ENHANCING THE RETRIEVAL OF SOCIAL WEB-BASED E-LEARNING CONTENT USING SEMANTICS EXTRACTED FROM DBPEDIA AND WORDNET ONTOLOGIES

[1]AMMAR ALNAHHAS, [2]BASSEL ALKHATIB, [3]AHMAD OMAR

[1]Faculty of Information technology engineering – Damascus University and Faculty of Informatics and Communication Engineering- Arab International University-Syria
[2]Assistant Professor at the Faculty of Informatics and Communication Engineering- Arab International University-Syria and the Faculty of Information Technology Engineering-Damascus University
[3]Damascus University – Damascus Syria

Emails: [1]eng.a.alnahhas@gmail.com, b-khateeb@aiu.edu.sy, [3]eng.aomar007@gmail.com

ABSTRACT

As E-learning Tools and techniques are becoming more common and compelling, many researches have emerged lately that aims at making it more flexible and applicable. Besides, the content is getting very large nowadays, so that it is very important to develop a more accurate and robust search techniques that help users find the best learning materials that exists all along the web specially on social learning websites.

In this paper we propose a new method to collect, index and retrieve learning materials, a collection algorithm is presented that can bring together content from various sources. We present a semantic indexing method that aims at weighting words of the document based on both DBpedia and Wordnet Ontologies, which proves more accurate results according to the analysis and comparison that are shown in this paper.

Keywords: *E-learning, DBpedia, WordNet, Search Engine, Social web.*

1. INTRODUCTION

E-learning systems has developed significantly in the last decade, it was spread as the extensive use of web technologies and internet services. E-learning provides a chance to many people to benefit through the web techniques to get the educational material and improve their academic level.

E-Learning becomes more accessible to almost everyone, and become quickly coming to the forefront of those solutions. Technology has proven to be a great enabler for learning by granting wider access and facilitating continuing education throughout an organization. It provides an infrastructure through which companies can develop interactive and engaging platforms to deliver skill-building, training information [1].

There are pros of eLearning, as Flexibility, it makes the eLearning content be done in sections, to fit around your already busy schedule. Also, we can describe E-Learning as Lower cost technology, it allows you to learn from anywhere. Mobility, considered one of the most advantage on E-Learning, instead of the confinement of a traditional classroom, you now have the flexibility to learn in any location you'd like [2].

There are many available web based e-learning systems, that follows well known LMS standards such as SCORM [3], where SCORM specifications are based on various other industry standards and specifications.

As there are now many online e-learning sources, it is becoming more difficult for students to find the exact match for some topic they are interested in. traditional search engines can help users find documents that contains some words of search queries which is not sufficient in many cases, some documents may be related to some field of study but it may not contain the keywords of this field explicitly which leads to the failure of traditional search engine to link queries related to this topic with the documents. Nevertheless, most concurrent search engines are still keyword based but many research works have been conducted in the field of adding semantics to IR systems.

It is obvious that the latent relation between documents and queries needs some deep sematic analysis of the document in order to extract this relation, this way we can help improve the search results of users and make it easy for them to find suitable document regardless of literal identification of contents.

Many researchers conducted their work in the field of searching and finding suitable content in E-learning management system, and many others tried to improve general purpose IR systems with semantic features, the authors of [4] tried to improve query expansion for information retrieval using Wikipedia, because they found two problems in using query expansion term: term relationships of listed terms are limited and unlisted terms have no expansion terms. To solve this issue, they depend on Wikipedia to present a new approach to extract more term relationships from Markov network for query expansion. In term relationship extracted from Wikipedia corpus is superimposed to the basic Markov network that is pre-built using the single local corpus. Therefore, a new larger Markov network is built with more and richer term relationship for unlisted terms as well as listed terms. Evaluation is performed on three standard information retrieval corpuses including ADI, CISI and CACM. Experimental results show that the proposed technique of superimposed Markov network is effective to select more and confident candidates for query expansion. But this solution is not enough, because they did not solve the problem of independent terms case which will not be included in any Markov “related” networks and they did not use and semantics of Wikipedia, instead they used statistical Markov network.

