KEEPING UP-TO-DATE IN THE MEDICAL FIELD AT NO COST

Nuno Escudeiro[1], Paula Escudeiro[2]

[1,2]Instituto Superior de Engenharia do Porto
Rua Dr António Bernardino de Almeida, 431, 4200-072 Porto
[1]LIAAD-INESC Porto L.A.
Rua de Ceuta 118, 6º, 4050-190 Porto
[2]GILT
Rua Dr António Bernardino de Almeida, 431, 4200-072 Porto
T.+351 228 340 500; F.+351 228 321 159; E. [1]nfe@isep.ipp.pt; [2]pmo@isep.ipp.pt.

Abstract

Medical information is available from many distinct sources in the web such as resources' hubs and medical news services.

Being aware of constant advances and innovation in the medical field is necessary to achieve a good work performance. However, keeping up to date in the medical field requires significant effort, and consequently a large cost.

This effort might be largely reduced by automating part of the process required to gather relevant information.

In the current paper we propose an automatic process to assist users to keep up to date in medical information. This process retrieves medical information from the web and organizes it in a taxonomy that is especially tailored towards users' specific needs.

Our results led us to conclude that it is possible to achieve a very significant reduction in the user work load and still have high quality content.

1. INTRODUCTION

Topic focused web portals reduce users' effort in the task of finding the right information at the right time and keeping up-to-date in their area of expertise. This reduction is achieved mainly by transferring part of that effort to an editor that works on behalf of all the portal users which are assumed to have common needs. However, creating and maintaining attractive and functional thematic web resources still demands for a high effort from web site editors.

This effort can be largely reduced by semi-automating some of the editor's tasks, mainly those related to content gathering and organization.

Having a medical resource focused on a specific area, that automatically gathers and organizes only the relevant information on a given area, seems valuable since it might present a permanently updated view of the advances and innovations in that specific area while requiring no effort from its users once it has been setup.

In the present work we propose a process that automates most of the hard working tasks that are required to keep a resource on medical information up-to-date and valuable.

In this process, editors are just required to: (1) set the web address of the sites that they usually visit when looking for fresh information on their area of expertise, the *seeds*; (2) define the taxonomy of topics representing the ontological structure required for the resource and (3) provide a few text paragraphs, previously collected from the seeds, that are representative of the topics in the taxonomy.

From these inputs, we apply text mining techniques to learn a classification model for the resource taxonomy and to extract text snippets from the seeds. These text snippets are then automatically classified and placed in the right topic on the taxonomy.

This process provides a huge reduction of the editor workload producing a resource with a quality that is close to that of the resource compiled by the editor without being assisted by our prototype.

2. AUTOMATIC RESOURCE HARVESTING

Our proposal may be seen as an automatic resource compiler, i.e., a system that seeks and retrieves a list of the most authoritative documents for a given topic [1]. In our work we are interested in collecting and organizing medical information, in a continuous effort to keep a web page up-to-date and organized according to a specific need.

Many automatic resource compilation systems have been proposed in the past.

With Thesus [6], users search documents in a previously fetched and classified document collection. Documents are classified on their content and link semantics. The system includes four components: acquisition, information extraction, clustering and query.

WebLearn [8] retrieves documents related to a topic, specified through a set of keywords, and then automatically identifies a set of salient topics, by analyzing the most relevant documents retrieved in response to the user query. The identification of these salient topics is fully automatic.

iVia [9] is an open source virtual library system that collects and manages resources, starting with an expert-created collection that is augmented by a large collection automatically retrieved from the web. iVia identifies relevant internet resources through focused crawling [4] and topic distillation approaches.

Personal View Agent [5] is another personalization system that learns user profiles to assist them when searching for information in the web. This system organizes documents in a taxonomy, which is user dependent and dynamic.

Metiore [2] is a search engine that ranks documents according to user preferences, which are learned from user historical feedback depending on the user objective.

Personal WebWatcher [10] is a system that observes users' behaviour – by analyzing page requests and learning a user model – and suggests potentially interesting pages.

The ARC system [3] compiles a list of authoritative web resources on any topic. The algorithm has three phases: retrieval of root and expanded set, analysis of anchor text and relevance inferring, compute authority and hub measures in the expanded set.

Letizia [7] is a user interface agent that assists a user browsing the web, suggesting potentially interesting links. Interest in a document is learned through several heuristics that explore user actions and current context.

In our work users' interests are specified by the web site editor through a taxonomy and by specifying the topics in the taxonomy through examples. From there on, the system learns the topic, periodically inspects the seeds identifying and retrieving text paragraphs that are assigned to the topics of interest.

