Abstract—Learning Content Management System (LCMS) is a powerful tool for supporting distance learning. One of the LCMS development problems is subject searching. Most users have no idea about subject name of what they are looking for. They only know about subject contents. Nowadays, search engine embedded with LCMS gives result based on string matching keyword based. Precision and recall of this method is low. This research proposes subject name search based on document content using weighted ontology. Ontology is built from extracted term. Each term is given a weight based on the number of its relation. User query is expanded based on its synonym in WordNet. It is also weighted and taken into account of its similarity with course ontology. System retrieves similar or same subjects based on user query. Precision and recall of weighted ontology search result is better than string matching keyword search.

Keywords—Learning Content Management System, system information retrieval, weighted ontology.

I. INTRODUCTION

Some recent years, web based learning system is well known in higher education. Internet and web technology give solution for providing and sharing information in all fields. Web based learning system commonly used is Learning Content Management System (LCMS) [1]. Basically, LCMS is a high strategic solution for planning, sharing, and managing all learning process in an organization including online, virtual classroom, and distance learning [2]. It gives an opportunity for users to get a connection to the world outside the classroom and to make some new research topics. It is recommended to be implemented in higher education for improving the quality of management and education to become more modern, and as a leading education provider, including lifelong education [3].

LCMS quality is determined by learning objectives, lecturers, audiences, learning environments, and learning resources. One of learning material quality measurement factor is LCMS completeness [4]. In addition, completeness is one of LCMS maturity level determinants. Accessing LCMS should be effective and efficient [5] so that user satisfaction level will be high [6].

The major problem existed in current LCMS is the difficulty of resource searching [7]. This problem is existed because knowledge organization management is poor and there are some differential terms used. Appropriate search model for LCMS is based on its content. It is not only based on subject name. Several search engines are available.

Users enter keywords then system will return the results. All results contain same characters as the keywords [8]. Sometimes, the returned results are not relevant because they are only string matching [9]. System will never give result as documents contain similar meaning (synonym and homonym) with keyword. This problem is not only the weakness of string matching search but also metadata search [10].

Both string matching and metadata search in LCMS make users search subject name harder. Example: a user wants to join a subject about “mathematics”; LCMS will retrieve all subjects containing “mathematics” term. While subjects do not contain “mathematics” term but have similarity meaning with “mathematics” is not retrieved. This can be solved by semantic searching that can handle synonym and homonym [10].

Semantic search is not only keyword search. It also check terms context to provide more relevant results. Semantic search is a search of semantic network. It is not a network of several documents. It is a network of resource relationships that indicate the real object. It contains readable machine information. It is a collection of semantic network links to HTML documents. Some data from different resources can make semantic network data more complete [11].

Ontology is a semantically related concept [9] that can make information and semantic meaning representation [12]. In LCMS, ontology is used to support semantic search. Ontology is able to read query for learning objects and retrieve indirect readable learning object relations. This ability is very complex if
it is applied in simple keyword search or meta
data [13].

Moodle as one of the biggest LCMS has a
weakness in subject search and retrieval. Search
process is only based on user keywords (string
matching). Hence, information retrieval
precision and recall is not optimum. This
research aims to increase subject name
retrieval precision and recall of LCMS using
weighted ontology from text document
extraction.

This paper systematic consists of (1)
Introduction; (2) System Architecture; (3)
Result and Analysis; and (4) Conclusion.

II. SYSTEM ARCHITECTURE

The procedure used in this research is depicted in
Figure 1.

Fig. 1 System Architecture of MWOS

Error! Reference source not found. depicts
that there are two layers of subject name
searching. They are front end and back end.
Back end layer is learning object repository and
indexed ontology of extracted document. Front
d end layer gets user query. Query is processed
and matched with ontology repository by an
agent matcher. The processes of query are
query extracted, query expansion, and query
weighting. Agent matcher do inference to result
the similar subject based on user query.
Inference is based on similarity between query
and learning object. Information retrieval
application in this research is called MWOS
(Modular Object Oriented Dynamic Learning –
Weighted Ontology Search).

A. Text Document Preprocessing

This step consisted of tokenization, stop
words filtration, and stemming. This is a
standard process in natural language processing.
We assume that the text documents are in
English. They are divided into individual
lexical units. This process is called tokenization
[14]. Stop words are defined in
http://www.lextek.com/manuals/onix/stopword
s2.html. They are pronouns, common verbs,
common nouns, adjectives, and frilly words
[15]. They are removed from the documents
because they are irrelevant to become indexed
term. Stemming is a process to remove
morphological and inflectional endings from
English words [16].

B. Ontology Building

Root of this ontology structure is general
LCMS. Subject is like Software Engineering,
Game Development, Software Evolution, and
Artificial Intelligence. Document contains
learning material uploaded by teachers. A
subject has several documents. Term is
extracted from text document. Ontology
building from unstructured text is based on [17].

C. Ontology Weighting

Each document extracted term is weighted
based on the number of its relation with other
terms. Weight is not based on term frequency
because the most frequent term does not reflect
the content of document. The One of methods
of giving a weight of a term in ontology is
based on term density [24]. The more a term
has relations with others the greater weight it
has. Formula of weight is based on equation 1.

\[ w(c) = \frac{\text{in-degree}(c) \times \text{out-degree}(p)}{\text{in-degree}(c) + \text{out-degree}(c)} \]  

\[ w(c) \] is a weight of concept c. in-degree (c)
is the in-degree of the concept c, while out-
degree (p) is the concept p; in-degree (O)
denotes the in-degree of the whole layered
structure, while out-degree (O) denotes the out-
degree of the hierarchical structure.