Researchers in [5] work in information retrieval by semantic similarity by computing the similarity between concepts. The concepts extracted from each source must be compared in terms of their meaning (i.e. semantically). This paper deals with a certain aspect of Semantic Web and semantics, that of semantic text association and text semantics respectively. They demonstrate that it is possible to approximate algorithmically the human notion of similarity using semantic similarity and to develop methods capable of detecting similarities between conceptually similar documents even when they don’t contain lexically similar terms. The lack of common terms in two documents does not necessarily mean that the documents are not related. Computing text similarity by classical information retrieval models (e.g., Vector Space, Probabilistic, Boolean). This paper take contribution is to experiment with several semantic similarity methods for computing the conceptual similarity between natural language terms using WordNet. The experimental results indicate that it is possible for these methods to approximate algorithmically the human notion of similarity reaching correlation (with human judgment of similarity) up to 30% for WordNet. But this approach even its successes with terms, it fails under phrases and sentences, besides WordNet does not provide good similarity criteria that can link word semantically, this is because the relations in Wordnet are very limited, they can only show generalization and specification relations which may not link many words that are semantically related.

This failure with phrases and sentences makes [6] work on semantically enhanced information retrieval with an ontology-based approach. They try to solve the limitation of keywords in query processing and information retrieval techniques, by depending on conceptual search, understood as searching by meanings rather than literal strings. The query execution returns a set of tuples that satisfy the SPARQL query. They then extract the semantic entities from those tuples and access the semantic index to collect all the documents in the repository that are annotated with these semantic entities. Once the list of documents is formed, the search engine computes a semantic similarity value between the query and each document, using an adaptation of the classic vector space IR model. Then semantic search result, it is assumed that the information available in standard Web pages (the document base) is indexed using the semantic knowledge, and weighting process applied based on the frequency of the occurrences of each semantic entity within the document. But the drawback in this approach is knowledge incompleteness. This problem refers to the need of retrieving accurate results when the semantic information is not available or incomplete because of lack of resources that contains the keywords.

Authors of [7] work on an ontology approach for Semantic information retrieval on the web, a system for semantic information retrieval has been presented. In this research they proposed a system for information retrieval based on ontologies, dynamic semantic network, and lexical chains, defining a strategy for scoring and ranking results by means of a measure of semantic relatedness between words. The proposed semantic relatedness metric performs optimally compared with other metrics in a general test set. we think that using this test sets and subject groups will make this approach inaccurate in many cases with problems related to the representation and organization of knowledge in the documents. Researchers in [8] created an RDF model of the IR data with the aim of enhancing their discoverability and easing their connections with lessons and documents, then provide a methodology for automatically enriching the data by exploiting relevant external entities that found the semantic matching by depending on people relative expertise on given subject, so the relevance score measures the interest of researcher, but this assumption find that the more a person works on a topic, the more knowledgeable he is. And this approach had the problem of number of users “contribution authors”, besides the results are measured on small data sets.

In [9] they describe a system for semantic web content mining; which depend on web pages categorized as the web is searched to derive an inherently up to date listing. This categorization used to allow the user to submit a query to the system; apart from keywords and sentences to be found as a whole, then indicates the depth at which each site has to be inspected, with the Language of pages to be found and the context that indicates the search area. This approach will reduce the proper results that may appear to the user, who grades pages according to the relevance with the user defined context. In [10] a review on ontology based information retrieval system is conducted, they found that the techniques for content description and query processing in Information Retrieval (IR) are based on keywords, and therefore provide limited capabilities to capture the conceptualizations associated with user needs and contents. Aiming to solve the limitations of keyword-based models, the idea of conceptual search, understood as searching by meanings rather than literal strings. They suggest many scenarios which depends on specific topics to use as keywords or specialized search engine.

After going through the literature, we can observe that most of the related work did not use the latent semantics inside the documents to help improving the search results especially when the content is unstructured like the E-learning case, we can observe that some researchers tried to use semantic web technologies as they try to express queries as SPARQL queries, while the others tried to use query expansion but none of them has ever used any semantic relations; some researchers tried to use WordNet to enrich the query keywords but relations in Wordnet are lexical and never express semantic relations while the latest research in query expansion [4] uses Wikipedia as a corpus where they used a Markov model to express relations, that is, a statistical approach and did not use DBpedia sematic relations that can perform better as we will present in our work. So, we can classify the literature into three divisions:

- Research that depends on semantic web approach, where documents should be structured or has meta data, and queries should be formatted in standard query languages, whereas we are dealing with unstructured text that most of e-learning web content follow.
- Research that tries to expand queries using similarity from lexical database such as WordNet, where query is enriched by terms that are lexically similar to the user terms.
- Research that tries to expand queries using statistical model inferred from a corpus, the terms are expanded by most frequently accompanying terms.

