Ontology Driven Focused Crawling of Web Documents

Dr. Abhay Shukla
Professor Department of Computer Engineering, SSA Institute of Engineering Technology, Kanpur
Address : 159-B Vikas Nagar Kanpur

Abstract - In recent year dynamism of the World Wide Web , the issue of discovering relevant web pages has become an important challenge. Focused crawler aims at selectively seeking out pages that are relevant to a pre-defined set of topics. Most of the current approaches perform syntactic matching, that is, they retrieve documents that contain particular keywords from the user's query. This often leads to poor discovery results, because the keywords in the query can be semantically similar but syntactically different, or syntactically similar but semantically different. Another drawback is that the query matching score is calculated taking into account only the keywords from the user's query. Thus, regardless of the context, the same list of results is returned in response to a particular query. Our objective is to present an approach for document discovery building on a comprehensive framework for Context-Ontology Driven Focused Crawling of Web documents. This framework includes means for using a complex ontology and associated context information. It also defines relevance computation strategy and with the help of algorithm and performance evaluation graphs it has been shown that crawling based on rich ontological structures and context information as background knowledge clearly outperforms standard crawling and focused crawling techniques.

Key words: ontology, focused crawler.

I. INTRODUCTION

The size of the publicly index able world wide web (WWW) has probably surpassed 14.3 billion documents [1] and as yet growth shows no sign of leveling off. Focused crawlers [2] aim to search and retrieve only the subset of the World Wide Web that pertains to a specific topic of relevance. Search engines [3] are therefore increasingly challenged when trying to maintain current indices using exhaustive crawling. Current search engines have the restriction on query length, enabling only a small set of terms to be contained in any query. The text-based search engines encounter the problem of ambiguity in words, for example, "guide" can be a 'book of information' or 'a person who advises or shows the way to others'. So, from the returned list of results the user often has to start his own search and choose the actually relevant documents among the result list. Yet, often the returned results are completely irrelevant and of no use. Searching and finding the required knowledge from the internet is a very arduous and tiring task.

The existing approaches have following drawbacks:-

1. They perform syntactic matching, that is, they retrieve documents that contain particular keywords from the user's query. This often leads to poor discovery results, because the keywords in the query can be semantically similar but syntactically different, e.g. 'thesis' and 'dissertation' (synonyms), or syntactically similar but semantically different, e.g. 'mouse' in the sense of a small rodent with a long tail and 'mouse' in the sense of a small hand-held device controlling
   the cursor on a computer screen.
2. The query matching score is calculated taking into account only the keywords from the user's query. Thus, regardless of the context and user's query, the same list of results is returned in response.
3. Though search engines, we can locate and retrieve documents of interest, they lack the capacity to make sense of the information those documents contain.

In this, we tried to improve the existing work in the area of intelligent and focused document crawling. The role and usage of context and context information for retrieving web documents has been considered. Therefore, considering the context in the query matching process, the quality of the returned URLs is enhanced. The returned URLs are better tailored to the needs of the user and less non-relevant URLs are filtered off. Contextual information [4] of the user is therefore an essential aspect to accomplish transparency in the searching process.

Again, the use of ontologies to specify the interrelations among context entities can ensure common, unambiguous representation of these entities. The whole work has been divided into four subsections:

A. Proposed architecture

B. An algorithm corresponding to proposed architecture
C. Example showing the usage of context
D. Performance graphs for measuring the performance

II. SECTION A

Context-Ontology Driven Focused Crawling of Web Documents

This section briefs the overview of framework and components of Context-Ontology driven focused crawling of web documents.

Overview

Fig. 1 shows the functional architecture of Context-Ontology driven focused crawling of web documents. The focused crawler [7] is started with a given set of URLs. The URLs are retrieved in the order of their rank. Normally the rank is assigned by the relevance measure. Next, preprocessor and separator extract promising links for the next crawling round. After preprocessing, entities are extracted i.e. words occurring in the ontology as well as context from the page and counted and relevance of the page is then calculated. Finally, with this, a candidate list of web pages in order of increasing priority is maintained in priority

