World Applied Sciences Journal 22 (6): 851-855, 2013
ISSN 1818-4952

DOI: 10.5829/idosi.wasj.2013.22.06.27655

Automated Semantic Checking Using Stemming Approach

Noraida Haji Ali, Noor Syakirah Ibrahim and Noor Maizura Mohamad Noor

Department of Computer Science, Universiti Malaysia Terengganu, Terengganu, Malaysia

Abstract: The importance of modeling is apparent with its use to predict cost and time requirement especially in completing a system. However, there is no equivalent standard for evaluating the quality of conceptual models. Consequently**,** students often have difficulty in designing UML class diagrams during the test. They tend to design UML class diagram regardless of the quality and accuracy of the model as design consistency. To overcome this problem, the semantics must be added by the lecturers when marking exam papers. This paper proposes automated checking semantic of object-oriented model using synsets extraction. Synsets extraction process includes synsets extraction using Rita.WordNet from WordNet database, stemming process and final synsets extracted. The final results consist of synonym sets (synsets) that have been stemmed. Synsets generated from WordNet will be analyzed to solve some problems in modeling, such as correctness and completeness. Hopefully, the use of synsets from WordNet will help enhancing the quality of modeling particularly in examining UML class diagram automatically.

Key words: UML class diagram · WordNet · Semantic · Synsets

INTRODUCTION

Background: Modeling is important to predict the cost and time required to complete the system. In essence, the quality of information system depends on the quality of conceptual models. This will affect the efficiency and effectiveness in the development of information systems. In the modeling, analysis of semantic quality can assist students in designing Unified Modeling Language (UML) class diagrams. By analyzing the semantics, we dig deeper to examine whether they form a set of relevant instruction in programming languages. This will help students to overcome the diversity of notation for naming elements because it is semantically valid. UML is used to meet visualization, development and documentation of the results for the processes in a system [1]. Designing the UML class diagram is an important phase in modeling. To examine problems in the modeling, the study from Thomasson shows the difficulties in designing the appropriate UML class diagram [2]. They are:

- The variation of the design form.
- Naming the notation element.
- Free in designing.
- Difficult to state the class or object.
- Difficult to elaborate the requirement.

Unhelkar (2003) lists the several levels of quality in practical UML-based projects. They are data quality, code quality, model quality, architecture quality, process quality, management quality and quality environment [3]. UML class diagrams designed by students often ignore the quality of the model such as the correctness and completeness [ref]. This should be addressed to ensure that no duplicate in class naming and the inheritance relationship is valid. For this purpose, WordNet is used as a tool to resolve cases of in consistency in UML class diagrams [4]. Generally, if A is synonym to B and B is synonym to C, we can conclude that A is also synonym to C [5]. This hypothesis will be adapted in matching the class name given by students and the answers scheme. In UML class diagram development, student will be asking to create a class name based on situation or case study. Answer scheme will be used to check whether the answers given by students were right or wrong. If the student did not answer according to the answers scheme, WordNet is used to check the synonyms of the class name given by the students. If the synonym sets from

Corresponding Author: Noraida Haji Ali, Computer Science Department, Universiti Malaysia Terengganu, Terengganu, Malaysia. Tel: +60-0199311230.

[1]http://wordnet.princeton.edu/man/wninput.5WN.html.

WordNet match the answers scheme, the answers given by students are accepted to be true.

Scope: This study focuses on solving problem related to automated semantic checking of UML class diagram using synsets extraction from WordNet.

The rest of the paper is organized as follows. The next section introduces the WordNet and its structure. Section 3 discusses synsets extraction process in detail which consist four main steps. Section 4 presents the evaluation results. Finally, we provide the conclusions and discuss future works in Section 5.

Wordnet

Introduction: WordNet has been developed as a useful tool which combines thesaurus and dictionary [6]. It is a large lexical database for various languages. Nouns, verbs, adjectives and adverbs are put together into sets of cognitive synonyms (synsets), each expressing distinct concept. WordNet organizes the lexical information in terms of word meanings and could be perceived as a lexicon based on psycholinguistic principles. WordNet System is developed on a network of Sun-4 workstations [7]. The software programs and tools are written using C programming language, UNIX utilities and shell scripts. WordNet systems divided into 4 parts. They are:

- WordNet lexicographers' source files.
- The software (Grinder) to convert these files into WordNet lexical database.
- WordNet lexical database.