DBpedia as a big source of shared knowledge can be used to find similar terms to query terms, this way we can expand the query with the related terms extracted from DBpedia, it is clear that if we add more terms we can get higher precision in the result but it can also decrease the recall dramatically and the ratio of false positive will be high, to overcome this problem we introduce the usage of WordNet as a lexicon that can find the specificity of a word, the more the word is specific the more it is important and can expressive, in this paper we propose to use WordNet to add weight to the terms that can enrich the query. So, Our paper presents a new method for indexing and providing content in e-learning systems, we collect content from as many sources as available, index this content in our system, and use semantic based technique to match learner query with the content indexed earlier, we use DBpedia ontology [11] and WordNet [12] to enhance the matching of the query, DBpedia is a crowd-sourced community effort to extract structured information from Wikipedia and make this information available on the Web. DBpedia allows you to ask sophisticated queries against Wikipedia, and to link the different data sets on the Web to Wikipedia data.

Our paper is organized as follows, section 2 shows our main system architecture, we show the main algorithm of indexing documents in section 3, section 4 presents the semantic matching algorithm, we show our implementation and discuss the results in section 5, and finally section 6 concludes the paper.

2. THE PROPOSED ARCHITECTURE

The main goal of our work is to build a system that can collect E-learning content that follows the SCORM standard, then we index this content in our system, so that the learner can query the content and find suitable results.

To make it easier for learner we propose to build a centralized e-learning search engine, this engine is able to collect content from many LMS systems and provide a search facility as it indexes and stores the content, in practical we find that Moodle [13] which is a free and open source LMS, is widely used in many institution to manage and provide content to students and reference, we adopt this system as our source of content, we implement an extension to Moodle that can provide our system with SCORM lessons added to the local repository, Our system calls the extension in all participating Moodle systems then it collects the content and index it as will be described in section 3, figure 1 illustrates the structure described.

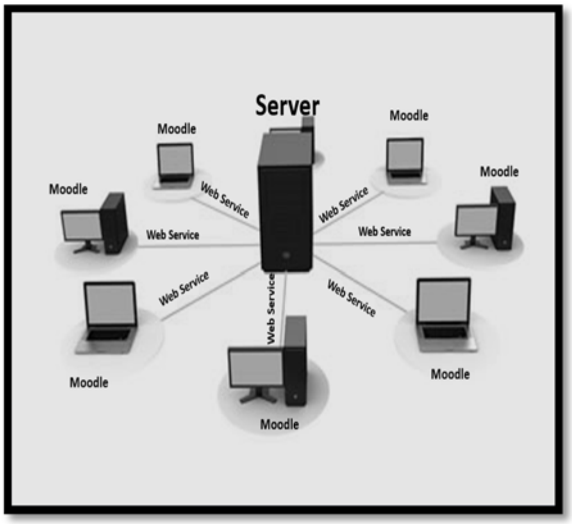

Figure 1: Proposed Architecture

This centralized search technique allows the learner to easily find the suitable content that matches his needs without the necessity to consult many sources.

3. INDEXING THE LESSONS

As the traditional IR systems, we have first to index the content, the model is similar to the standard inverted index that is used in traditional search engines.

As we are targeting SCORM standard file content; all files are stored in html format, thus, we have to index the content of all the html files inside a SCORM lesson, each html file in stripped and the text inside it is extracted, but as it is clear that html tags reflects some kind of semantics, there is a relation between the location of each word and the its importance inside the document. Our approach takes this fact into consideration, so the path of each word is considered to determine the weight of the word inside the document along with the frequency of its occurrence. For example, when the word appears in the title it is considered more important than when it appears in the body of the text.

Our approach supposes that each HTML tag is linked to a weight that it passes it to the children, some tags may increase the importance of its descendants like the <TITLE> tag, whereas other tags may reduce the importance of its descendants like the <P> tag. As tags can be nested the weight of the word is related to the weights of all the tags the wrap it starting from root tag down to the direct parent of the word.

To find out the weight of each tag, we used the human experience on a corpus, as follows:

We prepared a corpus of 200 documents that are in HTML format, these documents are reviewed by 12 persons that have experience in variety of fields, for each document a person is asked to rate 20 words in random locations, let us denote person number i as P_i, document number i as d_i and $w_{i,j}$ is the word number j in document number i. Now the location of each word $w_{i,j}$ that is chosen by an expert in some document is extracted, as we have html document, the location of a word is the XPath of it, that is the set of tags that contains the word directly and indirectly. Let us denote this list as $T_{i,j}$. The evaluation function of word $w_{i,j}$ for a person p_k is denoted as e(i,j,k).

