Ontology-based Annotation Recommender for Learning Material Using Contextual Analysis

Diana Purwitasari,
Esti Yuniar,
Umi L. Yuhana,
Daniel O. Siahaan
Informatics Department, Information Technology Faculty,
Institut Teknologi Sepuluh Nopember
Jl. Raya ITS - Gedung Teknik Informatika ITS Surabaya 60111, +62 (031)5939214
INDONESIA
diana@if.its.ac.id

ABSTRACT

Many learning materials displayed in web media are generally classified into certain categories. Discussed topics in categories could become annotations as metadata of the article. Annotation for a learner could be used to determine which materials should be read next to support self-study in student-centered learning model. Despite abundance materials that exist nowadays most provided annotations are done manually by user.

In this paper we present a recommender system that can automatically provide annotations to help user. The system could identify the topics discussed within article which is worked out by semantic approaches with Latent Semantic Analysis (LSA) and WordNet. Recommended annotations are obtained by determining proximity between contexts of categories and learning materials. Then recommended categories are structured into ontology model in order to share common understanding of annotation so we can publish the annotation later to different learning sites.

Our experiment used learning materials within domain Software Engineering. We utilized widely known categories from Software Engineering Body of Knowledge (SWEBOK) as our ontology-based annotations. Recommender system will found the appropriate annotations using WordNet approach when there is no result of recommended categories using LSA approach.

Keywords: *Automatic annotation, Semantic analysis, SWEBOK, Ontology*

INTRODUCTION

Annotation could be defined as a summary over certain information underlining the subject of content. Annotation in learning material has function to remark discussed topics within an article for assisting students in their learning (Marshall, 1998). Sometimes a material not only covers topics in one category but it also extends to cover some other categories. Annotation for a learner is used to determine which materials should be read next. With annotation students are encouraged to do self-study as expected in a learning model called student-centered learning.

Since large amounts of learning materials in the recent years are now available as electronic documents hosted upon web servers, developing annotation tools on the web has received attention from academia and industry. One of annotation tools called Web annotation tools (WATs) allows both individual and community to make and share the annotations (Raua et al, 2004). Nevertheless manually annotating learning material with such amounts is a time consuming and expensive process not to mention it is liable to human errors. Thus automatically annotating to all kind of data which consists of text added for the purpose of explanation or description with techniques like natural language processing is gaining a lot of interest (Devshri et al, 2008; Mallik et al, 2008). Search engines, browsers, and similar Web services or applications can take advantage of annotation once semantic descriptions of annotation are available to existing or newly created e-learning material (Lu et al, 2002). In doing so, it is advisable that annotation is based on consensual knowledge like widely known ontology model in biomedical domain called MEDLINE (Camous et al, 2007).

In this paper we presented annotation recommender system on e-learning materials using semantic approach. Figure 1 shows contextual diagram for the presented system. The system works over topics covered in the contextual contents of articles to give annotations. The system semantically analyzes implicit concepts in the documents using Latent Semantic Analysis (LSA) with the help of WordNet. LSA needs to know word occurrences of documents therefore recommender system will do text processing on learning materials with the help of Oracle Text. The recommended annotations are structured in ontology model in order to share common understanding so we can publish the same underlying ontology-based annotations later to different learning sites. We used knowledge taxonomy in SWEBOK Guide (www.swebok.org) as comprehensive set of guidelines for software engineering discipline.

