An Ontology-based Web Crawling Approach for the Retrieval of Materials in the Educational Domain

Mohammed Ibrahim[1][a] and Yanyan Yang[2][b]
[1]School of Engineering, University of Portsmouth, Anglesea Road, PO1 3DJ, Portsmouth, United Kingdom
[2]School of Computing, University of Portsmouth, Anglesea Road, PO1 3DJ, Portsmouth, United Kingdom

Keywords: Web Crawling, Ontology, Education Domain.

Abstract: As the web continues to be a huge source of information for various domains, the information available is rapidly increasing. Most of this information is stored in unstructured databases and therefore searching for relevant information becomes a complex task and the search for pertinent information within a specific domain is time-consuming and, in all probability, results in irrelevant information being retrieved. Crawling and downloading pages that are related to the user's enquiries alone is a tedious activity. In particular, crawlers focus on converting unstructured data and sorting this into a structured database. In this paper, among others kind of crawling, we focus on those techniques that extract the content of a web page based on the relations of ontology concepts. Ontology is a promising technique by which to access and crawl only related data within specific web pages or a domain. The methodology proposed is a Web Crawler approach based on Ontology (WCO) which defines several relevance computation strategies with increased efficiency thereby reducing the number of extracted items in addition to the crawling time. It seeks to select and search out web pages in the education domain that matches the user's requirements. In WCO, data is structured based on the hierarchical relationship, the concepts which are adapted in the ontology domain. The approach is flexible for application to crawler items for different domains by adapting user requirements in defining several relevance computation strategies with promising results.

1 INTRODUCTION

Vast amounts of information can be found on the web (Vallet et al. 2007) onsequently, finding relevant information may not be an easy task. Therefore, an efficient and effective approach which seeks to organize and retrieve relevant information is crucial (Yang 2010). With the rapid increase of documents available from the complex WWW, more knowledge regarding users' needs is encompassed. However, an enormous amount of information makes pinpointing relevant information a tedious task. For instance, the standard tools for web search engines have low precision as, typically, some relevant web pages are returned but are combined with a large number of irrelevant pages mainly due to topic-specific features which may occur in different contexts. Therefore, an appropriate framework which can organize the overwhelming number of documents on the internet is needed (Pant *et al.*, 2004). The educational domain is one of the domains that have been affected by this issue(Almohammadi et al. 2017). As the contents of the web grow, it will become increasingly challenging especially for students seeking to find and organize the collection of relevant and useful educational content such as university information, subject information and career information(Chang *et al.*, 2016). Until now, there has been no centralized method of discovering, aggregating and utilizing educational content(Group 2009) by utilising a crawler used by a search engine to retrieve information from a massive number of web pages. Moreover, this can also be useful as a way to find a variety of information on the internet(Agre & Dongre 2015). Since we aim to find precise data on the web, this comprehensive method may not instantly retrieve the required given the current size of the web.

Most existing approaches towards retrieval tech-

[a] https://orcid.org/0000-0002-9976-0207
[b] https://orcid.org/0000-0003-1047-2274

Ibrahim, M. and Yang, Y.
An Ontology-based Web Crawling Approach for the Retrieval of Materials in the Educational Domain.
DOI: 10.5220/0007692009000906
In *Proceedings of the 11th International Conference on Agents and Artificial Intelligence (ICAART 2019)*, pages 900-906
ISBN: 978-989-758-350-6

niques depend on keywords. There is no doubt that the keywords or index terms fail to adequately capture the contents, returning many irrelevant results causing poor retrieval performance(Agre & Mahajan 2015). In this paper, we propose a new approach to web crawler based on ontology called WCO, which is used to collect specific information within the education domain. This approach focuses on a crawler which can retrieve information by computing the similarity between the user's query terms and the concepts in the reference ontology for a specific domain. For example, if a user seeks to retrieve all the information about master's courses in computer science, the crawler will be able to collect all the course information related to the specific ontology designed for the computer science domain.

The crawling system described in this paper matches the ontology concepts given the desired result. After crawling concept terms, a similarity ranking system ranks the crawled information. This reveals highly relevant pages that may have been overlooked by focused standard web crawlers crawling for educational contents while at the same time filtering redundant pages thereby avoiding additional paths.

