.com

Vasudeva Varma
Search and Information Extraction Lab
IIIT-Hyderabad
AP, India
vv@iiit.ac.in

ABSTRACT

Focused crawling has wide number of applications in the area of Information Retrieval. It is a crucial part in building domain specific search engines, personalized search tools and extending digital libraries. Be it Google Scholar to search for scholarly articles or Google news to search for news articles, domain specific search is the most widely acclaimed application of focused crawling. Unfortunately, there are very few domain specific search engines available for Indian languages.

Sandhan is one such project which offers domain specific search for tourism and health domains across 10 major Indian languages. The amount of Indian language content on web is less compared to other languages. When we restrict the search space to a specific domain (say tourism) the probability of finding relevant pages reduces. Hence recall plays a major role in such a scenario. Due to the tendency of Indian language web pages linking to other language pages usually English, traditional crawling methods with well chosen seeds would end up crawling a lot of unnecessary content. This means that to gain a little recall we need to sacrifice precision and lot of resources.

In this work we try to explore ways of gathering Indian language tourism and health pages from the web for *Sandhan* using a language and domain specific focused crawler. With this setup we crawl the web extensively for Indian language tourism and health pages. We use different evaluation metrics to evaluate the quality of our crawl - precision, recall and harvest ratio. Using our approach we save nearly 80 % resources (disk space, bandwidth, processing time) while maintaining a recall of 0.74 and 0.58 for tourism and health domains respectively.

Categories and Subject Descriptors

H.3.3 [**INFORMATION STORAGE AND RETRIEVAL**]: Information Search and Retrieval—*Information filtering*

IKM4DR'12, November 2, 2012, Maui, Hawaii, USA.

General Terms

Languages, Experimentation

Keywords

Focused crawling, Web page classification

1. INTRODUCTION

Web search in Indian languages is constantly gaining importance. Many classic IR problems like domain specific search, web page classification, focused crawling, ranking etc need to be addressed for Indian languages. Though most of these problems seem to be solved for high resource languages like English, the solutions cannot be applied to Indian languages because of dearth of resources.

With the advent of increasing data, domain specific search is becoming inevitable. Even though generic search engines address the need of a normal user there are compelling reasons for building domain specific search engines which include technical, economical advantages and improving the search ability [5]. A domain specific search engine is supposed to index pages belonging to a specific domain and discard others. To do this there are 2 ways:

- Unfocused crawl with domain identification: In this strategy we crawl all pages and classify them as relevant or irrelevant using a domain classifier (a Web page classification algorithm). The relevant pages are indexed and irrelevant ones are discarded. The problem with this approach is that we need to crawl all the pages including the unnecessary ones. This is a severe wastage of resources and an unnecessary overhead.
- Build a Focused crawler: A Focused crawler on the other hand identifies URLs that are great access points to many relevant pages and fetches only those pages which are relevant. Here we do not have the luxury to look at the page and then decide whether or not they are relevant. Hence we have to take a decision which outlinks to include for further crawl based on the information available from the current page being parsed.

Focused crawlers can be broadly classified into 2 types:

- Close domain or topic specific crawlers: These type of crawlers fetch pages belonging to a specific topic like (HIV/AIDS, bicycling etc). Generally topics are defined using set of documents. Relevance of a page is

decided by the distance between the page and cluster centroid of the documents specifying the topic.

- Open domain or domain specific crawlers: These type of crawlers crawl pages belonging to specific domains like tourism, health, sports etc. Some of the existing crawlers are lawcrawler: searches legal information on the web, researchindex: specialized search for automatically finding computer science articles etc [5]. Generally, open domain focused crawlers are challenging to build because unlike close domains open domains cannot be defined using small set of documents. Therefore we need huge amount of high quality data to define the domain.

In this work we attempt to build an open domain focused crawler for *Sandhan* which will fetch **Hindi** tourism and health web pages. The rest of the paper discusses how we filter Indian language pages and classify the documents to build a focused crawler with good precision and high recall.