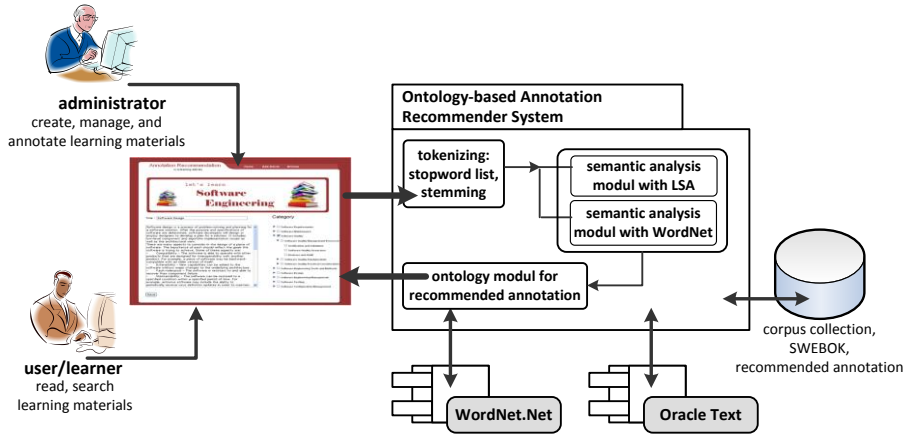

Figure 1. Ontology-based annotation recommender contextual diagram.

ANNOTATING LEARNING MATERIALS

Our proposed annotation is defined as semantic-based annotation which is obtained from analysis on contents of article using semantic approaches. The approaches used to produce recommended annotations are state-of-the-art methods of Latent Semantic Analysis (LSA) (Landauer et al, 1998) and WordNet (Miller, 1995). If there is no result of recommended category using LSA, the system will find the appropriate annotations using WordNet. Explanation of each method is described below.

Semantic Analysis using LSA

In this paper LSA becomes main method for getting recommended annotations. LSA determines proximity of semantic similarity between contexts words in a category and a learning material based on their contextual usage statistics. Some brief explanations will be provided here but more detailed accounts of the methodology of LSA are beyond the scope and could be found in the originator's manuscript (Landauer et al, 1998). LSA is a corpus-based computational method so the first step is to derive a document collection of learning materials into a term-document matrix where each term *t* in a document *d* is represented as a row in the (*t* by *d*) matrix. Then each column represents a document *d* which contains count of the number of times term *t* occurred in the document.

A stop list is used to eliminate commonly occurring words like particles or prepositions because term-document matrix could spread until thousands of unique terms. The system applies a stemmer for finding terms with the same root form to reduce the number of unique terms. Since term-document representation matrix still could have large dimension, the matrix size is reduced by eliminating terms whose weight value is under a threshold number. TF-IDF as term weighting scheme uses the product of term frequency and inverse document frequency to increase the significance of terms (Salton & Buckley, 1988). The weighting scheme implies that the best terms should have high term frequencies (TF) in individual documents. But if the high frequency terms are prevalent in the collection, inverse document frequency (IDF) factor favors terms that concentrated in a few documents of overall collection. IDF factor of term *t* is computed as *log N/n*. *N* shows number of documents in the collection and *n* shows number of documents in which a term *t* exists. Term weights are normalized to the length of document vectors. Normalized term weight has continuous value between 0 and 1. Term weight with higher value indicates that the term is important.

Collection of learning materials derives a large and sparse term-document matrix. Large aspect is caused by number of unique terms found in the collection which defines matrix dimension, while sparse aspect caused by term occurrences that vary in each document. As mentioned before the dimension of matrix could have high value in proportion to the number of unique terms. To reduce matrix dimension becomes a problem of linear algebra and will be solved with a linear algebra technique called Singular Value Decomposition (SVD).

SVD technique generates a reduced-dimensional matrix representation but still preserves the variance in the original data. Reduced-dimensional matrix called latent semantic space maps term-document representation into a concept space (Landauer et al, 1998). Since concept space is actually matrix, concept closeness between categories and documents can be calculated using elementary linear algebra technique as well like cosine similarity.

Annotation recommender could recognize similar concepts between a category and a document. Recognizing process is equivalent to searching process with terms in a category become query keywords. Categories that indicate the same concepts to a document become recommended annotations because some topics discussed inside the document are representing the same semantic ideas with the categories. However searching process might retrieve zero results. Therefore contextual analysis of the recommender could fail in suggesting annotation for learning materials.