The paper is structured into sections. Section 2 reviews related work and background; Section 3 introduces the proposed approach to architecture. In section 4 the experiment and the results are discussed. Section 5 provides a conclusion and recommendations for future work.

2 RELATED WORK

A web crawler is a software programme that browses the World Wide Web in a systematic, automated manner (Hammond *et al.*, 2016). There has been considerable work done on the prioritizing of the URL queue to effect efficient crawling. However, the performance of the existing prioritizing algorithms for crawling does not suit the requirements of either the various kinds or the levels of the users. The HITS algorithm proposed by Kleinberg (Pant & Srinivasan 2006) is based on query-time processing to deduce the hubs and authorities that exist in a sub graph of the web consisting of both the results to a query and the local neighbourhood of these results. The main drawback of Kleinberg's HITS algorithm is its query-time processing for crawling pages. The best-known example of such link analysis is the Pagerank Algorithm which has been successfully employed by the Google Search Engine (Gauch *et al.*, 2003). However, Pagerank suffers from slow computation due to the recursive nature of its algorithm. The other approach for crawling higher relevant pages was by the use of neural networks (Fahad et al. 2014) but even this approach has not been established as the most efficient crawling technique to date. All these approaches, based on link analysis, only partially solve the problem. Ganesh et al.(Ganesh et al. 2004) proposed a new metric solving the problem of finding the relevance of pages before the process of crawling to an optimal level. Researchers in (Liu *et al.*, 2011) present an intelligent, focused crawler algorithm in which ontology is embedded to evaluate a page's relevance to the topic. Ontology is "a formal, explicit specification of a shared conceptualization" (Gauch *et al.*, 2003). Ontology provides a common vocabulary of an area and defines, with different levels of formality, the meaning of terms and the relationships between them (Krisnadhi 2015). Ontologies were developed in Artificial Intelligence to facilitate knowledge sharing and reuse. They have become an interesting research topic for researchers in Artificial Intelligence with specific reference to the study domain regarding knowledge engineering, natural language processing and knowledge representation. Ontologies help in describing semantic web-based knowledge management architecture and a suite of innovative tools for semantic information processing (Tarus *et al.*, 2017). Ontology-based web crawlers use ontological concepts to improve their performance. Hence, it may become effortless to obtain relevant data as per the user's requirements. Ontology is also used for structuring and filtering the knowledge repository.

The ontology concept is used in numerous studies (Gauch *et al.*, 2003; Agre and Mahajan, 2015). Gunjan and Snehlata (Agre & Dongre 2015) proposed an algorithm for an ontology-based internet crawler which retrieved only relevant sites and made the best estimation path for crawling that helped to improve the crawler performance. The proposed approach deals with information path and domain ontology, finding the most relevant web content and pages according to user requirements. Ontology was used for filtering and structuring the repository information.

3 PROPOSED SYSTEM

Our proposed approach seeks to apply crawling to educational content, such as university course information, and sort it into a database through the calculation of the hierarchy similarity between the user query and the course contents. The crawler

consists of several stages; it begins with construction domain ontology which it uses as a reference of similarity between the user query and the web contents. The user query adjusts to generate query based ontology concepts and uses Term Frequency-Inverse Document Frequency (TF-IDF) for identifying terms for query expansion.

Firstly, we describe how information retrieval can be achieved in the ontology. For instance, if D is the number of documents annotated by concepts from an ontology O, the document is represented by vector d of concept weights. For each concept x ∈ O annotating d, dx is the importance of x in document d. It can be computed by using the TF-IDF algorithm as shown in Eq. 1:

$$d_x = \frac{ferq_{x,d}}{max_y freq_{x,y}} log \frac{|D|}{n_x} \quad (1)$$

Where $freq_{x,d}$ is the number of occurrences of x in d, $max_y freq_{y,d}$ is the frequency of the most repeated instance in d, and nx is the number of documents annotated by x, then cosine similarity between the query and the document is used as the relevance score for ranking the documents as shown in Eq.2.

$$cosine\ similarity(d,q) = \frac{\sum d_i . q_i}{\sqrt{\sum {d_i}^2} . \sqrt{\sum {q_i}^2}} \quad (2)$$

Where d the *ith* term in the vector for document and q the ith term in the vector for the query. The ontology-based query used as an input to the search engine module. The output of this phase is a set of documents which would be used for the crawling system and furthermore operate as a way by which to check all the web pages for validity (i.e. HTML, JSP etc.). If it is valid, it is parsed, and the parsed content is matched with the ontology and, if the page matches, it will be indexed otherwise, it will be discarded.