2. PREVIOUS WORK

Previous work in this area can be divided into 3 categories: classifier guided crawlers, crawlers which learn and metrics to evaluate focused crawlers. *Chakrabarti et al* built a classifier guided focused crawler, using a simple crawling strategy: If a page is relevant so might be its outlinks. They built a classifier which gets trained on initial set of positive and negative samples of the desired topic (say bicycling). The crawler starts with a small set of URLs called as seed URLs, fetches all of them and runs a classification algorithm on it. If the page is relevant then all its outlinks are added to the to-be-fetched queue [4]. They reported a precision of 0.4, but on a limited set of 10,000 pages. *Medelyan et al* used similar strategy for the medical domain, however they report the precision of their classifier rather than the performance of the focused crawler [9]. Other works include ontology based web crawling [12], [18]. *Aggarwal et al* provided a framework for *intelligent crawling* where the crawler gradually learns the linkage structure as it progresses [1]. Other similar works include [2], [13] and [8]. Works which gave evaluation metrics for focused crawler are [11],[10] and [16].

Many approaches mentioned above work with good precision but none of them can be used for building a web scale search engine, because while building a web scale search engine recall of the focused crawler is very important. Also most of these approaches work with good accuracy for close domain or topic specific crawlers none of them report the performance of the crawler for open domain problems like tourism or health. In fact *Fesenmeier et al* claim that such an approach followed by [4] will not work for open domain focused crawlers and specifically mentions that it will not work for the tourism domain possibly because it is difficult to define such an open domain with small set of documents. *Lie et al* mention that it is difficult to build a robust classifier for a highly unbalanced classification problem, with a very small training set of relevant pages [7].

We try to address this problem by building a classifier guided focused crawler consisting of a 3 class classifier (tourism, health, miscellaneous) built on a diverse Hindi web page collection [15].

3. APPROACH

In this work we build a classifier guided focused crawler for Hindi tourism and health pages. Though we present the experiments for only Hindi language pages, our approach is generic and can be easily extended to any other Indian language. The combination of an Indian language and an open domain poses the following challenges:

- Proprietary Fonts: Many Indian websites use their own proprietary fonts instead of Unicode, this creates difficulty in processing their text. The number of websites using these proprietary fonts is very high, hence missing out all such pages will severely affect our recall.
- Other language content: While considering Indian language pages (Hindi in our case), we will also get significant number of pages from other languages (mostly English) which have to be filtered out.
- Different Indian languages sharing the same script: Many Indian languages share the same script - for example *Hindi, Marathi, Sanskrit, Nepali and Konkani* all share the script *Devanagiri*, similarly *Bengali, Assamese and Maithili* share the same script *Bangla*. In such a scenario identifying language from character level information is not possible. Therefore identifying the language of the page becomes difficult.
- Scarcity of content: Content in *Hindi* language is very less when compared to English and other languages. So extracting tourism or health specific Hindi language pages is even tougher.
- Lack of training data: There are no ready made datasets available on which the classifier could be trained on, and for such applications we require huge and accurate training data (Note: The training data must come from the same distribution as testing data). Here our testing dataset consists of real life web pages. It is very difficult to get huge amount of good quality training data, and getting that would involve a good amount of manual labor.
- Open domain: As pointed out by [5] building domain specific web crawlers for domains like *tourism* is difficult because we can never accurately describe tourism to an algorithm, it is too generic to be explained by a set of documents.

Pingali et al proposed a working approach to overcome some of these challenges, but they do not address issues related to domain specific search [14]. Our approach mainly gains by the following modules:

3.1 Font Transcoder

One of the major problems with Indian language web pages is that those web pages authored before Unicode standardization use proprietary character encodings. Online versions of many popular Indian news papers, which can act as rich source of tourism or health information use proprietary fonts. As such it becomes difficult to parse such pages. To overcome this problem we use a Font transcoder that converts these pages into standard UTF-8 before parsing. This ensures consistency throughout the system, these non-standard pages are converted to Unicode during a preprocessing step while fetching. An open source tool called

Unigateway (a PHP port of Padma Converter) is used to do the transcoding. Once the conversion is done, all the other modules of the system work on the Unicode text [19].