Semantic Analysis using WordNet

LSA could only map equivalent concepts from documents to categories if only unique terms within categories occur in the documents. If the terms do not exist in the prevalent collection then there is no result of recommended categories using LSA. Annotation recommender system should find the most appropriate categories using different terms within document excerpts that have similar concepts. If LSA fails then the next task is choosing different terms that carry same meaning with the corresponding terms in categories. Some works have been done regarding to a problem of measuring similarities among different sentences (Wu & Palmer, 1994; Yang & Powers, 2005; Dao & Simpson, 2008). The similarity measurement of different sentences is accomplished with a lexical database for English language called WordNet which consists of noun, verb, adjective, and adverb (Miller, 1995). Words listed in WordNet are organized into a number of synsets or synonym sets. The synset contains a word and its lexical concepts connected by various semantic relations such as synonym (has similar meaning) or antonym (has opposite meaning) or hyponym (has subordinate meaning) or others.

In order to measure semantic similarity between two sentences, like a title of learning material s_i and a category c, each sentence should be tokenized into unique terms. Steps like using stop word list and stemming are applied to reduce the number of unique terms. The system will calculate semantic similarity for each combination of terms between two sentences (Eq. (1)). Here a term is going to be replaced with a synset in WordNet. An easy way to measure the semantic similarity between two synsets is to treat taxonomy of terms in WordNet as an undirected graph and calculate the distance of two synsets in taxonomy (Dao & Simpson, 2008). Two synsets or terms with the same idea will have closer graph path. Semantic similarity for a learning material's title and a category $sim(s_i,c)$ is an average value of distance between each combination of their unique terms (Eq.(1)).

$$sim(s_i, c) = avg(dist(t_x, t_y)_{t_x \in s_i, t_y \in c}) \quad \cdots\cdots (1)$$

Figure 2 illustrates distance calculation between two terms, *dist(t_x, t_y)*, based on their semantic meaning with a taxonomy sample. Here the words of …

- 'activity' has 3 hyponyms ('operation', 'creation', 'variation'),
- 'creation' has 1 hyponym ('design'), and
- 'design' has 2 hyponyms ('configuration', 'planning')

… such that the specific meaning of 'activity' depending on contextual sense could be 'operation' or 'creation' or 'variation'. The opposite term of hyponym is hypernym.

As an example, semantic distance between terms of 'operation' and 'configuration' will be 5 because the path of 'operation' and 'configuration' has 5 nodes ('operation' - 'activity' - 'creation'- 'design' - 'configuration'). Two terms with shorter path means that those terms have more similar meaning. Words of 'configuration' and 'planning' have more similar meaning than 'configuration' and 'variation'.

Calculating number of nodes in a shortest path between two different terms is the easiest way to determine similarity distance for those two terms. We used WordNet.NET library which is a .NET library to access the WordNet database for calculating distances between two terms *dist(t_x, t_y)* (Simpson & Malcolm, 2005).

PREPROCESSING LEARNING MATERIALS

LSA derives its conceptual similarity measurement from statistical values of term occurrences. Therefore learning materials collection should be text processed to dissect the terms within. It is necessary to know words that appear in article and their frequency values in order to obtain important words that represent an article.

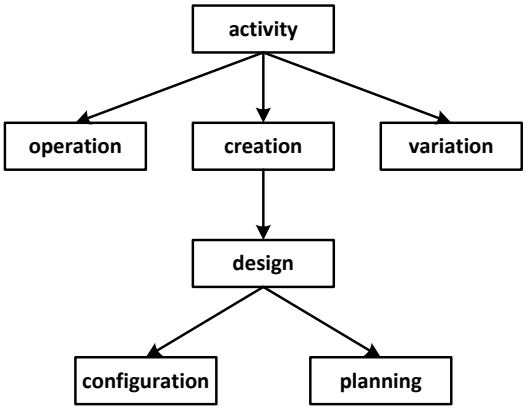

Figure 2. A taxonomy example for words in Software Engineering domain.

We used features from Oracle Text to get that information (download.oracle.com). Oracle Text is a technology that provides indexing, word and theme searching, as well as ability to see the text (Shea et al, 2008). Oracle Text supports several indexing types. We used English lexer to index learning materials because our collection and WordNet are English based-text. Next steps are removing terms on stop word list and stemming to reduce the number of indexed terms.