We asked the evaluators to assign values to the function e so that the higher the value is the more the word is important in the document, but as people may consider a different scale we had to normalize the results, for each person p_k and document d_i we calculated the average value:

$$av_{i,k} = \text{average}(e_{i,j,k}) \text{ for each } w_j$$

Now the values of function e is divided by the average and a new normalized function ne is defined as follows:

$$ne_{i,j,k} = e_{i,j,k} \;/\; av_{i,k}$$

We have to reflect the evaluation of each word to the tags that are parts of its location, it is obvious that when a word is more important, the importance tags of its path should be increased and vice versa. To calculate the importance of each tag in a document from the viewpoint of a single evaluator we calculate the average of the evaluations of the words that this tag appears in their location path:

$$ET_{t,d,p} = \text{average}(ne_{d,w,p}) \text{ for each w where } t \in T_{d,w}$$

Where $ET_{t,d,p}$ is the evaluation of tag t in document d from the viewpoint of expert p. We can find the total evaluation of a tag by averaging the evaluation of the tag in all documents for all persons:

$$ET_t = \text{average}(ET_{t,d,p}) \text{ for all documents d and all experts p}$$

For each word w found in a document d we define the importance of this word in this document as the product of the evaluation of all tags in the location path of this word:

$$imp_{w,d} = \prod_{t \in T_w} ET_t$$

If the word appears more than once in a document then the total importance of each word in a document is the sum of the importance of all instances of this word in this document:

$$imp_{w,d} = \sum_{all\ instances\ i\ of\ w} imp_{i,d}$$

The document and words are stored in an inverted index along with the importance of each word in each document.

4. PROCESSING QUERIES

The aim of our system is to link each query with the best lesson, the nature of the queries in our system is linked to the educational terms, so the student may search for a scientific concept such as theories, historical persons, mathematical idioms, and research topics.

Each query consists of one or more words and the goal is to find the best lessons that matches semantically and by content the required query.

The traditional search engines look for documents that contains the words of the query then they may order them according to some criteria, some modern search engines may apply some advanced techniques such as replacing word with its synonyms. The target of our system is to find the content that is similar to the query regardless of the existence of the search query words inside the documents, so we choose to look for the set of related words to the query inside the indexed documents.

To find a set of related words for a word we use the dbpeida, it is an RDF graph that contains linked data which can be queried using SPARQL query language. The DBpedia graph is very intensive and contains almost concepts from all fields of science.

The starting point of our method is to find the concept of the query in DBpedia then we find the adjacent concepts that are linked to this concept. To demonstrate our method let's assume that the query contains a single word (or a single term) for now. The first step is to find the concept that represents this word in DBpedia, and this is done easily by the lookup service of DBpedia, which is a service that converts literal words to URI of concepts in DBpedia. If we cannot find the word in DBpedia we assume that the list of related words in empty and we keep the original word to look for it.

SPARQL is used to find the adjacent concepts of the concept representing the query single word, DBpedia is represented as tuples of data in the shape (subject, property, object), we should look for all tuples that the subject is our concept, and extract all objects from these tuples. Now we have a list of concepts that are adjacent to the original concept. The indexed documents contains just a set of textural strings, so we need to represent the concepts as text, we choose to extract the value of the label property (represented by the URI rdfs:label) for each concept of the extract list of adjacent concepts, the resulting textual strings are considered to be the words to be search for in the inverted index and let us denote them by $WADJ_w$ which is the set of related words that are extracted using the former method for the word w, the SPARQL that we executed to get the result is as follows:

```
select distinct ?name where}} {{term []
?Concept . ?Concept rdfs:label ?name}}
```

For example, the result of the above query for the word 'earthquake' contains the word 'Seismology' which is clear a great extension.