3.2 Language Identifier

Language Identifier identifies the language of a web page. In the case of pages which contain multiple languages, the dominant language is taken as the language of that page. Language of a page can be identified in the following 3 ways:

- From the meta data of the page for example 'http://hi.wikipedia.org' clearly suggests that it is a Hindi page.
- From the character level information.
- From the N-grams profiles.

The first two approaches are problematic because meta data information (from URL or HTML lang property) might be erroneous and character level information cannot be used because many Indian languages have the same character set but different grammatical rules. In our Language Identifier we use a hybrid approach which looks at URL of a web page for gathering language specific clues (like language id or Indian language tokens) and N-gram profiles of parsed content of the web page. The N-gram profile of a particular language contains the frequent n-grams of that language. These files are generated by collecting the n-grams from large number of language specific documents in a separate process. The language of the web page is identified as the language whose n-grams are present most in that page [19]. For the extent of our research we have built a language identifier that can identify pages belonging to 10 different Indian languages: *Hindi, English, Gujarati, Punjabi, Bengali, Assamese, Marathi, Telugu, Tamil, Oriya*.

3.3 Web page classification

Web Page classification(WPC) is the task of categorizing web pages into different classes. This is an extended problem of document classification. A Web page classification might use a wide number of features from the simplest like URLs [6] to the complex ones like topic models [17]. We use the "bag-of-words" model, where each document is represented by a distribution over fixed vocabulary(s). Note that this model is language independent i.e. it does not require any language specific features. For our classification purpose we use a Naive bayes algorithm because it avoids the problem of over fitting and also works in scenarios where number of parameters greatly exceed the number of data points [3]. Our algorithm should be accurate and robust enough to classify pages running into millions with descent accuracy. For this to happen we must have huge amount of high quality data, not only that a very essential criteria for a classifier to perform with good accuracy is that the training data (data it gets trained on) must come from the same distribution as that of data it gets tested on (testing data). In our case testing data consists of real world web pages. Therefore the trick lies in careful sampling and efficient modeling of the distribution while collecting training data.

3.3.1 Data Collection

Before collecting the data we must clearly understand what exactly do we mean by a tourism or a health page. We call a page belongs to tourism domain if it contains any of the following information:

- Information about historical places.
- Tourist attractions.
- Travel Guide which includes cost of trip, best time to visit, transport facilities, nearby hotels, accommodation facilities, and nearby places of interest of a particular tourist spot.
- Food and other popular cuisines.
- On line services (if any) provided by the respective tourist spots.
- Latest news about a particular place which might include weather reports and any other news alerts.

Similarly documents belonging to the health domain are supposed to contain any of the following information:

- Complete information about any disease: Symptoms, precautions, cure and other related information.
- Information about medicines, their ingredients, possible side effects, availability and the type of medicine (*ayurvedic,allopathy,homeopathy etc*) .
- Information about clinics or hospitals, doctors etc.
- Latest research news about some ailments.
- Information related to nutrition, diet, personal hygiene etc.

We used the proposed guidelines to collect, classify and store Hindi web pages. These web pages contain information about tourist attractions, health topics and *Miscellaneous* content (neither tourism nor health). Our data collection process can be logically broken down into 3 steps: Query collection, firing them on a Search Engine and Classifying retrieved results. The process of data collection is described in figure 1.

Query Collection: This is a very crucial part of our system. If queries are not proper and diverse enough we cannot expect our training set of web pages to be an actual representation of entire data. Query collection is done manually with 30 people from different states in India speaking 10 different languages. Everyone of them is explained what qualifies to be a tourism query and health query and then each person is asked to prepare 3 set of queries:

- Regional Queries: These are tourism queries about specific places or locations where the person belongs.
- Non regional Queries: These are also tourism queries but are quite generic in nature (not related to the place where the person belongs).
- Health Queries: Any health related query.