Terms should be weighted in order to determine which words are really important in an article from indexing process. Annotation recommender system uses TF-IDF (Eq.(2)) as weighting scheme (Salton & Buckley, 1988). Weighting is statistical measurement used to evaluate word importance to document in the collection or corpus. The importance level increases proportionally to the number of words appearing in document but it is balanced by words frequency in the corpus. Weighting process reduces number of unique terms.

$$tf.idf(d_i, t_j) = tf_{i,j} \times idf_j \quad \cdots\cdots (2)$$

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,k}};\ idf_j = log\frac{|D|}{1 + |\{d: t_j \in d\}|}$$

Computation for the first part of TF-IDF, $tf_{i,j}$, consists of the nominator $n_{i,j}$ that shows occurrence number of term t_j in document d_i while the denominator is the sum of occurrence number of all terms in document d_i. Then for the second part, idf_j, calculates *log* value of total number of documents in the collection $|D|$ and number of documents where term t_j appears, $|\{d: t_j \in d\}|$.

ANNOTATION SYSTEM USING SEMANTIC APPROACH

Annotation recommender system has goal to generate annotations that have common understanding of both users and web agents. Therefore recommended annotations can be published later to different learning sites. A common understanding model is necessary to have for those sites. Ontology has been used as a formal vocabulary of semantic web that allows web agents understand web contents. Ontology consists of taxonomy and a set of inference rules describing semantic relationship between concepts. Our implementation gives recommended annotations which are structured in common ontology model. Therefore we have provided input of e-learning materials and ontology from certain domain knowledge. We have experimented using Software Engineering Book of Knowledge called SWEBOK (www.swebok.org). Figure 3 shows some of SWEBOK taxonomy which is acknowledged as taxonomy in Software Engineering domain knowledge.

The implemented system asks user to select a category as a starting base before finding recommended annotation is done. The system will consider other categories in taxonomy-based SWEBOK that related to base category like its parent, siblings, or children for recommending process. Figure 4 shows ontology model for our system to accommodate recommended annotation.

The main classes are ***Category*** and ***Article***. Related categories from base category can be obtained from relationships of *isSubCategoryOf* and *hasSubCategory* owned by Category class.

As an example 'Test Process' in Figure 3 becomes base category. It demonstrates that …

- 'Test Process' *isSubCategoryOf* 'Software Testing' to show its parent,
- 'Test Process' *hasSubcategory* 'Practical Considerations' and 'Test Activities' to show its children.

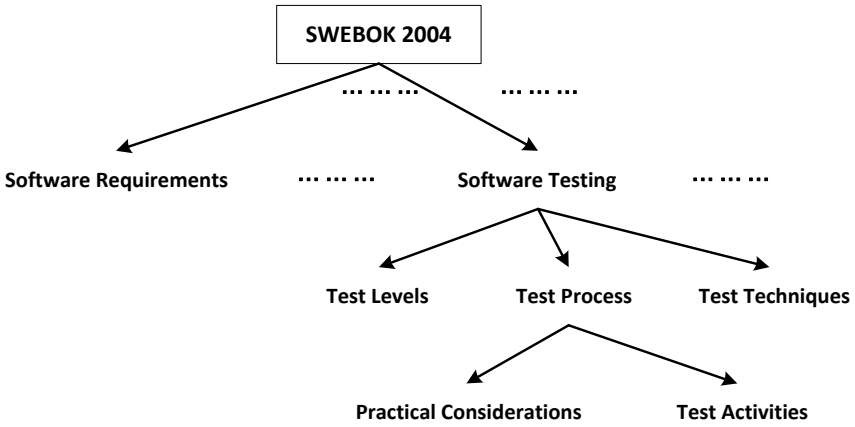

Figure 3. Sample taxonomy of categories from SWEBOK (www.swebok.org).

