OCCS: Enabling the Dynamic Discovery, Harvesting and Delivery of Educational Content from Open Corpus Sources

Séamus Lawless
Trinity College Dublin
slawless@cs.tcd.ie

Lucy Hederman
Trinity College Dublin
lucy.hederman@cs.tcd.ie

Vincent Wade
Trinity College Dublin
vincent.wade@cs.tcd.ie

Abstract

The World Wide Web (WWW) provides access to a vast array of educational content, a great deal of which is ideal for incorporation into eLearning experiences. However sourcing, harvesting and incorporating appropriate content has proven to be a complex and arduous task. This paper introduces a system that enables the discovery and classification of educational content from open corpus sources, such as the WWW, and facilitates the incorporation of such content into eLearning systems. The Open Corpus Content Service (OCCS) discovers, harvests and indexes content through the implementation of a focused web crawler, content classifier and indexer. This reduces the cognitive load placed on the educator by content authoring, allowing them to focus on the pedagogical design of eLearning offerings.

1. Introduction

Motivation: eLearning environments are attempting to respond to the demand for personalised interactive learning experiences by providing increased support for functionality such as personalisation, adaptivity and dynamic learning object generation [1]. As the current generation of eLearning systems attempt to support such features, one of the most significant problems being addressed is the reliance of such systems upon bespoke proprietary content. As a result, there has been a significant shift towards the separation of personalisation and adaptivity information from the physical learning content [2]. This provides the opportunity for eLearning systems to utilise content from external sources in the generation of educational offerings. Content can be selected, regardless of source, and inserted into an eLearning experience in a sequence that suits each individual learner.

Background: Open corpus educational content residing on the WWW is a resource that has yet to be thoroughly exploited in the field of eLearning. For the scope of this research, open corpus content can be defined as any content that is freely available for non-commercial use by the general public or educational institutions. Content can be sourced from web pages, research papers, digital repositories, blogs etc. Many countries are now investing in national digital content repositories to encourage the reuse of learning resources. Merlot [3] in the USA and the NDLR [4] in Ireland are just two examples of such repositories.

Currently utilising web content in eLearning systems requires significant manual effort on the part of the educator. They must source, harvest, describe and incorporate such content. This is an arduous task and leads to unnecessary cognitive load; the educator's efforts are better exerted on the pedagogical aspects of learning experience design. Web content inherently lacks consistency in its structuring. If the vast amount of content available on the WWW is to be leveraged for use in eLearning systems, methods of surmounting this heterogeneity of content must be implemented. Apriori means of discovery, classification and indexing are necessary to make it possible to better group and describe open corpus content.

Contribution: This paper presents the OCCS, a system that leverages content from open corpus sources for use in eLearning systems. This facilitates a reduction in the cognitive load and time pressures placed on educators as a result of content authoring. Educators can thus focus on improving the pedagogical design of educational offerings.

2. OCCS - Open Corpus Content Service

The OCCS enables the discovery, harvesting and indexing of content from open corpus sources. It is provided as an autonomous service, to minimize the impact on the architecture of eLearning environments.

Discovery: The OCCS employs a focused web crawler which conducts crawls that target content by topic. It generates and maintains caches of candidate learning content sourced from the WWW and selected digital repositories. The crawler is based upon Heritrix [5], an open source, extensible, web-scale, archival quality web crawler. Heritrix is implemented in Java and can perform *broad*, *focused*, *continuous* and

978-0-7695-3167-0/08 $25.00 © 2008 IEEE
DOI 10.1109/ICALT.2008.28

experimental crawling. Crawls are executed by incrementally selecting a URI from those scheduled, fetching and analysing the content, archiving the results and adding extracted URIs of interest to the schedule.

Classification: Heritrix is integrated with a language guesser called JTCL [6] and a text classifier called Rainbow [7] to create a focused, rather than general purpose, crawler. These tools govern what content is harvested and added to the content cache.