Along with the queries the person is asked to provide English translation of the queries and the intent of the query. By collecting queries in this way we are ensure that the

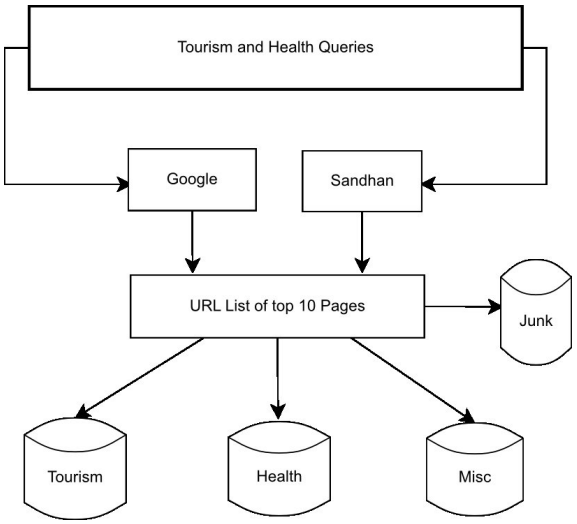

Figure 1: Data collection

coverage in all 3 domains (tourism, health and miscellaneous) is diverse enough. The division of regional and non-regional queries was done to ensure local and global coverage of tourist spots. Once we have the queries we translate all of them into Hindi. The translation is done manually so that no error creeps in this part. After translation we eliminate all duplicates (if any). We also make sure that all the queries are collected in isolation i.e. no one knows queries given by other people. In this way we collect 60 regional queries, 60 non-regional queries and 60 health queries.

Firing queries on search engines: Once we have the above set of queries we use a search engine to get the relevant web pages. Search engines like Google usually provide diverse results i.e. results from different hosts. For our work we actually fire the queries on Google as well as *Sandhan* (a search engine offering search in Indian languages) to ensure wider coverage and pages for diverse websites.

Classifying retrieved results: The results retrieved for each query are manually looked at and classified into one of the 3 defined domains as per the set guidelines. The top 10 results for each of Google and *Sandhan* are considered for manual classification. More than 10 results were also considered for some queries whenever the top 10 results had many miscellaneous pages. The pages going into miscellaneous category are the ones returned by the search engine in response to a tourism or a health query but belong to neither of the domains. Therefore the pages which are tagged miscellaneous are actually the boundary pages, in this way we do not make any specific efforts to collect miscellaneous pages. It has to be noted that a page is classified as tourism or health only if it strictly falls within our definition. It need not be completely relevant to the fired query. Pages which have no or meagre text are discarded. Though we present this data collection approach for building a robust WPC algorithm for tourism, health and miscellaneous pages, this approach is generic and can be applied to any domain where there is a need to model the distribution. The data collected above (in table 1) forms our training set. Each sample in the training set is manually labeled and hence we can safely assume that training set is highly accurate. A more detailed procedure for collecting data and storing it can be found in [15]. Using this dataset we build a 3 class naive bayes classifier with bag of words model. Table 2 shows the performance of our classifier. Our classifier works with a 10 fold overall cross validation accuracy of **81.89** %.

Table 1: Statistics of data collected

Domain	No of Web Pages
Tourism	885
Health	978
Miscellaneous	1354

3.4 Focused Crawling

For this research we implement 3 crawlers: a focused crawler for Hindi tourism pages, a focused crawler for Hindi health pages and an unfocused crawler. Both focused and unfocused crawler start with the same set of seed URLs which were manually collected in table 1. An unfocused crawler fetches a page and adds all its outlinks to the to-be-fetched tray. On the other hand focused crawler decides whether a web page is relevant or not using WPC algorithm described above. It adds the outlinks to the to-be-fetched tray only if that current page is relevant. With this set up we crawl the web till a depth of 3. Once the crawling process is done we categorize the web page into one of the 3 predefined baskets (tourism/health/miscellaneous) using the WPC algorithm described in section 3.3. The tables 3 and 4 show the language and domain statistics of the 3 crawlers respectively.