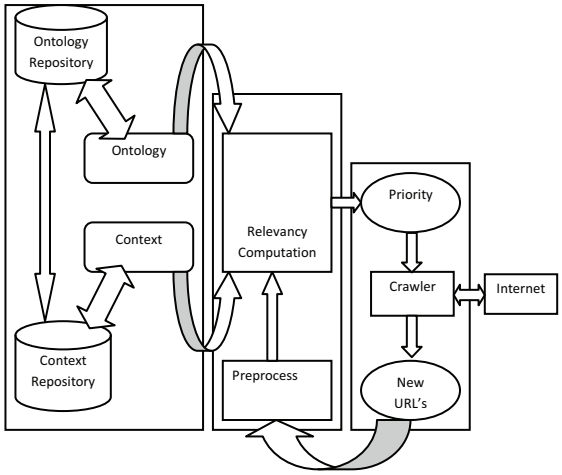


Fig. 1: Context-Ontology Driven Focused Crawling Architecture

queue.

Ontology [8] is a formal and declarative representation which includes the vocabulary (or names) for referring to the terms in that subject area and the logical statements that describe what the terms are, how they are related to each other, In this, Context Ontology defines a common vocabulary to share context information in a pervasive computing domain; and include machine-interpretable definitions of basic concepts in the domain and relations among them.

Architecture of Context-Ontology driven focused crawling of web documents

This section presents the architecture of the Context-Ontology driven focused crawling of web documents. The various components of the system, their input and output, their functionality and the innovative aspects are as follows.

Ontology and ontology repository

Ontologies are commonly used for a shared means of communication between computers and between humans and computers. Ontology has classes of objects, subclasses of objects and relationship among them, describing the kinds of entities in the world. Inference rules in ontologies add up further power. Ontology Repository allow users and agents to retrieve ontologies and metadata through open Web standards and ontology service. Ontology repository deals with ontologies to be stored in the registry describing the semantics of particular domains. Any components might wish to consult an ontology, but in most of the cases the ontologies will be used by mediator related components to overcome data and process heterogeneity problems.

Context and Context repository

Every ontology corresponds to a set of context objects that, among other things, describe the roles and tasks related to previous ontology utilizations. Several types of contexts are defined for different groups of users (e.g. for

ontology engineers, for domain experts) and Semantic Web resources (e.g. ontologies, parts of ontologies), while parts of the information comprised by a context object [9] are intended for automatic, for human processing, or both. The context repository maintains a database of several types of context data. The context repository is designed as a domain knowledge base management using the ontology. The context repository builds the semantic network and manages the instances of the semantic context and the relations between the objects and the services based on the domain ontology.

Preprocessing

A plain file from the internet in an arbitrary format and style is taken. The goal of the preprocessor is a table which allows comfortable processing regarding the later following Relevance Computation. At the same time promising links are extracted for the next crawling round as illustrated in Fig. 2.

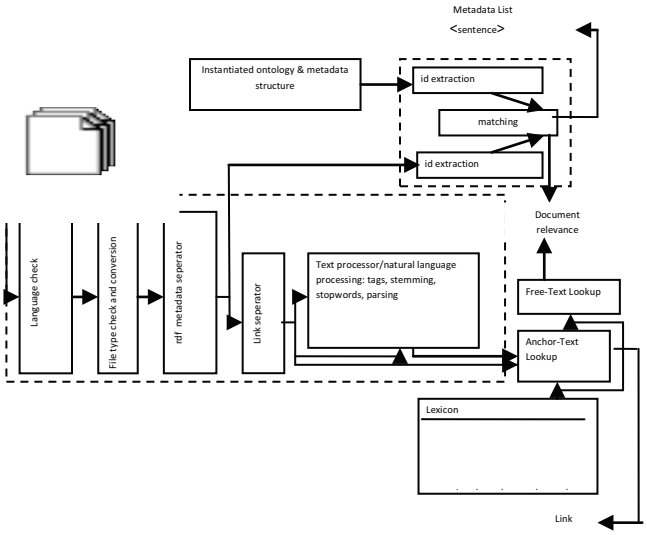


Fig. 2. Preprocessing

Relevance Computation

Relevance measure is a function which tries to map the content (e.g., natural language text, hyperlinks, etc.) of a Web document and, if available, the RDF-based metadata contained in a Web document against ontology and its existing, already collected, context information to gain an overall relevance score. This component maps the structure of the document against the ontology using different relevance measures. Depending on the level of correlation the score is returned. The documents itself as well as the contained links are ranked,

Definition (Relevance Function): The function f takes a document d, the instantiated ontology O, and Context information C as input .It results in *r := f(d, O, C), with r Є R being the relevance score.*

