2011 International Conference on Information and Intelligent Computing
IPCSIT vol.18 (2011) © (2011) IACSIT Press, Singapore

TIC: A Topic-based Intelligent Crawler

Hossein Shahsavand Baghdadi and Bali Ranaivo-Malançon

Multimedia University, Faculty of Information Technology, Cyberjaya, Malaysia
Email: bahamin.shahsavand@gmail.com, Email: ranaivo@mmu.edu.my

Abstract. The Web has been growing and now it is the most important source of information in the world. However, searching the web is a significant issue in many areas such as web mining. *Topic-based Intelligent Crawler* is a focused crawler to download the relevant pages to seed page regarding to their topics. If their topics are found similar, the pages would be considered as relevant. TIC is a crawler with ability to identify the topic of pages automatically and make decision about the similarity between them. One complete cycle of this process consists of four main stages. First we need to identify the hub pages associated to the seed page. After that, we should clean the seed and its hubs to extract pure text information. The third step is to identify the topic for each page and the last activity is finding out the hubs with a similar topic with the seed page.

Keywords: HTML*;* Crawling*;* Topic

1. Introduction

Although the growth of the Web arise many problems in information technology domain, it also implies several issues for mankind. The huge amount of information available in the Web requires special means to be searched. Therefore users who are not familiar with this media and who apply a simple method to find their needs may be unsatisfied easily. In this work, we intend to provide a tool for better search and easy use for crawling the Web.

Topic-based Intelligent Crawler (henceforth called **TIC**) is a focused crawler that allows users to download and classify relevant pages from seed page based on their topic similarity. The identification of the topic is done automatically, and thus users have just to provide a seed page and will get at the end of the process similar pages. In this work, we use the term “topic” as a stream of words which represent the content of text. A topic is different from a *title*, which is also a sequence of terms but rather represent the name of a work and does not necessary represent the content of this work.

Finding the similarity between pages based on their title is not an accurate way because the page’s title is not always stands for its content. It is therefore more accurate to work with document’s main topic to determine such relevancy. On the other hand, there are some issues to identify the web page’s topic. The first issue that we are dealing with is cleaning the HTML and extracting the plain text from it.

Today most of web pages are in HTML format and include many different sections such as main text, links, Ads and so on. So the only part that is required to identify the page’s topic is the main text. Since there are normally more than one topic is considerable for a text document, we need to perform an algorithm to identify the main topic for each web page. Recognizing the main topic is the second issue that we should address. Although there are several available approaches to identify a general document’s topic, applying them on an HTML page in order to identify the main topic need some modification.

Crawling the relevant pages is a great task which can facilitate the crawling process for those who are working on a specific purpose and need to collect the relevant pages in a specific area. TIC provides them such facility and collects the similar pages by considering their topics. As it has mentioned before, topic stands for content of document and can be participated in as document similarity determination process. It is therefore possible to identify the documents topic first and then make decision about their relevancy base on identified topics.

In TIC, we start the crawl cycle with cleaning the seed page by removing the irrelevant tags and extracting the main information in plain text format. HT2X[ML] [1] is the tool that we exploit to clean the HTML file. After cleaning the seed, we try to identifying the seed's topic by modified Chen's algorithm [2]. We then performing the same process for all seed's hub pages and provide a list of seed's topics to make a comparison in terms of semantic relevancy using LSA technique. At the end we store the relevant hubs with similar main topics to hub's main topic.

The rest of this paper is structured as following. In Section II we investigate some related works to various parts of TIC. Section III includes the methods that we have used to develop TIC. In Section IV we explain the details about TIC implementation and in Section V we explain the evaluation of TIC and the results that we achieved. In Section VI we propose some approaches to improve the reliability and accuracy of current version of TIC. Finally, Section VII belongs to summary and conclusion of our works.

2. Related Works

Normal crawlers are sensitive to some keywords determined by user and filter the web pages based on those keywords. If the linked pages to seed page contain exact those keywords, they would be downloaded and if they don't, they would be rejected. This approach is not accurate and miss the pages include some antonym words or relevant semantics. On the other hand determining the efficient keywords is not always a simple task and cannot be done by all the users.

Focused crawlers have been created for specific purpose and try to not downloading irrelevant pages. Focused crawlers are different in both objective and method. However almost all focused crawlers are looking for relevant pages, this relevancy has different meaning among them.