D. Query Weighting Process

Preprocessing text applied in learning object
documents is also done in query before
weighting process. They are tokenization, stop
words filtration, and stemming process. Then,
query is weighted. It is weighted based on user
query order. Example, a user gives keyword
"game development using fuzzy". Queries used
for search are “game”, “develop”, and “fuzzy”.
The weight for each query is taken a count as described below.

\[ \text{Game} = 3 \]
\[ \text{Develop} = 2 \]
\[ \text{Fuzzy} = 1 \]

Game is given weight 3 because it takes the first entrance of three queries. It is entered before develop and fuzzy. Develop is given weight 2 because it takes the second position.

The next calculation is:

\[ \text{Game} = \frac{3}{(3+2+1)} = \frac{3}{6} = 0.5 \]
\[ \text{Develop} = \frac{2}{(3+2+1)} = \frac{2}{6} = 0.33 \]
\[ \text{Fuzzy} = \frac{1}{(3+2+1)} = \frac{1}{6} = 0.17 \]

E. Query and Ontology Matcher

Matching operation shows an alignment \( A' \) for a pair of ontology \( o \) and \( o' \). There are several parameters to expand matching process definitions. They are \( A \) input alignment, matching parameter (e.g. weight and threshold), external resources used for matching process (word similarity or thesauri). Matching process is assumed as \( f \) function. From a pair of ontology \( o \) and \( o' \) which will be matched, an alignment \( A \), a set of parameter \( p \), and resource \( r \) produce an alignment \( A' \). \( A' = f(o, o', A, p, r) \).

It is depicted in Fig. 3.

Matching process between two ontology or more is called multiple matching. Multiple matching is assumed as \( f \) function that match ontology \( \{o_1, o_2,..., o_n\} \). An input alignment \( A \), a set of parameter \( p \), and a resource \( r \) produce an alignment \( A' \) among those ontologies [18]. \( A' = f(\{o_1, ..., o_n\}, A, p, r) \).

After user enters query, matching process will be applied. Query will be compared with document ontology. Query and document ontology similarity is based on ontology and WordNet hierarchical structure. Similarity between two words is taken account by Wu & Palmer formula (2).

\[ \text{Sim}(c_1,c_2) = \frac{2 \times N_3}{N_1 + N_2 + 2 \times N_3} \]  

\[ (2) \]

F. Moodle

This weighted ontology will be implemented in Moodle (Modular Object Oriented Dynamic Learning Environment). It is a web based learning and uses social pedagogical principles. It is an open source software and under GNU license. Almost 50 thousand sites in the world are used Moodle. Features provided in Moodle are site, course, and user management. Course management is consisted of assignment, chat, forum, glossary, lesson, quiz, learning material, survey, wiki, and workshop. User management is enrolment and user roles.

Moodle still has some weaknesses. One of them is search process. System will search user keywords into subject description. The example is that user gives query “natural science”. The result is all courses contained terms “natural science” in their description. Subjects that have no terms “natural science” in the description is not a result, even though they have “natural science” in their documents. To optimize the result, MWOS uses WordNet and weighted ontology. Since MWOS aims for subject retrieval, only table related with course that is used.

III. EXPERIMENT RESULT

For this research, there are 5 subjects. They are WebDevelopment, Software Evolution, Artificial Intelligence, Computer Network, and Game Engineering. Each subject has 3 documents doc, pdf, and txt. Experiment is to prove that weighted ontology search is better than ontology and keyword based search in precision and recall. Formulas to count precision and recall are:

\[ \text{Precision} = \frac{\text{Relevant retrieved}}{\text{Retrieved}} \]
\[ \text{Recall} = \frac{\text{Relevant retrieved}}{\text{Relevant}} \]  

\[ (3) \]
\[ (4) \]
\[ F \text{ Measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \]

**TABLE I**

<table>
<thead>
<tr>
<th>Keyword</th>
<th>E-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Moodle</td>
</tr>
<tr>
<td>Software Engineering</td>
<td>0</td>
</tr>
<tr>
<td>Game Development</td>
<td>0</td>
</tr>
<tr>
<td>Fuzzy Method</td>
<td>0</td>
</tr>
<tr>
<td>Bayesian Network</td>
<td>0</td>
</tr>
<tr>
<td>Evolution of software process</td>
<td>0</td>
</tr>
<tr>
<td>Local Area Network</td>
<td>0</td>
</tr>
<tr>
<td>Maturity level of a software</td>
<td>0</td>
</tr>
<tr>
<td>Game design architecture</td>
<td>0</td>
</tr>
<tr>
<td>Dynamic user requirement</td>
<td>0</td>
</tr>
<tr>
<td>Web development</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1 shows that weighted ontology (MWOS) can retrieve subject name based on its text document uploaded. Moodle only retrieves subject name based on its subject. In MWOS, users can enter anything query in their mind to retrieve subject name. MWOS also retrieves subject based on similarity of keyword given user. The similarity is based on WordNet. F Measure of MWOS’ subject retrieval is better than Moodle’s.

IV. CONCLUSIONS

MWOS can increase F-Measure of subject retrieval. F-Measure increase 0.81. Users do not have to know subject name when they input query. They are just asked to enter query they want to study. Weighted ontology can be applied for other search process. It is not only applicable in LCMS. The weakness of this research is the matching process is still meta data matching. For the next research, matching process for a pair of RDF can be developed.

REFERENCES