Steps in determining relevance using sentence query similarity

1. To compute the document query similarity **sim(D,Q)** vector-space model is used which uses cosine coefficient to measure the similarity[5]. Therefore,
   Retrieval relevance score of a document D is
   RSV(D,Q)=sim(D,Q)
2. After including sentence query similarity in relevance score of D, another formula is used
   RSV(D,Q)=Sim(D,Q) $\sum_{i=1}^{n}$ (Si,Q)
   Here, the second term on right hand side is the contribution to sentence query similarity. Here n denotes the number of sentences. C(Si,Q) denotes the similarity between Si (the ith sentence in D) and the query Q. Computing C(Si,Q) is based on the degree of co-occurrence of words between Si and Q. It is computed as

$$C(S,Q) = \begin{cases} \dfrac{|S \cap Q|}{|Q|} & \text{if } |S \cap Q| \geq (\tau |Q|) \\ 0 & \text{Otherwise} \end{cases}$$

   Here, $|S \cap Q|$ represents the count of the common indexing terms between S and *Q*.

$|Q|$ denotes the number of indexing terms in Q.

The constant α in equation (2) works as a weighting factor for the contribution by the sentence-query similarity. The exponent k in above equation is used to control the degree of importance of the high values of the ratio $|S \cap Q| / |Q|$ compared to the lower ones. As k increases, the high ratio becomes more important than the lower ones. τ is used to nullify the sentencial contribution in the cases where the number of common words is small.

3. Now to compute the similarity between anchor text and the query, it uses cosine coefficient and the contribution by anchor text is computed as

$$\sum_{i=1}^{n} \text{Sim}(Li,Q)$$

Where, Sim represents the cosine coefficient measure.

4. After that, the incoming link's anchor text Li in $D_{a(i)}$ is treated like a sentence in D, here $D_{a(i)}$ denotes the document which contains anchor text Li But the weight given to the similarity between an anchor text and the query can be different from that between a sentence and a query. The importance of the anchor text for the relevance of D can be different from that of a sentence in D. The contribution by the anchor texts Li's whose links point to document D to the relevance of D is calculated as

$$\beta \sum_i C(L_i,Q)$$

5. The last step is computation of named page finding task [21]. For this the relevance score is obtained by incorporating all contributions discussed above

$$RSV(D,Q) = sim(D,Q) + \alpha \sum_{i=1} C(S_i,Q) + \sum_{i=1} sim(Li,Q) + \beta \sum_{i=1} C(L_i,Q)$$

Instead of simple q ity queue is maintained so that relevant documents are retrieved first.

Merit of Proposed Framework

The proposed framework extends existing work in the area of intelligent and focused document crawling[10] which provides the following main achievements. These are:-

1. Enables retrieval based on context rather than keywords.
2. Can improve the quality of the retrieved results.
3. Capture the semantics of the user's query and of the contextual information [6] that is considered relevant in the matching process.
4. Makes the user's query more information rich and thereby provides means for higher precision of the retrieved results.
5. Can serve as an implicit input to a query that is not explicitly provided by the user. This prevents filtering out the undesired documents that require this input from the user, which leads to higher recall of the retrieved results.

III. SECTION B

Algorithm of Computing Final Match

Parameter : matchtable, contexttable, source

```
1.  queue ← new priorityqueue
2.  matchmatrix ← matchtable
3.  contextmatrix ← contexttable
4.      result ← 0
5.        for ( i = 0; i < size (source) ; i++ )
6.          source ← source [ i ]
7.          score ← 0
8.            for ( j = 0; j < size ( contextmatrix) ; j++)
9.              similarity ← compare (source, contextmatrix[ j ],matchmatrix [ i ])
```

```
10.              score ← score + similarity
11.              addtoqueue (queue, source, score )
12.             end for
13.            result ← result + score
14.           result ← result / noofelementinqueue
15.          end for
16.        end for
```

IV. SECTION C

Example

To illustrate the performance of retrieved documents a data set of seven documents has been taken and the searching is performed taking in account, usage of context and without context.

1) http://en.wikipedia.org/wiki/Mouse_(computing)

Mouse (plural mice or mouses) functions as a pointing device by detecting two-dimensional motion relative to its supporting surface. Physically, a mouse consists of a small case, held under one of the user's hands, with one or more buttons.