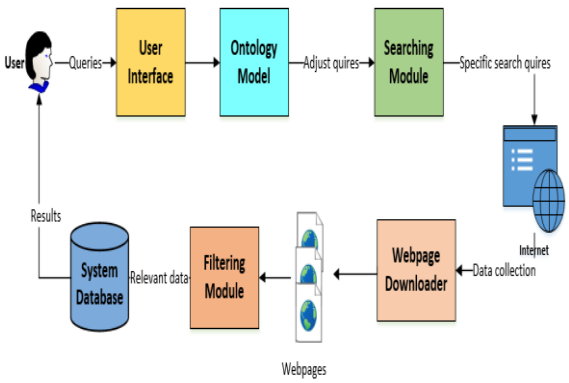

Figure 1: WCO crawler main architecture.

architecture of the proposed approach is illustrated in Fig.1. The user interacts with the crawler using a simple interface designed to allow the user query insert.

3.1 Ontology Model

This model consists of two main parts. Firstly, in building reference ontology, a well-known open source platform, Protégé (Ameen *et al.*, 2012), is used to build the reference ontology file. Protégé is a java based tool providing a graphic user interface (GUI) to define ontologies. The concepts in the user search queries will be matched with the concepts in the reference ontology file. The second aspect is to assign weights to the concepts and their levels in the ontology as shown in Table 1. A case study of a computer sciences course hierarchy (Programs 2013) is used. Computer sciences are the root class in our ontology and has seven subclasses (information system, computer science, are the root class in our ontology, and with seven subclasses (Information system, Computer science, Artificial intelligence, Health information, Computer generated visual and audio effects, Software engineering and Games) as shown in Fig.2.

Each of the subclasses has a further subclass, for instance artificial intelligent is further subdivided into machine learning, cognitive modelling, neural computing, knowledge representation, automated reasoning, speech &natural language processing, computer vision as shown in Fig. 3.

In order to clearly understand the relationship between the ontology concept and their levels in the ontology we apply the following definitions:

Definition 1: Let C be a concept, L the levels of the concepts of an ontology and *LW* the weight of the level. *F* is the frequency of the concept in the URL parsed contents. The concept weight in each level can be obtained using Eq. 3.

$$Concept\ Weight\ \ Level\ (CWL) = LW \times F \quad (3)$$

Definition 2: Let SC be the set of concepts of an ontology. We define the levels of a concept ci as: levels(ci) = {l ∈ SC|l ∈ hyponyms(ci) ∧ l is a level}, where l is a level if hyponyms(l) = ∅.

The upper level in the ontology is given a higher weight than the lower level, for example, the term computer sciences in the URL is more specific to the field of study than the term information system which is also a field within the computer science domain shown in Table 1.

Table 1: Ontology weight table for Computer Sciences courses.

Level of concepts	Ontology terms in the course title	Weight (W)
1	Computer sciences	0.5
2	Information systems	0.3
2	Software engineering	0.3
2	Artificial intelligence	0.3
2	Health informatics	0.3
2	Games	0.3
3	Computer architectures	0.2
3	Operating systems	0.2
3	Information modelling	0.2
3	Systems design methodologies	0.2
3	Systems analysis & design	0.2
3	Databases	0.2
3	Computational science foundations	0.2
3	Human-computer interaction	0.2
3	Multimedia computing science	0.2
3	Internet	0.2
3	e-business	0.2
3	Information modelling	0.2
3	Systems design methodologies	0.2
3	Systems analysis & design	0.2