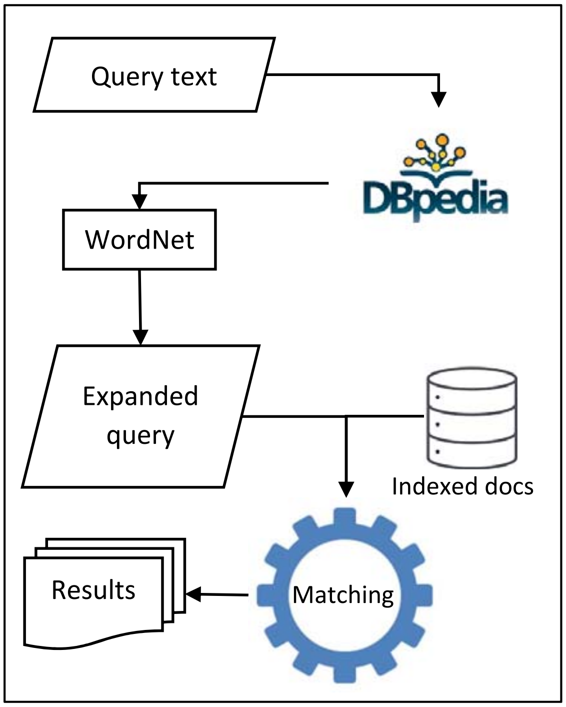

Figure 2: Query Processing Process

We can notice that using DBpedia will enrich the query of the user with many useful terms that can express the deep semantics of the intent of the user but by the other hand it may add terms that can generalize the query which may in turn add unwanted result, this problem is solved later.

If the query contains more than one word: q=w1 w2 … wn, we find out the set WADJ for each word separately, then the intersection of the sets is considered to represent the Query:

$$WADJ_q = \bigcup_{w \in q} WADJ_w$$

The idea behind taking the union of the sets is that the concepts that the user may be interested in should be all enriched by more query words, besides using this technique may increase the precision of the results as it may lead to more related documents.

There are a lot of related concepts (more than 100 on average) for a single concept, some of these concepts may be very generic and has no related semantics to the domain of the original query, so we should qualify each concept in this set so that specific concepts are considered more important than the general ones. Moreover, adding more terms may cause the system to overfit, that is, the recall will be very low and the system as a whole will be useless.

To solve this, we choose to benefit through WordNet. The WordNet thesaurus represent the relation of the words, it has two main relations: hyperneme and hyponeme, these two relations are opposite and represent the fact that the concept has a more general or specific conceptual meaning, see figure 2 for example. This means that WordNet is a directed tree that has a root and the more you are deep the more specific you are, the height of the concept is the number of relations that connect this concept to the root:

$$H(c) = H(hypr(c)) + 1$$

$$H(root) = 0$$

Where hyper(c) is the direct hyperneme of the concept.

It is clear that the height of the concept reflects the specificity of the concept, thus, this measure is suitable to be used to qualify the concepts that are extracted from DBpedia.

To find out the suitable documents, we firstly get the documents that contains any word in the set $WADJ_q$, then we assign each document a weight that is the sum of the weighted importance of words in $WADJ_q$, that is, for each word w we find the weighted importance of it in a document d as follows:

$$wimp(w) = imp_{w,d} * h(w)$$

The weight of the document is the sum of all the weighted sum of the words that exists both in this document and $WADJ_q$:

$$weight_d = \sum_{w \in d \cap WADJ_q} wimp(w)$$

The documents are sorted in descending order according to this weight and are finally viewed to the user, figure 2 shows the whole process.

5. IMPLEMENTATION AND RESULTS

To implement our proposal, firstly we developed a web server that is able to receive queries from learners and process it as mentioned above. The server indexes the documents in background as a continuous process, it collects learning materials from participating learning sources, which is Moodle systems that is equipped by a special plug-in, we have developed this plug-in to allow access to SCORM content stored inside the Moodle system, and send them back to our server to be indexed.

To evaluate the results of our proposal, we have to find out the precision and recall of our system:

We choose 60 queries based on recommendation from students in our university, this makes the queries arbitrary and reflect the actual needs of real learners.

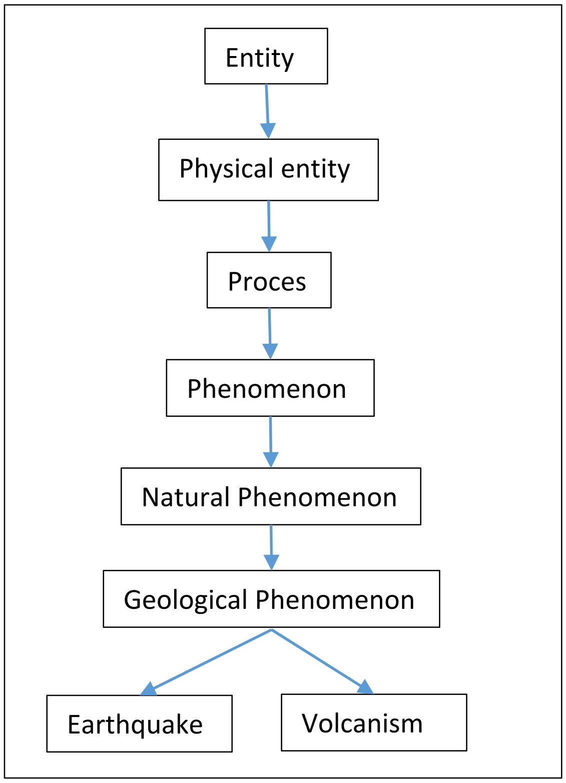

Figure 3: Example Of Wordnet Structure Showing The Word 'Earthquake' Hierarchy

We index 150 documents from different fields and fetched from various sources, then we applied the 60 quires to our system and get the result documents of each one respectively.