2) http://www.webopedia.com/TERM/m/mouse.html

A device that controls the movement of the cursor or pointer on a display screen.

3) http://en.wikipedia.org/wiki/Rodent

Rodentia is an order of mammals also known as rodents, characterised by two continuously-growing incisors in the upper and lower jaws which must be kept short by gnawing.

4) http://en.wikipedia.org/wiki/Mouse

A mouse (plural mice) is a rodent that belongs to one of numerous species of small mammals.

5)http://encarta.msn.com/encyclopedia_761569922/Mouse_(rodent).html

Mouse (rodent), common name for any small member of three families of rodents; large species of one of the families to which mice belong are known as rats.

6)http://www.fysh.org/~zefram/mouse/mouse_nbuttons.txt

Mice are currently available with a wide variety of numbers and types of buttons. The number of buttons may be one (Apple's infamous design), two (older PC mice), three (Unix mice and newer PC mice),

7)http://www.fvwm.org/doc/unstable/commands/Mouse.html

Defines a mouse binding, or removes the binding if Function is '-'. Button is the mouse button number. If Button is zero then any button performs the specified function.

Searching without using context

Search: Mouse

Now, using existing approaches all the seven documents containing the keyword mouse is returned whether it is related to computer or rodent.

Searching using Context Search: Mouse (Animal)

Now, using the proposed approach when the user added the context along with the keyword then the retrieved documents will be of mouse (animal) type only.

Result

1) http://en.wikipedia.org/wiki/Rodent

Rodentia is an order of mammals also known as rodents, characterised by two continuously-growing incisors in the upper and lower jaws which must be kept short by gnawing.

2) http://en.wikipedia.org/wiki/Mouse

A mouse (plural mice) is a rodent that belongs to one of numerous species of small mammals.

3)http://encarta.msn.com/encyclopedia_761569922/Mouse_(rodent).html

Mouse (rodent), common name for any small member of three families of rodents; large species of one of the families to which mice belong are known as rats.

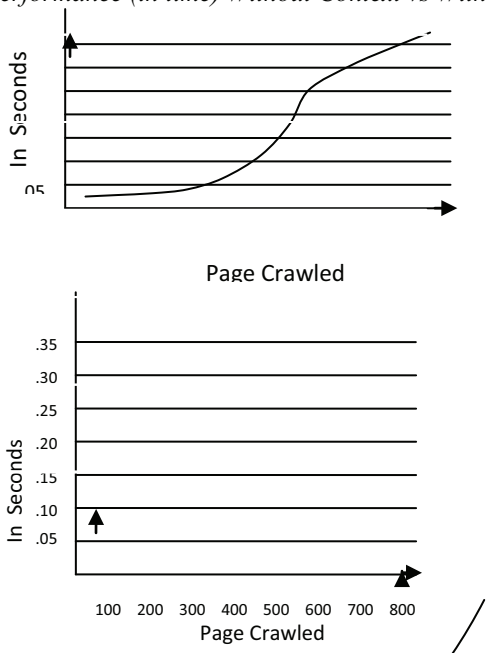
V. SECTION D

Performance Evaluation

The evaluation study shows how the different relevance measures perform in real life and how the quality of the input ontology influences the performance of the crawler based on context. The most crucial evaluation of focused crawling is to measure the rate at which relevant pages are acquired, and how effectively irrelevant pages are filtered

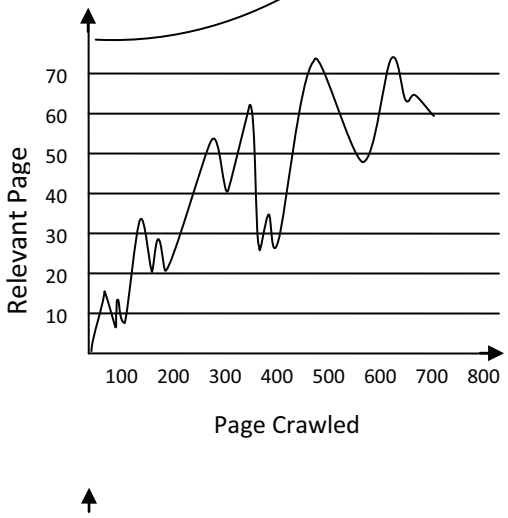
off from the crawl. The graphical representation of performance of retrieved documents in terms and relevancy is discussed below.