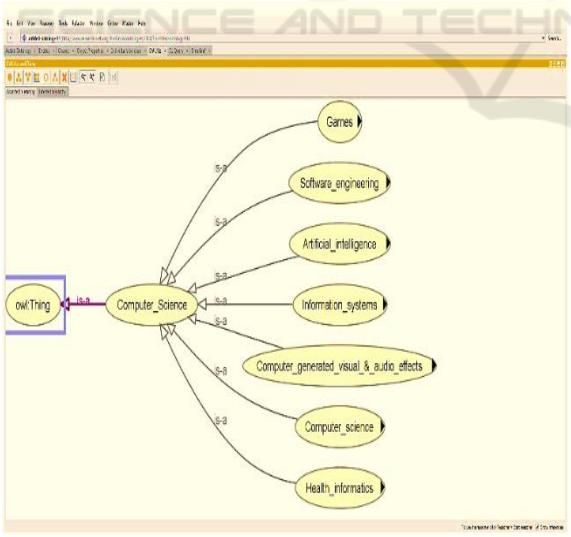

Figure 2: OWL Viz view of Computer Science Classes generated by Protégé.

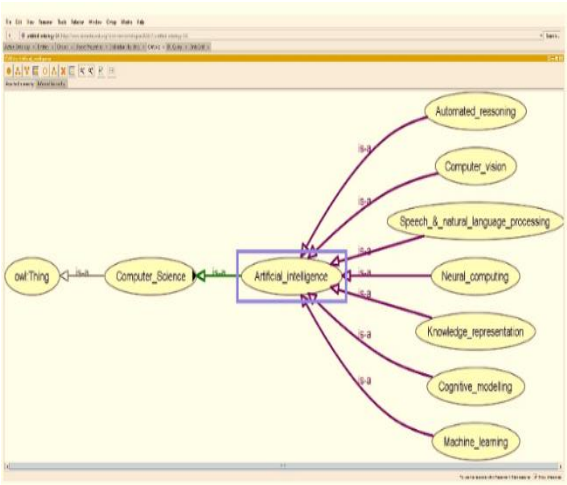

Figure 3: OWL Viz view of Artificial Intelligent Classes generated by Protégé.

Crawling Algorithm Implementation

The main source of educational information regarding university courses in the United Kingdom is UCAS (ucas.com 2018). Before we begin the crawler process, it is necessary to identify the information that needs to be crawled because each URL contains much information that is not relevant. In this paper, course information includes all the important information that a student may require relating to the course such as course title, the field of study, major subject, level, fee, university location, entry requirements and the language of the course). Fig. 4 shows the crawling algorithm steps of the crawler process:

1. User gives query terms through GUI of a crawler
2.Obtain the seed URL
3.If the webpage is valid, that is it of the defined type
4.Add the page to the queue.
5.Parse the contents.
6.Check the similarity of the parsed contents with the reference ontology concepts by using Eq.1
7.If the page is relevant then
8. Add to the index
9. Otherwise the page does not need to be considered further
10. Get the response from the server
11.Add the webpage to the index
12. Get the response from the server and, if it is OK, then
13.Read the Protégé file of ontology and match the content of a webpage with the terms of the ontology.
14. Count the Relevance Score of the web page as defined by the algorithm and add the webpage to the index and the cached file to a folder.

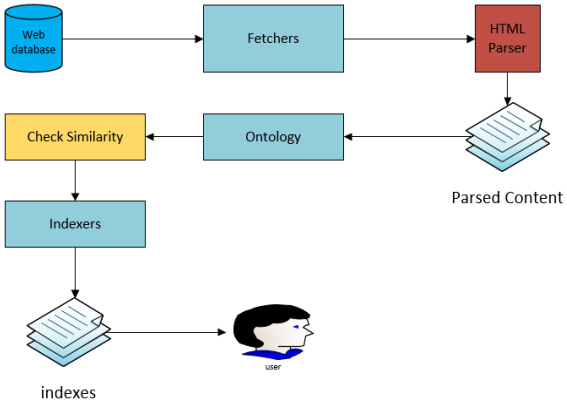

Figure 4: Crawling algorithm architecture.