4. EVALUATION METRICS

In this section we evaluate the quality of the crawl with 3 metrics namely precision, recall and harvest ratio.

$$Precision = \frac{\#\ of\ relevant\ pages}{Total\ \#\ of\ pages\ fetched} \quad (1)$$

Once all the required pages are fetched by the crawler we find out the percentage of relevant pages using the WPC algorithm. Note that we are making an assumption that the WPC algorithm works with a high accuracy (testing accuracy) and therefore can be used for evaluating the quality of crawl. We do this because there are millions of pages in a crawl and manually looking at each page to judge its relevance is not feasible.

$$Recall = \frac{\#\ of\ relevant\ pages\ fetched\ by\ focused\ crawler}{Total\ \#\ of\ relevant\ pages} \quad (2)$$

Since we are using the same WPC algorithm which works with good precision, we safely assume that in both focused and unfocused crawls, the total number of relevant pages at depth 3 is equal to the total number of pages identified as tourism/health in the unfocused crawl.

$$Recall = \frac{\#\ of\ relevant\ pages\ fetched\ by\ focused\ crawler}{\#\ of\ relevant\ pages\ fetched\ by\ unfocused\ crawler} \quad (3)$$

Harvest ratio measures the average number of relevant pages retrieved over different time slices of the crawl [11].

5. RESULTS

With the metrics described above we evaluate the quality of our crawl. Table 5 and Figure 2 show the performance of our crawlers. In case of unfocused crawlers the definition

Table 5: Quality of crawl

Crawler	Precision	Recall
Unfocused crawler tourism	0.105	
Unfocused crawler health	0.106	
Focused tourism	0.4	0.74
Focused health	0.36	0.58

of relevance changes with whether we wish to fetch tourism pages or health pages that is the reason we show 2 different rows in table 5. Going by our definition of recall (described in equation 3) we note that unfocused crawlers cannot have a recall. Figure 2 shows the harvest ratios of the unfocused and focused crawlers respectively. The x-axis shows the number of URLs fetched and y-axis shows the corresponding precision. By examining table 3 we can easily infer that while crawling Indian language pages (Hindi in our case) we find significant number of other language pages (English predominantly, followed be unidentified), though we started with all Hindi seed URLs. This behavior is consistent with all the 3 crawlers. This affects our precision, because we consider a page to be relevant only if it belongs to the desired language (Hindi in our case) and desired domain. If we are able to successfully eliminate other language pages from our crawl we can improve our precision by 10 to 15 % for both tourism and health focused crawlers.

6. CONCLUSION

In this work we try to explore ways of gathering **Hindi** tourism and health pages from the web for *Sandhan* using a language and domain specific focused crawler. Since we are working with Indian languages we encounter many problems like encoding issues, language identification, domain identification and noisy/undesired pages creeping in our crawl. We build a font transcoder module to handle encoding issues. We build a robust Indian language identifier (for 10 languages) which recognizes the language of a web page. Tourism and health being open domains we experience various hurdles in domain identification. We choose to use a WPC algorithm based approach to solve this problem. We know that the test data for our WPC runs into millions and comparatively training data is very less. We also realize that size of the training data does not matter much if it is unable to represent the real world distribution of tourism, health and miscellaneous web pages. In our approach we have stressed on this aspect i.e. the training and testing dataset must come from the same distribution.

We have achieved this by efficient sampling while collecting training data as described in section 3.3.1. We built a focused crawler using this WPC algorithm. This reduced the number of undesirable pages (mostly found to be English pages) in our crawl. The focused crawler for health achieves a huge recall of 0.58, which means that it was able to fetch nearly 60 % of all relevant pages with a crawl size of 16 % of exhaustive crawl. The focused crawler for tourism achieves a huge recall of 0.74, which means that it was able to fetch nearly 75 % of all relevant pages with a crawl size of 20 % of exhaustive crawl. In both the cases we were able to save almost 80 % of the resources.

7. FUTURE WORK

In future we plan to eliminate other language pages from our crawl by identifying language of a page from its URL and then fetching the page if and only if it belongs to the desired language. We also plan to augment our language identifier to handle many more languages so that we reduce number of unidentified pages in our crawl. We also want to build web page classification algorithm using different features other that content words (title words, URLs etc). Rather than using only one crawling strategy we wish to experiment with many other strategies some of which are listed below :

- Deciding relevance based on the anchor text of URL : 'A page is relevant if its anchor text consists of tourism or health specific words'
- Relevance based on domain: 'A page is relevant if it comes from a trustworthy domain'.
- Taking multiple parents of a child node: 'A page is relevant only if all its parents are relevant'.
- Exploiting the link structure: 'If a particular subgraph forms a clique with most pages being relevant then all other nodes might be relevant'.