3. METHODOLOGY

The main goal of the current work is to reduce the work load that is required from the website editor to keep the content of the medical information resource up-to-date.

This reduction is achieved by automating the tasks that are required to update the resource.

To harvest the relevant information, according to user interests, we have devised a methodology with four stages (Figure 1):

Acquisition: in this stage we download the seeds previously specified by the editor and stored in a database. These are URL for valuable sources of information usually visited by the editor when looking for information to update the web site on an unassisted mode.

Pre-processing: the content of these web pages, downloaded from the seeds, are split into text paragraphs. Extracted paragraphs are stored in text files and registered in the resource database. Each of these paragraphs is then pre-processed and indexed to build a doc-term matrix representing the corpus. After this, the database is updated on the number of extracted paragraphs per seed; this will serve as a measure of the seed relevance.

Learning: in this stage we use Support Vector Machines to learn a classification model for the taxonomy which has been previously provided by the resource editor. This classification model is learned from a set of exemplary text paragraphs – that are representative of all the classes in the taxonomy to learn – also provided by the editor. The classification model will henceforth be used to classify new incoming text paragraphs. The learning stage can be skipped if the current classification model is accurate; if it is not, the editor may manually label additional paragraphs and then use them to rebuild the classification model which is expected to be more accurate.

Organization: the final stage. The classification model, generated in the previous stage, is applied to new incoming text paragraphs assigning them to a given topic. This information is used to build the html code for the resource's web page.

3.1. Architecture

Our methodology includes five core functions to ensure its goals:

Download seeds: connects to the database to get the seeds URLs and then downloads them.

Split into paragraphs: splits web pages into separate paragraphs and saves each paragraph as a text file. Then, it indexes all the saved paragraphs building a weights matrix (TFxIDF coding) [1] of the terms in the paragraphs that will be used in the classification process – and in the learning process as well, if needed.

Update model: rebuild the TFxIDF weigths matrix from the exemplary paragraphs and their corresponding labels – that have both been provided by the editor; these will be used to build a new classification model. The classification model is rebuilt only when the editor has new exemplary documents that can eventually generate a more accurate model.

Manually classify paragraph: when selecting a paragraph, the editor can choose to select a label for that paragraph. This label represents the topic in the resource taxonomy to which the paragraph belongs.

Make webpage: goes through the labels – manually assigned by the editor or automatically assigned by the classification model – to select paragraphs grouped by label – topics in the resource taxonomy. When it comes to sorting the labels, the prototype first looks at the manual classification, explicitly set by the editor. If that is not specified, we use the label automatically assigned by the classification model to that paragraph.

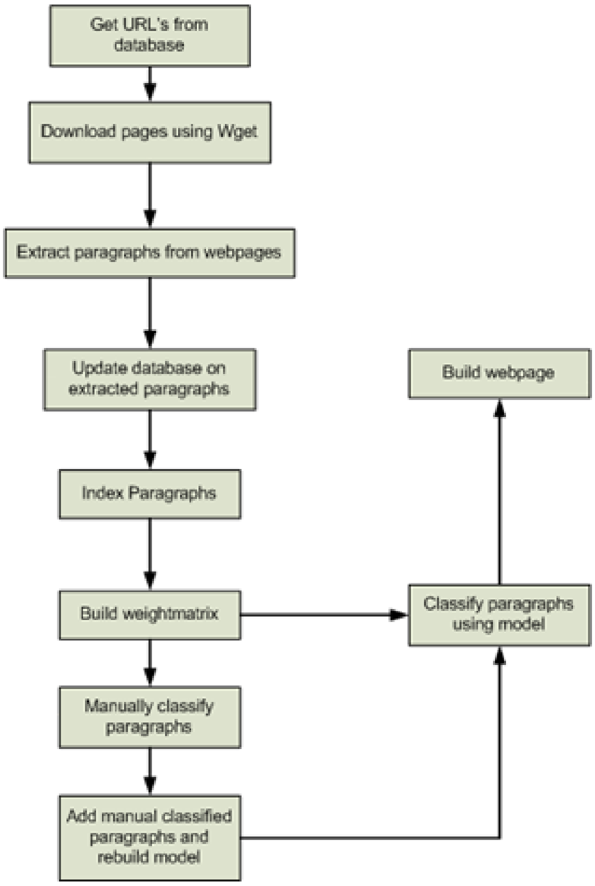
Get URL's from database
Download pages using Wget
Extract paragraphs from webpages
Update database on extracted paragraphs
Index Paragraphs
Build weightmatrix
Manually classify paragraphs
Add manual classified paragraphs and rebuild model
Classify paragraphs using model
Build webpage

Fig. 1. Methodology

3.2. Prototype

Some preliminary work was necessary in order to automate the harvesting task. Initially we had to identify and catalogue the seeds to crawl. In the current work we have relied on three seeds: the Science Daily [15], a provider of the latest research news on Health and Medicine among other fields, the National Institutes Health [13], a medical research agency from the US Department of Health and Human Services and on the Medical Application web page of the On Semiconductors company [14].