Let’s denote the query with order i as Q_i, then each document in the list of result documents of query Q_i is denoted as $D_{i,j}$ where i is the order of the query and j is the order of the document.

We choose 10 experts to evaluate our system for each expert E_i with order i we asked this expert to evaluate the likelihood of document d to be in the result of query q, that is:

$L_{i,d,q}$ is the evaluation of Expert i of document d to be in the results of query q

We have 90000 evaluations in total, and to calculate the precision of our system we have to find the average evaluation of each query related by each document by averaging the evaluations of all experts, that is:

$$E_{d,q} = \sum_{for\ all\ expert\ i} L_{i,d,q}$$

To make sure that the evaluations of each single expert is rational, we omit the out-layer values that has a significant difference to the average, and then we recalculate the average again.

For each query, we are concerned in the documents that has positive evaluation, that is: $E_{d,q} > 0$, the set of these documents is ordered according to the value of E for each of them which is denoted as OE, the result is the perfect expected result list of the query, i.e. OE_1 is the most related document to the query, and OE_n is the least related document (but still related).

To find the accuracy of our system we should compare the results of each query (denoted by D earlier) and the perfect results list OE.

Let us define the function PR that represent the perfectness of a result list according to a query as follows: if we have the list RL which is the result set of a query, recall that for each document in RL we have $E_{d,q}$ the pre-calculated evaluation of this document related to this query discussed earlier, and O_d is the order of document d in the set RL, then:

$$PR(RL) = \sum_{for\ each\ document\ i\ in\ RL} (n - O_i) * E_{i,q}$$

Where n is the number of document in OE

The prefect result is PR(OE) and we can find the accuracy of the actual result RL by the following equation:

$$NPR(RL) = PR(RL) / PR(OE)$$

We can prove mathematically that the value of NPR(RL) cannot exceed 1 and it is suitable to represent the accuracy of query q that generates the result RL.

To find the total accuracy of the system we average the accuracy of all experimental queries. The same process is conducted but on a traditional search engine using the same queries and documents, to find out that the accuracy is:

Table 1 shows the individual query accuracy for both systems, where the first column refers to our system, and the second one shows traditional search engine accuracy

Table 1: The Experimental Results

Query	Our model	Baseline
Solar System	0.514625717	0.091637867
Biology	0.424889945	0.241458569
Black Holes	0.533798823	0.490319237
Anatomy	0.433790216	0.572002065
Triangle Area	0.281019581	0.211680258
Programming Objects	0.236643242	0.180761456
Sport	0.916152898	0.90998767
Human Body	0.653792924	0.70413459
Firon	0.212950601	0
Egypt Firon	0.680167598	0
Smart Card	0.193981103	0
Chemistry	0.431558653	0.506821889
Protons	0.594597319	0.30732789
Hormones	0.610268657	0.555462687
Sexually Dimorphic	0.689989785	0.157303371
Quad Core	0.265829596	0
Acceleration	0.455060458	0.503036832
Physics Acceleration	0.768601698	0.512638622
Planets	0.275184186	0.25373842
Dynamics	0.344957996	0.143649834
Dynamic Physics	0.700070983	0.189372813
Newton Law	0.720888846	0.721081237
Blood Cell	0.675410983	0.561559986
Human Nerve	0.80453193	0.502059968
Law of Gravity	0.758881988	0
Michael Jordan	0.818181818	0.781818182
Vectors and Arrays	0.332405173	0.043066642
basketball	1.234939759	1.222891566
Human Enzymes	0.678266837	0.134692677
Enzymes	0.315253511	0.146961089
Thermal Equilibrium	0.302223987	0.10802224
CVD	0.840210356	0.115291262
Light Spectrum	0.336279275	0.132125497

Optical Prism	0.054031588	0
Programming Loop	0.293777374	0.27514255
Magnetic Field	0.468857847	0.266392769
Particles	0.350638913	0.313509604
Volcano	0.201135826	0.371036441
Endocrine	0.412844037	0.550323799
Average	**0.508120308**	**0.327623323**

The table above reflects that the use of proposed sematic approach has a significant effect on the results of the search engine and is step forward, we can notice that queries which express more general concepts has achieved better in proposed method, this is due to the fact that general terms has many related terms which is one of the benefits of our method, that is, matching more specific documents for general queries.