As instance, [3] proposed a focused crawler. They applied an algorithm called "shark-search algorithm", which is an improved version of the fish-search algorithm proposed by [4]. They used topic similarity of vector space model [5] and make parameter for URL ordering process. They made a comparison between topic keywords and content of web pages and determine the similarity among them. In fact, if downloaded page carries high similarity weight, the page and its embedded URLs will assess related to that topic. This similarity weight consists of two parts: *content similarity* that comes from the content of webpage and *anchor text similarity* which related to page's URL and its similarity to page topic.

3. Steps in TIC

TIC has fore main steps for a complete crawling cycle. These steps are described in following: **Finding current page hubs with regular expressions**

Crawling process starts with an HTML seed. After considering it as a current page, TIC tries to extract its hubs. To do this, we used Regular Expressions (RE). In HTML, all the URLs are embedded in the form of attributes by "href" name and using RE makes it possible for us to extract all "href" attributes. By using this filter, we are able to recognize all tags that include a link. So the input for this part is an HTML file and the output would be all the embedded links in that HTML. After extracting these URLs, we keep them in queue to be used in next steps.

3.1. Cleaning HTML documents with HT2X[ML]

To identify the page's main topic, we need the plain text of each page. HTML has tagged data and doesn't provide plain text. Hence, we need to clean the HTML in order to extract pure text. HT2X[ML] [1], is a sophisticated HTML converter to convert HTML tags into XML and plain text format. In terms of output format, HT2X[ML] has two different output format; XML and plain text. Since XML is a totally structured format, it is suitable to those purposed that needs the structured data, especially in data migration. Here we need the plain text to determine the most significant parts in each sentence. HT2X[ML] lets the users to select any tags that they need. It has the plenty of all possible tags and the users just need to select which tags they are interested. Furthermore, it provides an option to choose the important tags automatically. In this case, the tags which are most probable to carry the main information will be selected. The tags like <p>, <a> or <b> are some example of these kind of tags. In TIC we set HT2X[ML] to use this option.

3.2. Identifying Topic with Modified Chen's Algorithm

Identifying the page's main topic is the most prominent issue in TIC project. there are many approaches to identify a document's main topic. They use different methods to address this issue and they obtain various results based on their techniques. Chen's algorithm [2] is one of the most accurate and suitable approaches which has been used in TIC. Although this algorithm has proposed to identify general documents, we perform some modification to push its accuracy and make it proper for our work. [6]

After cleaning HTML page by HT2X[ML], plain text is available and we can perform modified Chen's algorithm. This algorithm has different steps and needs some tools. Sentence Separator, Stemmer, and Stanford Parser are the tools that we need to exploit in topic identification step.

3.3. Document similarity with LSA

The second issue in TIC is determining the similarity between identified page's topics. This similarity is used to make decision about the relevancy of pages in pages in hub selection section. LSA is the technique that we have used to recognize such similarity. We create the final matrix in LSA and select pages that their main topics have the greatest value in this matrix. Figure 1 illustrates the data flow in TIC.

4. Implementation of TIC

Is the Microsoft's programming platform to create new applications. .NET is a visual programming language and let the programmers to use great variety of visual components to create a proper interface and have effective interaction with end users. On the other hand, it provides the ability to have secure communication through the modules and implement different software architecture models.

Visual Studio .NET is a programming platform and supports multi programming language syntaxes such as Visual Basic, C and Java. In fact, .NET has a middle code and translates all input syntaxes into this code and then converts it to machine language. TIC has been coded by VB.NET syntax because of its strength in string processing. Like all developed software by .NET, TIC can be run on a computer with Microsoft Windows operating system.