3.2 Filtering Module

Although results are retrieved through centred crawling, a mechanism is needed to verify the relevance of those documents to the required domain of interest. All pages stored within the repository will be evaluated and their relevance scores calculated thereby removing unrelated documents. A weight table containing weights for every term within the metaphysics is used. In the weight table, weights are assigned to every term in our ontology. Terms common to several domains are assigned a lower weight, whereas terms that square measures specific to bound domains take a higher weight. The task of assigning the weight is established by knowledge consultants. Using this approach, the link pages found within the page repository which are examined and crawled support the relevancy of domain ontology.

4 IMPLEMENTATION AND EVALUATION

4.1 Experimental Setup

The system has been developed using a Java framework (version JDK1.8) as well as the NetBeans (version 8.02) being utilized as a development tool on a Windows platform as shown in Fig.5. The system runs on any standard machine and does not need any specific hardware in order to run effectively. We downloaded approximately 6000 webpages from UCAS and recorded their links in our database. Our experiment focuses on a crawling topic of a "computer sciences course". The reference ontology model is created using a Protégé tool. A web based user interface is used to prompt the user to give a query as shown in Fig.6. In Table 2 a sample of

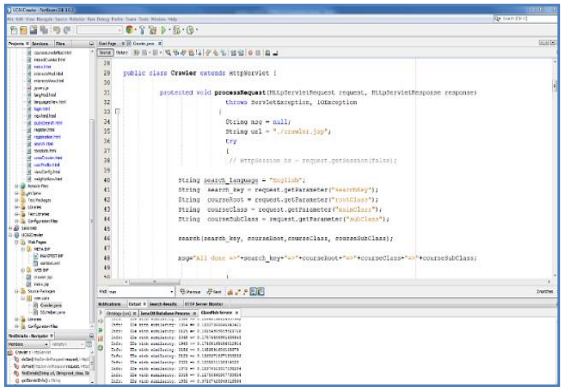

Figure 5: Snapshot of crawler code in NetBeans platform.

Table 2: Sample of results WCO Crawler for course information.

Feature	Value
Course title	Artificial Intelligence
Course qualification	MSc
Course URL	https://digital.ucas.com/courses/details?coursePrimaryId=7fea172a-efdd-4c02-9950-edf34da09124&academicYearId =2018
Course description	Artificial Intelligence (AI) forms part of many digital systems. No longer is AI seen as a special feature within software, but as an important development expected in modern systems. From word-processing applications to gaming, and from robots to the Internet of Things, AI tends to be responsible for controlling the underlying behaviour of systems. Such trends are forecast to grow further.
University name	University of Aberdeen
Field of study	Information technology
Main subject	Computer science
Major subject	Artificial Intelligence
Course UK Fee	£6,300
Course location	University of Aberdeen, King's College, Aberdeen,AB24 3FX
Entry requirement	Our minimum entry requirement for this programme is a 2:2 (lower second class) UK Honours level (or an Honours degree from a non-UK institution which is judged by the University to be of equivalent worth) in the area of Computing Science. Key subjects you must have covered: Java, C, C++, Algorithms problem solving and Data Structures.

information can be seen which has been crawled from the UCAS website with reference to a computer sciences master course entitled *Artificial Intelligence.*

This information, stored in the MySQL database, will be available to use for application in any research work.

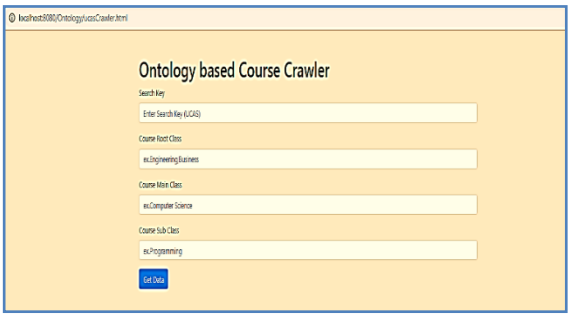

Figure 6: WCO crawler interface.

4.2 Evaluation Metrics

In order to measure the performance of the proposed approach, a harvest rate metric has been used to measure the crawler performance and a recall metric has been used to evaluate the performance of the webpage classifier.