WordNet Structure: One of the important parts in WordNet is lexicographer files that contain the synsets. To extract the synsets from WordNet, the lexicographer files should be explored first [8]. The lexicographer files in WordNet will be processed by grind[1] which then will generate a database that suitable to be used with WordNet library, interface, code and other applications. In WordNet search results, these lexicographer files will appear when some options are selected for extended view. The format of lexicographer files is described in wninput[1].

```
Sense n
[{synset_offset}]
[<lex_filename>]word1[#sense_number][, word2…]
```

Fig. 1: Format in WordNet Search Results

The file numbers which correspond to each lexicographer files are encoded in several parts of WordNet system as an efficient way to indicate a lexicographer files name. The file lexnames[1] lists the mapping between file names and numbers and can be used to correlate between programs or end users. The Results Window plays a pivotal role to display the search word from WordNet database. The sense matching in the search results will be displayed in particular format. Figure 1 show the format used to structure the words in WordNet.

The descriptions for each syntax in the format are discussed in Table 1.

Ri Ta. Word Net: Many ways have been done to produce a good system or model. One of that is by using tools such as CONCEIVER++, an understanding-based program debugger for object-oriented programming language [9] and Rational Rose that applies UML to cover semantic domain and strong architecture or design [10]. A tool is also needed to prove the Framework for Analyzing Semantic of Object-Oriented Model (FASOOM). RiTa covers a range of computational language tasks including text analysis, generation, display and animation, text-to-speech, text-mining and access to external resources such as WordNet. Most of the RiTa toolkit is implemented as a Java library consisting of ten independent packages. The core collection of objects contain approximately 20 classes within the rita.* package, all of which follow the same usage and naming conventions. Additional packages provide support for these core objects, but are not directly accessed in typical usage [11]. RiTa.WordNet *(RiWordNet)* as one of the core objects in RiTa tools has been chosen for this purpose which provides straightforward access to the WordNet ontology. It supports all the common WordNet relation types, including synonyms, antonyms, hypernyms,

Table 1: Description of WordNet Search Results Format

Structure	Description
Sense n	Sense number of the search word
synset_offset	Byte offset of the synsets in the data.pos file corresponding to the syntactic category (N, V, Adj, Adv)
lex_filename	Name of the lexicographer file that the synsets come from
word1	First word in the synsets (not necessarily the search word)
sense_number	WordNet sense number assigned to the preceding word
synset_offset, lex_filename and sense_number	Generated if the appropriate Options are specified

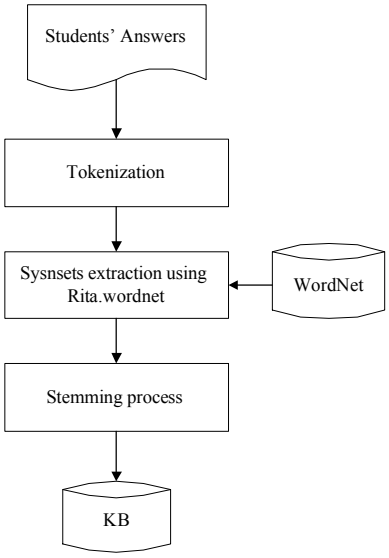

Fig. 2: Synsets Extraction Process

& hyponyms, holonyms, meronyms, coordinates, similars, nominalizations, verb-groups, derived-terms, glosses, "see-alsos", examples and descriptions, as well as distance metrics between terms in the ontology [12]. The next section will discuss in detail about process in synsets extraction.

Synsets Extraction Process: Synsets extraction process is a part of FASOOM framework that has been discussed in detail in [13]. This phase describes several steps in synsets extraction process from WordNet using Rita.WordNet as shown in Figure 2.

Students' Answers: Students' answers are UML class diagram that have been extracted and stored in a file. The properties of UML class diagram that included in file are name of UML class, attributes, operator, parameter and type.

- UML class name: The name of class
- Attributes: Instances of property that owned by the class. As an example; the class Transaction can have the attributes such as date and time.
- Operator: Represent the functions or tasks that can be performed on the data in the class. As an example, the class Account can have the operator likes withdraw and deposit.
- Parameter: A parameter specifies a type of argument and the value it takes in the call to an operation. An operation can have any number of parameters or none at all.
- Type: A data type is a classifier whose instances are identified only by their value. For example; date, time, string, integer and many more.

Tokenization: A token is an instance of a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing. The process of breaking a text up into its constituent tokens is known as tokenization. Standard methods in tokenization are [14]:

- Separate on whitespace
- Alphabetic strings
- Alphanumeric strings

Students' answers stored in a file are ordered by UML class diagram properties. Hence, tokenization is implemented using the separate on whitespace method to make sure that only UML class name are extracted from students answers which stores in a file.

WordNet Database: WordNet store synsets with its definition and relation. A list of synsets has been extracted from WordNet after a search word is inserted. These synsets can be used to compare the synonyms in UML class name.