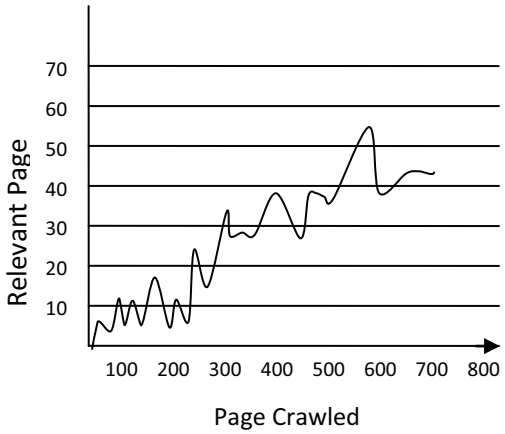
Performance (in time) Without Context vs With Context



In the first case, the pages crawled will be more when the time is increased. Since it not matching with the context so fake documents are also retrieved along with the useful documents. Whereas in second case, the performance in time will take little bit extra time when context is used but pages crawled will be more relevant.

Performance (in relevance search) Without Context vs With Context





In the first case, the performance in terms of relevancy is not good since context is not used. Whereas in second case, the performance in terms of relevancy is good because it filtered out non-relevant documents. So, it leads to high quality results.

VI. CONCLUSION

Search engines make an unprecedented amount of information quickly and easily accessible; their contribution to the web and society has been enormous. However, the "one size fits all" model of web search may limit diversity, competition, and functionality. Increased use of context in web search may help. As web search becomes a more important function within society, the need for even better search services is becoming increasingly important. On the semantic web, data has structure and ontologies describe the semantics of the data.

The proposed solution i.e. *Context-Ontology Driven Focused Crawling of Web Documents* overcomes the shortcomings of existing crawling approaches. uses the available contextual information and ontologies to semantically express user queries. The use of contextual information resulted in higher quality of the retrieved results. It makes the user's query more information-rich and thereby increases the precision of the retrieved results. Moreover it serves as an implicit input to a query that is not explicitly provided by the user. This allows our matching algorithm to select relevant documents that would be filtered out otherwise. When rich contextual information is available, it provides a potential resource for improving the performance of proactive retrieval systems. A context ontology is utilized to resolve inconsistent vocabularies in knowledge sharing and rule merging. The presented approach has to be implemented in future. Also, the ontology is currently expressed in OWL. So, the support for different semantic languages should also be extended. Thus, in the future, discovery of complex web service may be approached using the proposed crawling approach.

REFERENCES

[1] "World Wide Web has at least 14 billion pages. Tilburg, July 19, 2006. Original Tilburg University press release (July 10, in Dutch); Daily estimated size of the World Wide Web.
[2] S. Chakrabarti, M. van den Berg, and B. Dom, "Focused crawling: a new approach to topic-specific web resource discovery", in WWW-8, 1999.
[3] A. McCallum, K. Nigam, J. Rennie, and K. Seymore, "Building domain-specific search engines with machine learning techniques," in Proc. AAAI Spring Symposium on Intelligent Agents in Cyberspace,1999.
[4] Kouadri Mostefaoui, S., Tafat-Bouzid, A., Hirsbrunner, B., Using Context Information for Service Discovery and Composition (2003), Proc. of the 5th Conf. on information integration and web-based applications and services, Jakarta
[5] Myung-Gil Jang," Web Document Retrieval Using Sentence-query Similarity", Computer Science Dept., Yonsei University
[6] Van Setten, M., Pokraev, S., Koolwaai, J., Context-Aware Recommendation in the Mobile Tourist Application Compass (2004),in Adaptive Hypermedia 2004, Eindhoven.
[7] S. Chakrabarti, M. van den Berg, and B. Dom. Focused crawling: a new approach to topic-specific web resource discovery. In WWW-8, 1999.
[8] Deborah L. McGuinness," OWL Web Ontology Language Overview ", W3C Recommendation 10 February 2004
[9] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In Seventh International World Wide Web Conference, Brisbane, Australia, 1998.
[10] J. Budzik and K.J. Hammond. User interactions with everyday applications as context for just-in-time information access. In Proceedings of the 2000 International Conference on Intelligent User Interfaces, New Orleans, Louisiana, 2000. ACM Press.