The results show a significant improvement in results when using enriched queries by semantically related terms, the improvement is about 60% which reflect a good pointer that this method may be promising if we can improve the semantic relations and get more accurate words.

Comparing the results with the literature, we can see that the most recent research [4] that uses a statistical model to expand the query, where they showed an improvement in their model over older ones, they could increase the accuracy by about 30% using a corpus built form Wikipedia and Markov model, whereas our semantic model showed a better improvement over baseline model, this is really expected as adding terms in statistical manner may not necessarily find suitable terms if a document from a new corpus which is going to be searched, whereas using a global knowledge base that contains data from almost all fields proves more accurate results.

By the other hand comparing our method with the similarity based retrieval models that uses a lexical ontologies like WordNet to find the similarity between terms [5] they stated that they could achieve a 30% better precision which mean that our method achieves a far better results, the usage of lexical database cannot find the semantically related words, it can find lexically similar words that may share some semantics with the original terms but will not get a full list of related concepts.

Our results show that expanding queries with weighted terms that are extracted form a global ontology is promising and can improve the information retrieval considerably. Especially when e-learning articles are to be retrieved, it is more efficient to enrich the query by terms from the domain rather than adding terms that statistically occur with the original terms or adding terms that lexically similar to the original term. Besides it is more feasible for e-learning purposes to index unstructured data and process human written queries that most previous research in e-learning IR does not provide.

Practically, we have implemented a search engine that uses the proposed method, the system indexed data from a single source, but as hundreds of educational institutes that provide e-learning contents uses Moodle, we can easily add many sources to our search engine, it is as easy as adding a plug-in to the Moodle system that we have already developed, this will turn our search engine into a global e-learning source that can be used by students all over the world to find suitable documents, it is clear that adding documents from many fields will not decrease the precision of our system as the proposed method depends on a global and overwhelming ontology "DBpedia" that covers almost all fields of science.

The application consists of a search engine with web based interface, the query is processed online, the SPARQL query that is needed in our method to find similar terms is executed using the online endpoint that DBpedia provides which is relatively fast and can provide results in real time, the results are parsed using a special parser then the matching algorithm is applied to find the suitable lessons that are already indexed, the indexer runs in the background, it collects content from specific sources.

Figure 4 demonstrate the process used in our real-life application, the main indexed content is SCORM lessons, that is a single compressed file that contains materials and meta-data, the learning materials are represented in HTML.

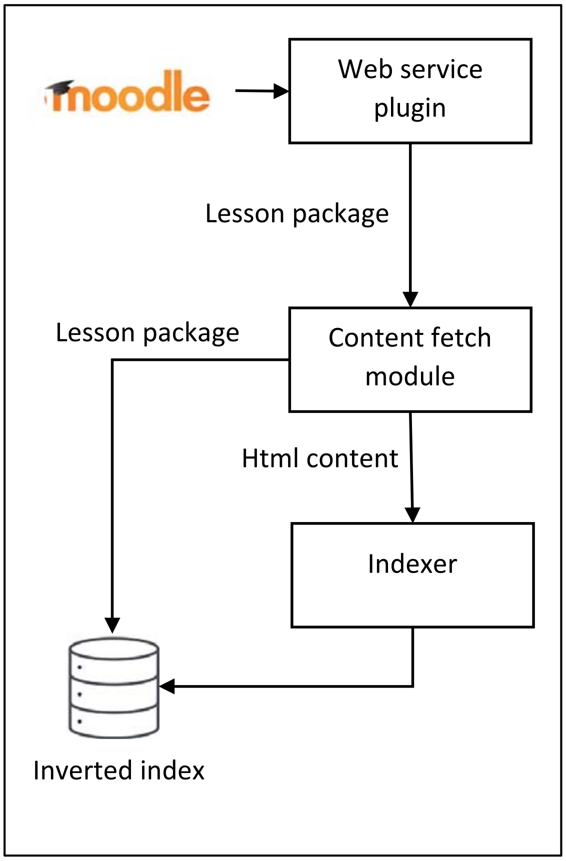

Figure 4: Structure Of Indexing Content In Our Application

The plug-in we developed for Moodle searches the system for all existing SCORM files and send list of them via Restful web service protocol to our server. Our server downloads contents from source system via another web service, then it extracts the content of the lesson. The content to be indexed is the HTML files inside the lesson, so the indexer looks for these files using the SCORM index, then for each HTML file our indexing algorithm is applied, the lesson itself is stored without modification but the inverted index of the lesson contains terms from all files inside the lesson, so when retrieving the content, the lesson is considered as a whole.