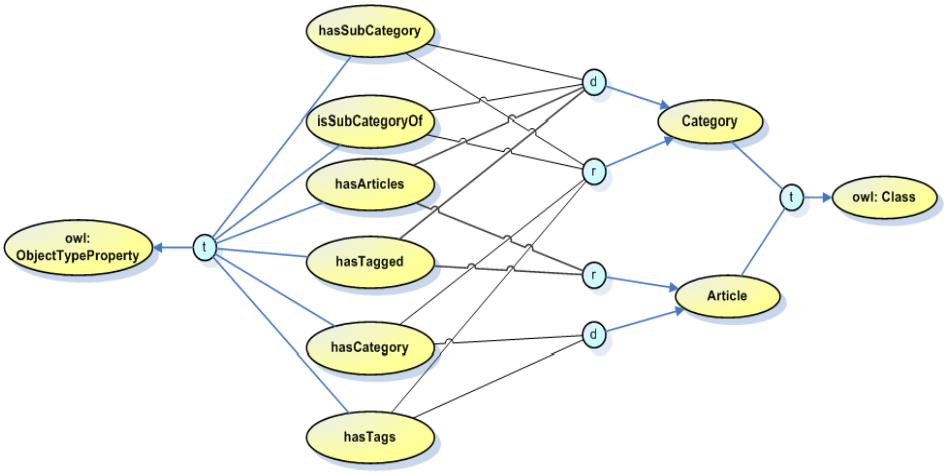

Figure 4. Ontology model for proposed annotation.

Types of relation in our ontology model are parent, siblings, or children. However our model has drawback with direct usage of relationships since it cannot explain sibling. The system will have to look for the parent of focused category first and then find other sub categories with the same level of focused category to recognize its sibling.

Using base category of 'Test Process' from Figure 3, the system will find similarity concepts of the learning material with ...
- its parent ('Software Testing').
- its siblings ('Test Levels', 'Test Techniques'), and
- its children ('Practical Considerations', 'Test Activities')

Figure 5 shows designed interface for implementation of recommender system. Here is an example of user asking annotations for a learning article entitled with 'Software Design'. User selected 'Software Quality' as base category to find recommended annotations. List of categories in the left side of Figure 5 is written based on SWEBOK taxonomy.

Figure 6 shows the results of recommended annotations. The system only lists some categories to become the recommended results. The categories should have concept similarity value ranked as the highest 30% within range value of 0 to 1. Next the user chooses any category from the recommended results. User in Figure 6 selected three categories. Then system will annotate the article with user selected categories.

Figure 5. Interface for managing article and asking annotation.

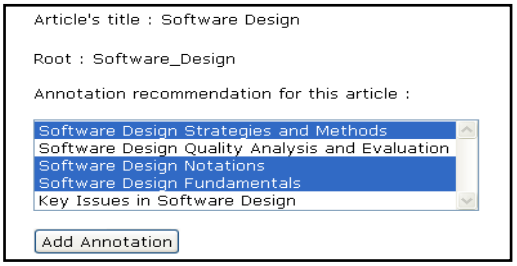

Figure 6. Confirmation interface for deciding recommended annotation.

SYSTEM EVALUATION

We have used some libraries for our system implementation such as Oracle Text (download.oracle.com) and WordNet.Net (Simpson & Malcolm, 2005). The first library, Oracle Text, is applied on tokenizing process while the latter, WordNet.Net, is used on analyzing process. Then we conducted testing to see performance of annotation recommender system. We prepared sample collection of 30 learning materials about Software Engineering downloaded from Wikipedia articles (www.wikipedia.org, February - July 2010). Dataset for experiment contained short articles with average length about 130 words after preprocessing steps. We used short articles because it is easier to confirm the correctness of recommended annotation and article's content. The preprocessing result made an indexed list contains ±20 unique terms in an article. By considering threshold value resulted from weighting scheme TD-IDF, the number of unique terms in an article is reduced. Threshold value is set to the highest 30% of term weight value in an article. Thus LSA transforms term-document matrix into concept space with less than half of article's content length.

Our experiment has a goal to evaluate precision of recommended annotations. Precision can be evaluated if there is an answerset of the correct categories for annotating a learning material. We defined the answer set of 30 documents. That was another reason why we selected short articles because it is easier to read, analyze, and annotate them manually. Precision represents the ratio of the number of appropriate categories from recommended annotations to the number of categories from answerset. Precision value is calculated by comparing number of recommendations matched to the number of overall recommendations. Average precision value for LSA, WordNet and combination of both is 71.4%, 46.3%, and 78.6% respectively.