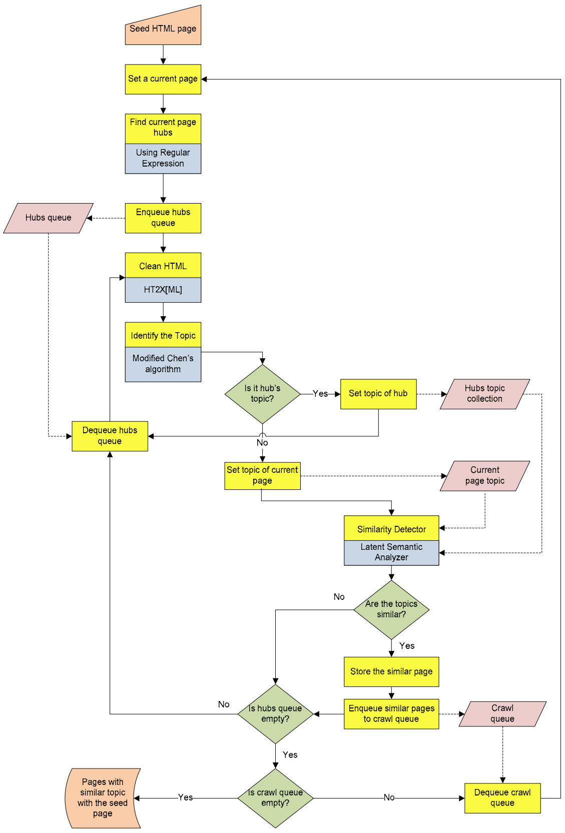

Figure 1 Data Flow in TIC

The main architecture has been used to develop TIC, is Layered Architecture. TIC has three main layers to separate different kind of tasks. These three layers are UI (User Interface) to interact with users, BOL (Business Object Layer) to perform main calculation and routines and Utilities which has been used as a great library to do some specific jobs.

The UI layer is an interface to make a proper interaction between users and BOL. All the inputs and settings should be set by this layer. Also the outputs can be monitored through UI. The BOL layer includes all processing routines associated with crawling process, Topic Identification, and Determine Similarity. The BOL consists of four separate classes; cProcessor, cTopic_Recognizer, cLSA, and cVariable. Utilities layer consists of two main engines. The first is an HTML parser, which accepts its URL for input. The result would be a set of parsed nodes. The second part in this layer is NLP tools including a chunker, tagger, statement separator, and Stanford parser. This module calls Antolope dll files via API technology. Figure 2 illustrates the modules and architecture of layers in TIC.

5. Evaluation of TIC

Topic identifier is one of the most prominent modules in TIC. Topic identifier stands for all functions, modules, and data types which are created a section to identify the topic for a webpage. In fact, topic identifier should be able to determine a stream of terms as a topic for any arbitrary webpage. The method which we have used to evaluate topic identifier was human result against machine result. We have selected 200 random pages from Wikipedia and compared the embossed topic and the identified topic by TIC. By this comparison we found out that in 20% of cases the two topics are totally matched. In 66% they were partially matched and in 14% of pages the topics were different. Figure 3 shows the result of this experiment. The detail of experiment is available in previous publication. [6]

There are three main reasons to topic identification failure.

5.1. File Format Issue

With a short look at the monitoring grid, we can observe immediately that most of the failures happened for some URLs which are not stand for a markup page. TIC as a web crawler extracts all the links embedded inside the seed page and consider them as hub pages. However, some of them are not link to the page with markup code (.htm, .html, .asp, .php) and carry the address for some other files like PDF or PNG files. In these cases, TIC cannot extract any textual information and there will be no topic.

5.2. Language Issue

The second reason for topic identification failure is the webpage language. English is the working language for this version of TIC. Therefore, all modules and functions in TIC are working under English language and are not able to recognize the topic for the pages in other languages. By adding a language recognizer module in future, it will be possible to identify the topic for pages in any languages.

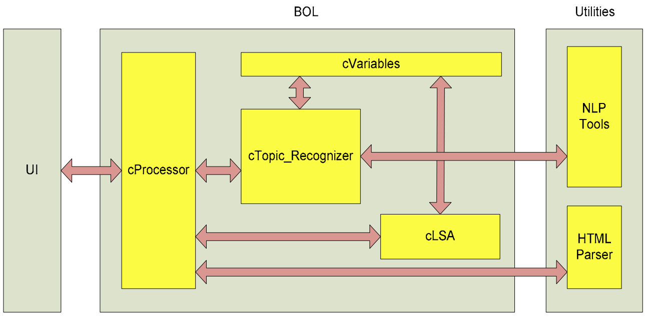

Figure 2 Layer Internal Architectures and Interaction between modules

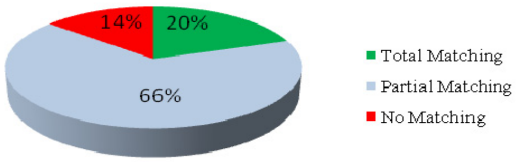

Figure 3 Percentage of Different Results for Automatic Topic Identification Algorithm [6]

5.3. HTML Structure Issue

The third cause of failure in topic identification part is the structure of webpage. It is previously mentioned that topic stands for the stream of terms which carry the semantic of text inside the document. What if the webpage doesn't include any text inside? To illustrate it, some pages have no paragraph text and include some pictures and links or just a list of words. In these cases, there will be no body text to extract and identify the topic for them.