6. CONCLUSION

It is important to build a semantic search engine for e-learning content to help learners find most suitable documents that matches their queries. In this paper we showed a new method to build an e-learning search engine that uses the semantics of the content; it uses DBpedia to extract semantically related words to the query terms; the query is enriched by these words after adding weights to them by using WordNet. Our search engine plays the role of topic matcher that helps users find documents linked to specific topic, which is usually done using search engines. We showed our algorithm that contains new indexing and retrieval techniques in details and compared the results of this method with a traditional search engine by building a human based corpus and apply both engines to it, we proved that our system is more reliable than traditional search engines, comparing the results with literature our methods shows that it has promising results over statistical query expanding methods and lexical database query enhancement methods, the usage of semantic relations shows a good advancement in e-learning information retrieval.

As the usage of ontologies proves a good result, it is a good starting point to improve this kind of search methods, for future work it is good to conduct research about merging relations form more ontologies or find a better method to weigh the relations, many other ontologies may be tested as well, where weights can be calculated according to the kind of semantic relation, as the goal of any information retrieval system is to find the best matching document that the user is looking for, the search query in e-learning system reflects the intention of the user to look for a specific educational field, therefore, expanding queries using semantic relations is suitable here, but in some other systems where quires should be matched by documents lexically, this method will not be suitable, nevertheless, many other applications need to match query with content semantically where our proposed method may achieve good results and deserve to be tested, so as an additional future work, we are going to develop our method to be used in other fields that content may be compared semantically, such as social web contents, where text is related to topics and users may be interested in some topic but they should express query in natural language.

To conclude this paper we can observe that information retrieval system for social web content is an important topic to work on, the field of IR in general and in e-learning system in special has many researches in the last years, on the other hand ontologies and knowledge bases has proved to be a good tool in research, they are a very powerful tool to represent human knowledge as linked data, using these tools to enhance IR in the field of e-learning will have a good results, we showed that using DBpedia which is an ontology built using Wikipedia and WordNet have improved the accuracy of document retrieval in e-learning information retrieval system.

REFERENCES

[1] Friedman, E. *E-LEARNING PROS AND CONS*. 2014 [cited 2017; Available from: http://blog.eskill.com/e-learning-pros-cons/.

[2] *Pros and Cons of eLearning*. 2013 [cited 2017; Available from: https://www.digitalchalk.com/blog/pros-and-cons-of-elearning.

[3] Bohl, O., et al. *The sharable content object reference model (SCORM)-a critical review*. in *Computers in education, 2002. proceedings. international conference on*. 2002. IEEE.

[4] Gan, L. and H. Hong, *Improving query expansion for information retrieval using wikipedia*. International Journal of Database Theory and Application, 2015. **8**(3): p. 27-40.

[5] Hliaoutakis, A., et al., *Information retrieval by semantic similarity*. International journal on semantic Web and information systems (IJSWIS), 2006. **2**(3): p. 55-73.

[6] Fernández, M., et al., *Semantically enhanced information retrieval: An ontology-based approach*. Web semantics: Science, services and agents on the world wide web, 2011. **9**(4): p. 434-452.

[7] Rinaldi, A.M., *An ontology-driven approach for semantic information retrieval on the web*. ACM Transactions on Internet Technology (TOIT), 2009. **9**(3): p. 10.

[8] Silvello, G., et al., *Semantic representation and enrichment of information retrieval experimental data*. International Journal on Digital Libraries, 2017. **18**(2): p. 145-172.

[9] Cesarano, C., A. d'Acierno, and A. Picariello. *An intelligent search agent system for semantic information retrieval on the internet*. in *Proceedings of the 5th ACM international workshop on Web information and data management*. 2003. ACM.

[10] Bansal, M. and J. Arora, *A Review on Ontology Based Information Retrieval System*. International Journal of Engineering Development and Research, 2016. **4**(2): p. 263-265.

[11] Auer, S., et al., *Dbpedia: A nucleus for a web of open data*. The semantic web, 2007: p. 722-735.

[12] Miller, G.A., *WordNet: a lexical database for English*. Communications of the ACM, 1995. **38**(11): p. 39-41.

[13] Dougiamas, M. and P. Taylor, *Moodle: Using learning communities to create an open source course management system*. 2003.