The harvest rate, according (Pant & Menczer 2003)is defined as the fraction of the web pages crawled that are relevant to the given topic. This measures how well irrelevant web pages are rejected. The formula is given by

$$Harvest\ rate = \frac{\sum_{i \in V} r_i}{|V|} \quad (4)$$

where V is the number of web pages crawled by the focused crawler in current; ri is the relevance between the web page i and the given topic, and the value of ri can only be 0 or 1. If relevant, then ri = 1; otherwise ri = 0. in the Fig.7 performance comparison between the proposed crawler which is based ontology and a traditional crawler.

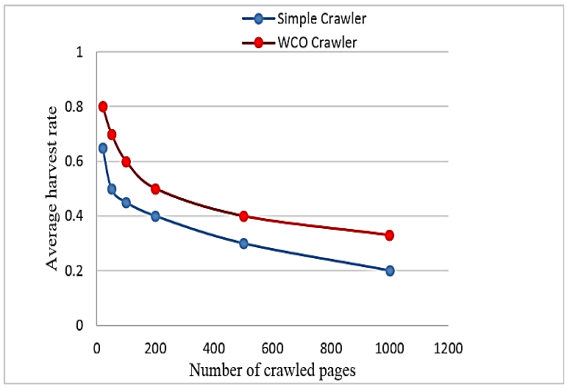

Figure 7: Performance comparison average harvest rate between proposed crawler which is based ontology and traditional crawler.

In Fig. 7, the x-axis denotes the number of crawled web pages and the y-axis denotes the average harvest rates when the number of crawled pages is N. According to Fig 7, the number of crawled web pages increases and we see that the number of crawled web pages of WCO is higher than that of a simple crawler which has not used ontology in the crawling process. Moreover, the harvest rate of a simple crawler is 0.42 at the point that corresponds to 100 crawled web page in Fig 7. However, the values indicate that the harvest rate of the WCO is 0.6 and 1.3 times larger than that of the simple crawler.

The recall metric is the fraction of relevant pages crawled and used to measure how well it is performing in finding all the relevant webpages. To further understand how we can apply the recall metric, let RS be the relevant set in the web and St be the set of the first pages crawled. The formula of the recall metric will be as follows:

$$Recall\ metrics = \frac{|RS \cap S_t|}{|RS|} \quad (5)$$

Fig. 8 presents a performance comparison of the average recall metrics for a simple crawler and the proposed crawler for 5 different topics. The x-axis in Fig.8 denotes the number of crawled web pages and the y-axis denotes the average recall metrics when the number of crawled pages is N. We realized that the average recall of the WCO is higher than that of the simple crawler.

Based on the performance evaluation metrics, the harvest rate and the recall metrics, we can conclude that the WCO has a higher performance level than that of a simple crawler.

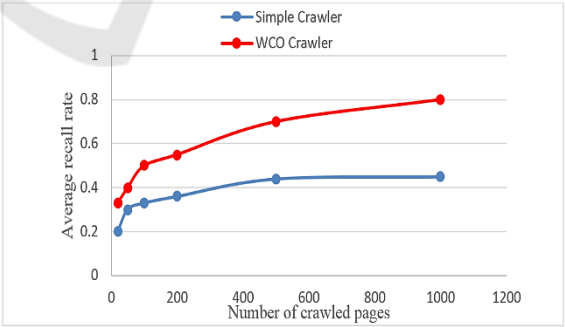

Figure 8: Performance comparison average recall metrics rate between proposed crawler which is based ontology and traditional crawler.

5 CONCLUSION

The main aim of our paper is to retrieve relevant web

pages and discard those that are irrelevant from an educational web page. We have developed an ontology-based crawler, called WCO, which retrieves web pages according to relevance and which discards the irrelevant web pages using an algorithm. In this study, a concept of ontology provided a similarity calculation of levels of the concepts in the ontology and the user query and the relationship between them were used. It is therefore intended that this crawler will not only be useful in exploiting fewer web pages, such that only relevant pages are retrieved, but will also be an important component of the "Semantic Web", an emerging concept for future technology. The evaluation results show that the ontology based crawler offers a higher performance than that of a tradition crawler. This improved crawler can also be applied to areas such as recruitment portals, online music libraries and so forth.