While comparing these strategies we wish to evaluate our focused crawler with many more evaluation metrics like F-Score, search length, remoteness, crawl robustness etc.

8. ACKNOWLEDGMENTS

We acknowledge the funding we have received from Ministry of Communication and Information Technology (MCIT), Government of India and Cross Language Information Access (CLIA) consortium members helped us.

Table 2: Confusion Matrix of Naive Bayes classifier

Tourism	Health	Miscellaneous	Precision	Recall	F-Score	
502	48	78	0.85	0.8	0.83	Tourism
6	500	45	0.75	0.91	0.82	Health
80	121	707	0.85	0.78	0.81	Miscellaneous

Table 3: Language Statistics of Focused and Unfocused crawler

	Unfocused crawler	Focused Crawler for tourism	Focused Crawler for health
Hindi	94995	31182	21298
English	83234	6557	9229
Gujarati	386	15	47
Punjabi	17	0	3
Bengali	275	3	21
Assamese	82	10	10
Marathi	921	220	152
Telugu	1543	12	59
Tamil	278	3	37
Oriya	1	0	0
Unidentified	26298	2530	4346

Table 4: Domain statistics for Focused and Unfocused crawler

Crawler	Tourism	Health	Total
Unfocused-crawler	21843	21967	208030
Focused-health	-	12691	21298
Focused-tourism	16226	-	40566

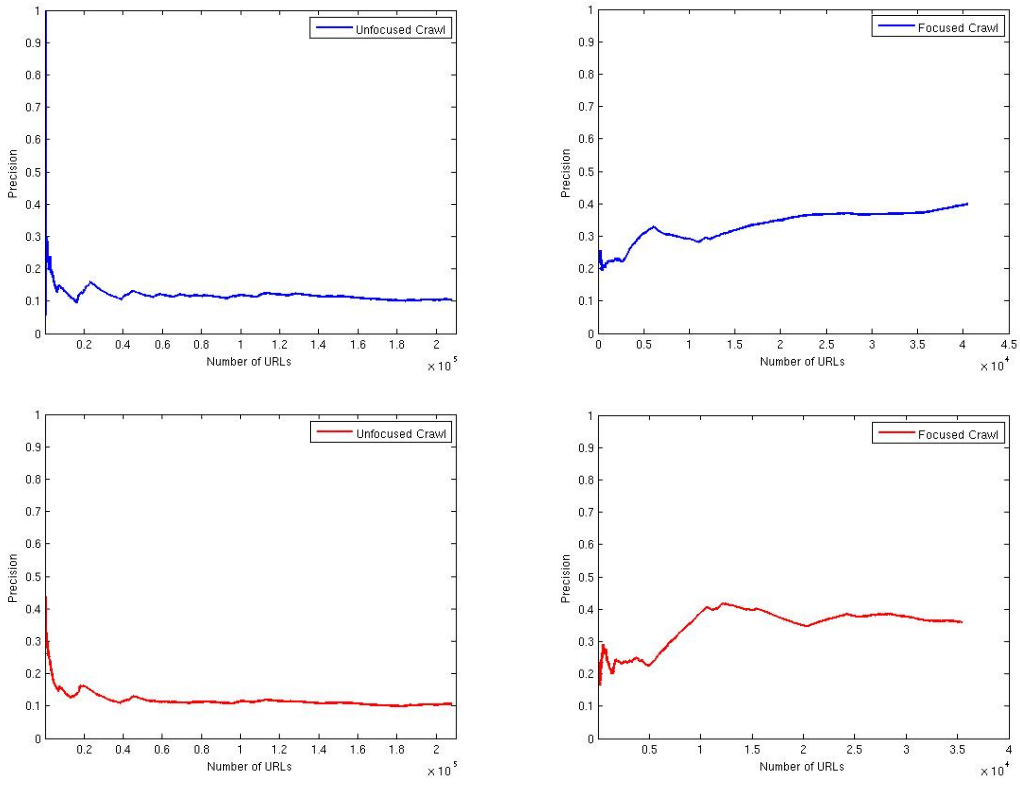

Figure 2: Performance comparison of unfocused and focused crawler in tourism and health domains. Blue and red color indicate tourism and health domains respectively.