The content in each of these seeds has been analyzed to define a model for representing relevant objects appearing in it.

The crawling process, allowing the download of potentially interesting web pages, was designed and tested. In our prototype we are using wget [12] to download seeds.

A method for pre-processing web pages, generating relevant documents (text paragraphs) and representing them on the previously defined model – a TFxIDF document-term matrix – has also been deployed. Text processing is done using the Lucene library [11].

To determine whether recently gathered text paragraphs are relevant or not, we apply a text classifier – based on a classification model that has been previously learned for the resource's taxonomy. Relevant paragraphs are then catalogued on this taxonomy.

These fresh objects are placed in the resource's web page which is automatically updated according to the topic taxonomy.

4. EVALUATION

A preliminary evaluation procedure was carried out based on a taxonomy with three distinct topics (*Medical devices*, *Infectious diseases* and *Chronic illness*) and 100 text paragraphs, previously extracted from the seed web pages. All these documents have been previously labelled by the resource editor so we can compare these manual labels to those automatically generated by the classifier.

We have observed an average accuracy, over these three topics, of 88%. This performance is achieved with a very big reduction in the workload that is required from the resource editor to keep the resource updated. Building a classification model with this accuracy (88% on average) requires the editor to label an average of 32 text paragraphs, a gain of 68 label that are no longer required. Once the classification model is available the site is continuously updated at no extra cost, i.e., without requiring the editor to label any new text paragraph.

5. CONCLUSIONS AND FUTURE WORK

Our evaluation plan relies on a single experiment with a small dataset with 100 text documents. Additional experiments are required to provide more robust conclusions. However, we should notice that this is a real dataset, composed by text snippets directly extracted from real web pages. This brings more realism to our experiment than if it has been performed on a repository dataset.

Our prototype is able to retrieve and organize the content of a web resource, keeping its quality high – close to 90% that of a resource that is kept by the website editor without any automatic assistance – with a very significant reduction in the workload required from the editor.

We are now working on more complex taxonomies and evaluating this methodology against bigger corpora. In another line of work we try to devise algorithms that can reduce further the label complexity – the number of labels that are required to learn the taxonomy.

References

[1] Baeza-Yate, R., Ribeiro-Neto, B., Modern Information Retrieval, Addison Wesley, 1999

[2] Bueno, D., David, A.A., "METIORE: A Personalized Information Retrieval System", Proceedings of the 8th International Conference on User Modeling, Springer-Verlag, 2001

[3] Chakrabarti, S., Dom, B., Raghavan, P., Rajagopalan, S., Gibson, D., Kleinberg, J., "Automatic Resource Compilation by Analyzing Hyperlink Structure and Associated Text", Proceedings of the 7th International World Wide Web Conference, 1998

[4] Chakrabarti, S., Berg, M., Dom, B., "Focused crawling: a new approach to topic-specific resource discovery", Proceedings of the 8th World Wide Web Conference, 1999

[5] Chen,C., Chen, M., Sun, Y., "PVA: A Self-Adaptive Personal View Agent System", Proceedings of the SIGKDD Conference, 2001

[6] Halkidi, M., Nguyen, B., Varlamis, I., Vazirgiannis, M., "Thesus: Organizing Web document collections based on link semantics", The VLDB Journal, 12, pp 320-332, 2003

[7] Lieberman, H., "Letizia: an Agent That Assists Web Browsing", Proceedings of the International Joint Conference on AI, 1995

[8] Liu, B., Chin, C.W., Ng, H. T. , "Mining Topic-Specific Concepts and Definitions on the Web", Proceedings of the WWW Conference, 2003

[9] Mitchell, S., Mooney, M., Mason, J., Paynter, G.W., Ruscheinski, J., Kedzierski, A., Humphreys, K., "iVia Open Source Virtual Library System", D-Lib Magazine, Vol. 9, No. 1, 2003

[10] Mladenic, D., Personal WebWatcher: design and implementation, Technical Report IJS-DP-7472, SI, 1999

[11] http://lucene.apache.org/java/docs/, accessed on September 2010

[12] http://www.gnu.org/software/wget/, accessed on September 2010

[13] http://health.nih.gov, accessed on September 2010

[14] http://www.onsemi.com/PowerSolutions, accessed on September 2010

[15] http://www.sciencedaily.com; accessed on September 2010