ACKNOWLEDGMENT

The authors wish to express their gratitude to the Iraq government through the Higher Committee Education Development program (HCED) for their financial support in this study. Also, would like to thanks the school of Engineering and school of computing at the University of Portsmouth for their contribution to participate in the experiment. Finally, the authors want to thank the anonymous reviewers and editors, whose insightful comment and corrections made a valuable contribution to this article.

REFERENCES

Agre, G. and Dongre, S. 2015.A Keyword Focused Web Crawler Using Domain Engineering and Ontology. *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no. 3, pp. 463–5.

Agre, G.H. and Mahajan, N. V. 2015. Keyword focused web crawler. *2nd International Conference on Electronics and Communication Systems, ICECS 2015*, pp. 1089–92.

Almohammadi, K., Hagras, H., Alghazzawi, D. and Aldabbagh, G. 2017. A survey of artificial intelligence techniques employed for adaptive educational systems within e-learning platforms. *Journal of Artificial Intelligence and Soft Computing Research*, vol. 7, no. 1, pp. 47–64.

Ameen, A., Khan, K.U.R. and Rani, B.P. 2012. Creation of ontology in education domain. *Proceedings - 2012 IEEE 4th International Conference on Technology for Education, T4E 2012*, no. May 2015, pp. 237–8.

Chang, P.C., Lin, C.H. and Chen, M.H. 2016. A hybrid course recommendation system by integrating collaborative filtering and artificial immune systems. *Algorithms*, vol. 9, no. 3.

Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A.Y., Foufou, S. and Bouras, A. 2014. A survey of clustering algorithms for big data: Taxonomy and empirical analysis. *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 3, pp. 267–79.

Ganesh, S., Jayaraj, M., SrinivasaMurthy, V.K. and Aghila, G. 2004. Ontology-based web crawler. *International Conference on Information Technology: Coding Computing, ITCC*, vol. 2, pp. 337–41.

Gauch, S., Chaffee, J. and Pretschner, A. 2003. Ontology-based personalized search and browsing. *Web Intelligence and Agent Systems: An international journal*, vol. 1, pp. 219–34.

Group, D.E. 2009. *Leveraging Content from Open Corpus Sources for Technology Enhanced Learning*. *October*, viewed 12 November 2018, <https://www.scss.tcd.ie/Seamus.Lawless/papers/thesis.pdf>.

Hammond, D.A., Smith, M.N. and Meena, N. 2016. A Brief History of Web Crawlers. *Chest*, vol. 149, pp. 1582–3.

Krisnadhi, A.A. 2015. Ontology Pattern-Based Data Integration. *Phd*, p. 217.

Liu, Z., Du, Y. and Zhao, Y. 2011. Focused Crawler Based on Domain Ontology and FCA. *Journal of Information and Computational Science*, vol. 8, no. 10, pp. 1909–17.

Pant, G. and Menczer, F. 2003. Topical Crawling for Business Intelligence. *Proceedings of 7th European Conference on Research and Advanced Technology for Digital Libraries ECDL 2003*, vol. 2769, Springer, Berlin, Heidelberg, pp. 233–44, .

Pant, G. and Srinivasan, P. 2006. Link contexts in classifier-guided topical crawlers. *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 1, pp. 107–22 .

Pant, G., Srinivasan, P. and Menczer, F. 2004. *Crawling the Web*. *Web Dynamics* .

Programs, U.D. 2013. Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science.

Tarus, J.K., Niu, Z. and Mustafa, G. 2017. Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning. *Artificial Intelligence Review*, pp. 1–28.

Ucas.com 2018. The Universities and Colleges Admissions Service in United Kingdom. viewed 15 October 2018, <https://www.ucas.com/>.

Vallet, D., Castells, P., Fernández, M., Mylonas, P. and Avrithis, Y. 2007. Personalized content retrieval in context using ontological knowledge. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 3, pp. 336–45.

Yang, S.Y. 2010. Developing an ontology-supported information integration and recommendation system for scholars. *Expert Systems with Applications*, vol. 37, no. 10, pp. 7065–79.