9. REFERENCES

[1] C. C. Aggarwal, F. Al-Garawi, and P. S. Yu. Intelligent crawling on the world wide web with arbitrary predicates. In *Conference proceedings of World Wide Web 2010*, 2010.

[2] N. Angkawattanawit and A. Rungsawang. Learnable crawling: An efficient approach to topic-specific web resource discovery. In *2nd international Symposium on communications and Information Technology (ISCIT 2002)*, 2002.

[3] C. Bishop and S. S. en ligne). *Pattern recognition and machine learning*, volume 4. springer New York, 2006.

[4] S. Chakrabarti, M. van den Berg, and B. Dom. Focused crawling: a new approach to topic-specific web resource discovery. In *Journal of Computer Networks*, volume 31.

[5] D. R. Fesenmaier. *Domain Specific search engines*, pages 205–211. 2006.

[6] M. Kan and H. Thi. Fast webpage classification using url features. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 325–326. ACM, 2005.

[7] H. Liu, E. Milios, et al. Probabilistic models for focused web crawling. 2010.

[8] H. Liu, E. Milios, and J. Janssen. Focused crawling by learning hmm from user's topic-specific browsing. In *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 732–732. IEEE Computer Society, 2004.

[9] O. Medelyan, S. Schulz, J. Paetzold, M. Poprat, and K. Marko. Language specific and topic focused web crawling. In *Proceedings of the Language Resources Conference LREC 2006, Genoa, Italy.*, 2006.

[10] F. Menczer, G. Pant, and P. Srinivasan. Topical web crawlers: Evaluating adaptive algorithms. volume 4, pages 378–419. ACM, 2004.

[11] F. Menczer, G. Pant, P. Srinivasan, and M. E. Ruiz. Evaluating topic-driven web crawlers. In *SIGIR '01 Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 241–249. SIGIR ACM Special Interest Group on Information Retrieval, Aug 1999.

[12] D. Mukhopadhyay, A. Biswas, and S. Sinha. A new approach to design domain specific ontology based web crawler. In *10th international conference on information technology (ICIT 2007)*, pages 289–291, Jan 2008.

[13] G. Pant and P. Srinivasan. Learning to crawl: Comparing classification schemes. volume 23, pages 430–462. ACM, 2005.

[14] P. Pingali, J. Jagarlamudi, and V. Varma. Webkhoj: Indian language ir from multiple character encodings. In *Conference Proceedings of World Wide Web 2006*, 2006.

[15] P. N. Priyatam, S. R. Vaddepally, and V. Varma. Hindi web page collection tagged with tourism health and miscellaneous. In *8th international conference on Language Resources and Evaluation (LREC 2012)*, May 2012.

[16] P. Srinivasan, F. Menczer, and G. Pant. A general evaluation framework for topical crawlers. volume 8, pages 417–447. Springer, 2005.

[17] W. Sriurai, P. Meesad, and C. Haruechaiyasak. Hierarchical web page classification based on a topic model and neighboring pages integration. 2010.

[18] K. Stamatakis, V. Karkaletsis, G. Paliouras, J. Horlock, C. Grover, J. R. Curran, and S. Dingare. Domain-specific web site identification: The crossmarc focused web crawler. In *8th international conference on Language Resources and Evaluation (LREC 2012)*.

[19] V. Varma, B. Reddy, A. Mogadala, S. R. Vaddepally, P. N. Priyatam, and M. Bhagavatula. Input processing for clia. In *22nd issue for technological development for Indian Languages (TDIL 2011) [to appear]*, 2011.