JTCL dictates what languages are accepted for each cache. We require English language content only for this research. JTCL is a Java implementation of TextCat, a text categorisation library which has been applied to create a written language identification tool. During a crawl the content at each URI is passed to JTCL. If the language of the content is deemed to be English, it is passed on to Rainbow. Rainbow is a statistical text classifier based upon BOW, a collection of C libraries for text mining and retrieval. Rainbow supports all the algorithms in BOW: naïve bayes, SVM, on-line linear etc. Naïve bayes is the classification method used in the OCCS.

Rainbow must be trained in advance of each crawl for the relevant subject area. A combination of keyword files, ODP categories and positive / negative training sets are used during training. The classifier builds a statistical model of the subject area based upon these inputs. During a crawl, Rainbow uses this model to ascertain the relevancy of crawled content to the subject area and assigns all content a relevancy rating.

A relevancy boundary is manually set in advance of each crawl, which dictates how 'on-topic' the content must be to gain inclusion in the cache. If the content is assigned a relevancy rating above this boundary then it is added to the subject specific cache and any extracted links are added to the crawler URI queue. If the content is assigned a rating below the relevancy boundary it, and its subsequent links, are discarded.

Indexing: All of the content cached by the focused crawler is stored in ARC files. NutchWAX [8] is open source software for indexing ARC files. It adapts the fetcher step of Nutch, an open source web crawler, to process ARC files rather than crawl the WWW and has plugins to add extra fields to the index. The Hadoop [9] framework is used to run the indexing jobs. Hadoop is an open source version of Google's mapreduce and GFS untilities.

NutchWAX sequentially imports the cached ARC files and extracts, parses and indexes the content objects or URIs. Upon completion an index is created for the entire collection of ARCs. NutchWAX can be deployed under a servlet such as Tomcat to provide a free-text search interface for the content cache. The results are ranked for relevance according to Nutch.

The Wayback Machine [10] and Wera [11] provide the ability to link a content index with a web archive and allow the visualisation of archived material. The Wayback Machine is not compatible with NutchWAX so Wera was selected to implement query resolution and visualisation of search results. Wera is an archive viewer that allows free-text search across ARC files.

2.1 OCCS Evaluation

To evaluate the OCCS a focused crawl was conducted for the topic "*SQL Programming"* and the quality of the content returned was examined. The content was required to cover all areas of the SQL module of the 3BA25 Information Management course in the BA(Mod) Computer Science Degree in TCD.

Crawl Preparation: The keywords.txt file is used in the training of Rainbow and has a direct influence on what content is accepted into the cache. A query is sent to the Google API for every keyword in the file. The ten results returned from each call are added to Rainbow's training set as positive domain examples. The top result for each API call is also added as a seed URI for the crawl. To generate the keyword file, the index from C.J. Date's "An Introduction to Database Systems" [12] was provided to three domain experts. These experts selected the terms which they felt fell within the scope of the focused crawl. The three distinct selections were then merged to form one list. Upon training completion, a seed file of 131 URIs had been generated. This was manually reviewed. Seed URIs are the start points for the crawler and can dictate path followed. For this research the seed sites selected were largely in the domain of education, as this was the context within which the content would be used.

The second factor that can affect the content cache generation is the Rainbow relevancy boundary. For this evaluation, the relevancy boundary was set at 0.9, or 90%. Only content Rainbow rated as being 90%+ accurate to scope was included in the content cache.

Crawl Execution: The focused crawl ran for a total of 46 hours 41 minutes. At that point 36,196 pages had been added to the content cache and this was deemed sufficient for the purposes of the evaluation. During the course of the crawl 473,259 URIs had been discovered with 370,064 scheduled for download. Of these 67,144 had been downloaded and passed to JTCL for language classification. JTCL labelled 61,527 URIs as English and passed them to Rainbow. Rainbow adjudged 36,196 results to be above the relevancy boundary, these were included in the cache.