LSA evaluation examined indexed terms of an article and terms of a category as search keywords to find similarity distance between concepts of an article and a category. WordNet-based evaluation only looked at terms in article's title and category to find its similarity distance. Therefore recommended categories as annotation resulted from WordNet have loose relation with article's content. If LSA method could not give any recommendation then WordNet-based method

could become alternative though its recommendation might not have exactly related concepts.

CONCLUSION

Annotation could be used to determine which materials should be read. A learner is encouraged to do self-study for succeeding student-centered learning model using annotations. Our implementation system generates recommended categories as annotation so user could select the most appropriate annotations. The system semantically examines concepts in learning materials and categories with combination method of LSA and WordNet. Ontology model for categories provided by SWEBOK taxonomy could be used to extract other categories with similar concepts. WordNet is responsible for the extraction task. The recommender system has a shortcoming caused by terms in categories. If the category terms might not exist within indexed terms from document collection then the system looks for other terms with nearest concepts by using WordNet. Since testing scenario in this current work is only applied with short articles, our next work is about doing thorough analysis with more learning materials.

ACKNOWLEDGMENT

The research work in this paper is supported by Institute of Research and Public Services, Institut Teknologi Sepuluh Nopember Surabaya, Indonesia, under the Grant No. 0535/I2.7/PM/2010.

REFERENCES

Camous, F., Blott, S., & Smeaton, A.F. (2007). Ontology-based MEDLINE Document Classification. In Proc. of 1st Intl. Conf. Bioinformatics Research and Development. Berlin, Heidelberg, 439–452.

Dao, T. N., & Simpson, T. (2008). Measuring Similarity Between Sentences. Retrieved April 10, 2010, from http://opensvn.csie.org/WordNetDotNet/trunk/Projects/Thanh/Paper/WordNetDotNet_Semantic_Similarity.pdf

Devshri, R., Sudeshna, S., & Sujoy, G. (2008). Automatic Extraction of Pedagogic Metadata from Learning Content. *International Journal of Artificial Intelligence in Education*, 18(2), 97-118.

Landauer, T.K., Foltz P.W., & Laham D. (1998). An Introduction to Latent Semantic Analysis. *Discourse Process*, 25(2-3), 259–84.

Lu, S., Dong, M., & Fotouhi, F. (2002). The Semantic Web: Opportunities and Challenges for Next-Generation Web Applications. *Information Research*, 7(4).

Mallik, A., Pasumarthi, P., & Chaudhury, S. (2008). Multimedia Ontology Learning for Automatic Annotation and Video Browsing. In Proc. of 1st ACM Conference on Multimedia Information Retrieval. Vancouver, Canada, 387–394.

Marshall, C. C. (1998). Toward an Ecology of Hypertext Annotation. In Proc. of 9th ACM Conference on Hypertext and Hypermedia. Pittsburgh, USA, 40–49.

Miller, G.A. (1995). WordNet: A Lexical Database for English. *Communications of the ACM 38*, 39-41.

Raua, P.P., Chen, S.H., & Chin, Y.T. (2004). Developing Web Annotation Tools for Learners and Instructors. *Interacting with Computers*, 16(2), 163-181.

Salton, G., & Buckley, C. (1988). Term-Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, 24(5), 513-523.

Shea, C., Faisal, M., Ford, R., Lin, W., & Matsuda, Y. (2008). Oracle Text Application Developer's Guide, 11g Release 1 (11.1). Retrieved April 10, 2010, from http://download.oracle.com/docs/cd/B28359_01/text.111/b28303.pdf

Simpson, T., & Malcolm, C. (2005). WordNet.Net. Retrieved May 10, 2010, from http://opensource.ebswift.com/WordNet.Net

Wu, Z., S., & Palmer, M. (1994). Verbs Semantics and Lexical Selection. In Proc. of 32nd Annual Meeting on Assoc. for Computational Linguistics. NJ, USA, 133–138.

Yang, D., & Powers, D.M.W. (2005). Measuring Semantic Similarity in the Taxonomy of WordNet. In Proc. of 28th Australasian Conf. on Computer Science. Newcastle, Australia, 315–322.

Copyright statement