In terms of time analysis, the average processing time for one page is 15742.5 ms which has been obtained from 200 random pages. As it has been mentioned before, many factors are influenced the interval time. The amount of embedded text in webpage and the Internet connection quality are the most significant factors which can change the interval time in TIC.

6. Discussion and Future Work

As all other applications, TIC has some constraints which reduce its usability. Some of these constraints have been discussed in previous chapter. Obviously, eliminating all or some of those issues can improve the reliability of TIC and make it more usable for the users. Previously, we mentioned three main limitations for TIC: language issue, bad-structured HTML issue, and synonymy issue.

Current version of TIC is able to process only English pages. This limitation is emerged from NLP tools which are able to process only English texts. To address this issue, two approached are considered. In the first approach, we can use the modules which are able to process in other languages. The next issue about current TIC is the HTML file with poor structure. HTML has a standard which indicates the tags and their usage. According to this standard, for example, the main text of each page should be embedded into <p> tag which represents the paragraph concept. All the HTML authors should follow this standard to create their Web pages; however, sometimes they do not do this. In this case, we are facing with the HTML pages which are able to be previewed by the browsers but not able to be processes by TIC. Synonymy is another issue which reduces the accuracy and usability of TIC. As it has been described in previous chapters, after identifying the main topic for the seed and the hub pages, TIC tries to determine the similarity between identified topics. This version of TIC makes decision based on common words between the topics. However, sometimes the topics do not have any words in common but they carry the same semantic. This problem may causes by synonym words. To illustrate it, the verb "buy" and verb "purchase" have exactly the same meaning but TIC cannot recognize the synonymy between them. To address this issue, we can use a lexical database which makes us able to find out the synonymy relation between the words.

7. Conclusion

TIC is a focused crawler which is going to help users to crawl relevant pages as easy as possible. The only thing that the users need to provide is the URL for the first webpage as the seed page. TIC determines the topic (not title) for the seed and all hub pages linked to seed page. Then it downloads all relevant pages based on topic similarity.

To achieve this purpose, TIC exploits some tools and techniques. It first extract the hub URL's associated with seed page by regular expressions and then tries to remove the irrelevant tags by HT2X[ML] [1] which is a sophisticated HTML converter. After extracting the pure text information, TIC attempts to identify the page's topic by modified Chen's algorithm [2]. This new algorithm is based on weighted chunks and determines most weighted chunks as the topic for a webpage. After identifying the topic for seed and all hub page, TIC determine the hub pages with most relevant topics to the seed's topic by LSA and store them as the output. This process is one complete crawling cycle and could be iterated as many as user wants by considering the previous hub pages as a new seed.

TIC is implemented in .Net framework. The HTML parser is developed with C# language and the other parts are implemented by VB.Net language. Layers architecture has been used for TIC which consists of three main layers. User Interface (UI), Business Object Layer (BOL), and Utilities are the main layers in TIC.

We changed the selected elements is Chen's algorithm and selected the NPs instead of nouns and head of VPs instead of verbs. In this part we achieve 86% matching (for both total and partial matching) among 200 random pages from Wikipedia.

8. References

[1] Shahsavand Baghdadi, H., & Ranaivo-Malancon, B, "HT2X[ML]: An HTML Converter". ICEIE. Kyoto, Japan: IEEE. 2010.

[2] Chen, K.-h.. "Topic Identification in Discourse." Taipei, Taiwan: Department of Computer Science and Information Engineering, National Taiwan University. 1995.

[3] Hersovici, M., Jacovi, M., Maarek, Y., & D. Pelleg, M. "The shark-search algorithm. An application: tailored Web site mapping". seventh international conference on World Wide Web, Brisbane, Australia. 1998, pp. 317-326.

[4] De Bra, P., & Post, R.."Searching for Arbitrary Information in the World-Wide Web: the Fish-Search for Mosaic". Second WWW Conference. Chicago, 1994.

[5] Baeza-Yates, R., & Ribeiro-Neto, B. "Modern Information Retrieval". ACM Press. New York. 1999.

[6] Shahsavand Baghdadi, H., & Ranaivo-Malancon, B. " An Automatic Topic Identification Algorithm". Journal of Computer Science 7 (9). 2011, pp 1363-1367.