Upon crawl completion the cache was indexed. WERA could then be used to search the cache for appropriate content using a free-text interface. Results are displayed according to Nutch ranking. When a result is clicked, WERA uses the NutchWAX arcretriever to link to the content in the cached ARC files and display it on screen.

Evaluation: To evaluate the quality and validity of the cached content it was necessary to create a means of assessment whereby domain experts could semantically analyse search results and document their applicability. The evaluation was conducted via WERA. Selected users performed specific search queries on the index for a pre-defined intended audience. These users then rated each of the top ten results for: relevancy, accuracy, quality, rank at which the results drop below an acceptable standard and rank of the best result.

These ratings were performed using a Likert scale; the most widely used type of psychometric response scale used in questionnaires and survey research. In Likert questionnaires, respondents specify their level of agreement with a statement. Having discussed this with Mary Sharp, a data analysis expert from TCD, a ten point scale was used, as users tend to err on the side of caution and choose a neutral response if one is offered. Three domain experts (SQL lecturers) were selected to analyse and critique the search results for each query performed. This produced interesting and valuable results. The overall relevancy of the content returned by the search engine was very pleasing. There were at least a couple of highly relevant, high quality results for each query performed. This was an essential outcome in the context of this research. It was fundamental that the focused crawls were caching content of both high relevancy for the subject area and also of high quality.

The general pattern was for the relevancy of results to deteriorate lower down the results list; this is normal and, in fact, even desired. In an ideal scenario, the most relevant result to the query, for the intended audience would rank first. However this is extremely difficult to achieve as the intent behind each query and the target audience are not known to the search engine. There was high correlation between assessors on all the characteristics examined. This gives confidence in the results. Some characteristics are more readily measurable than others. The expert can quite easily rate the relevancy of a piece of content, as they are aware of the query performed, the information need and the target audience. However measures such as content quality are more abstract and difficult to quantify.

There was a noticeable reduction in the number of commercial results ranked in the search list compared to a search performed in any of the most popular web search engines. This was a consequence of the seed set used by the focused crawler. The seeds were largely educational in nature, and as these sites tend to link to other educational material rather than commercial sites, the crawler remained largely within the educational domain. This seed set was chosen as the content required was educational in nature.

3. Conclusion

The ability to leverage the vast amounts of knowledge and information available on the WWW would signify a step forward in the evolution of eLearning systems. It would enable learners and educators to incorporate content sourced from the WWW and digital repositories into their eLearning offerings. The OCCS is a novel and powerful tool that facilitates a reduction in the cognitive load and time pressures placed on educators as a result of content authoring. It enables educators to focus on the pedagogical aspects of educational offering design through a first attempt to tap into the massive potential of open corpus educational content for eLearning.

4. Acknowledgements

This research is funded by the Embark Initiative of the Irish Research Council for Science Engineering and Technology; funded by the NDP.

5. References

[1] Brusilovsky, P. “Adaptive Hypermedia”. In User Modelling and User-Adapted Interaction, Springer, 01.

[2] Henze, N.; Nejdl, W. “Adaptation in Open Corpus Hypermedia”. *IJAIED* 12 (2001)

[3] Merlot. http://www.merlot.org

[4] NDLR. http://www.learningcontent.edu.ie

[5] Heritrix - http://crawler.archive.org

[6] JTCL - http://textcat.sourceforge.net/

[7] Rainbow – Statistical Text Classification. http://www.cs.cmu.edu/~mccallum/bow/rainbow/

[8] NutchWAX – http://archive-access.sourceforge.net/

[9] Hadoop Framework. http://lucene.apache.org/hadoop/

[10] Wayback Machine http://www.archive.org/

[11] Wera – Web Archive Access http://archive-access.sourceforge.net/projects/wera/

[12] Date, C.J. “An Introduction to Database Systems” (7th Ed.). Addison-Wesley Longman. Boston, MA. 1999