Extraction using Rita.WordNet: Rita.WordNet provides some *getAllSynsets()* method of Rita.WordNet has been used for extracting synsets from WordNet. This method returns string of words (in array) in each synset for all senses of word with part of speech (nouns, verbs, adjectives and adverbs), or null if not found.

Stemming Process: Some synsets that were extracted from WordNet contains suffix. Hence, stemming techniques used to produce the root of the resulting synsets. Thus, Porter Stemmer Algorithm was chosen to perform this technique. Examples of synsets that has been stemmed is shown in Table 2.

Evaluating the stemming algorithms will lead into two common problems for word standardization such as under-stemming errors or over-stemming errors [15]. In this research, some of the synsets extracted have no meaning after stemming process is applied because of over-stemming errors. Examples of synsets with over-stemming errors are shown in Table 3.

Table 2: Examples of Synsets Before and After Stemming

Before Stemming	after Stemming
Accounting	account
Exploiter	exploit
Segmentation	segment
Categorization	categorization
Sorting	sort

Table 3: Example of Synsets with Over-Stemming Errors

Before Stemming	after Stemming
Division	divis
section	Sect
Explanation	Explan
History	Histori
Bill	Bil

Several suffixes to be removed have been chosen to cope with over-stemming errors except -ATION, -ATOR, -EMENT, -MENT, -ENT, -ION and -E. We only choose certain suffix to be removed because of error detected when synsets are extracted.

Knowledge Base: The end result is the synsets that were extracted after the stemming process will be stored in knowledge base. These synsets are stemmed synsets extracted from WordNet. Details of the result from synsets extraction will discuss in the next section.

RESULTS AND DISCUSSION

Synonyms are important to make sure that answers given by students are true. For the early stage, a simple program is designed to extract synonym sets (synsets) using Rita.WordNet. Several words that usually used as class name in UML have been tested and the outputs are shown in Table 4.

Eight words are used as a sample data in order to produce the synonym sets results. They are student, educatee, class, department, customer, user, account and university. Student and educatee actually are two words those synonyms to each other. We test these two words to see that if they can give the same output.

As shown in Table 4, a search word can gives several synsets or no synsets at all. This provides several choices for students in naming the UML class diagram correctly even though they are not provided in answers scheme. Rita.WordNet has been used to extract synsets from WordNet. Other extractor tools such as TextCatch[2], TextToOnto[3] and Email Extractor[4] have been reviewed and obviously Rita.WordNet is the best choice. The used of RiWordNet are because of its significant in this research. We can conclude that RiWordNet are:

Table 4: Synsets Output from Rita.WordNet

Search Words	Synsets
student	Synsets 0: pupil
	Synsets 1: educatee
	Synsets 2: scholar
	Synsets 3: bookman
Educatee	Synsets 0: student
	Synsets 1: pupil
class	Synsets 0: category
	Synsets 1: family
	Synsets 2: form
	Synsets 3: grade
	Synsets 4: course
	Synsets 5: stratum
	Synsets 6: division
	Synsets 7: year
Department	Synsets 0: section
client	Synsets 0: customer
	Synsets 1: node
	Synsets 2: guest
user	Synsets 0: exploiter
account	Synsets 0: history
	Synsets 1: chronicle
	Synsets 2: story
	Synsets 3: report
	Synsets 4: explanation
	Synsets 5: score
	Synsets 6: accounting
	Synsets 7: bill
	Synsets 8: invoice
university	No Synsets!

- Easy to use and understand
- Access directly to WordNet database through RiTaServer.
- Give the simple output as requested
- Easy to customize with own need (use Java programming)

Even though RiTa.WordNet gives a lot of benefits compared to other tools, some problems appear when we search the words in the same synsets. As an example in Table 4, when we search for the word *educatee*, *student* and *pupil* have appeared to be the synsets for educatee. But, when we search for the word student, there are two other more synsets such as *scholar* and *bookman*. If student is a synonym to *educatee* and *scholar* is a synonym to *student*, we can say that the word *scholar* should be also synonym to *educatee*. However, the results that we have gained show that RiTa.WordNet cannot give the overall synsets for the search words. So, some modification needs to be done in order to achieve the all synsets for a search word.

After the synsets has been extracted using Rita.WordNet, stemming process will be starting to return the synsets into its stem. For example, the synsets "section" will become "sect". The use of stemming process is to expand the scope of the search synsets for the given words. However, modification of the Porter Stemming Algorithm has been done to avoid some problems in stemming process such as under-stemming and over-stemming.

CONCLUSION

Some problems in modeling can be overcome by proposing extraction methods for WordNet's synsets. This method will be adapted to analyze the semantics in object-oriented model. The use of WordNet for applying its synsets that have been extracted into the study of semantic is an alternative to the use of WordNet in different way.

The lack of synsets obtained from Rita.WordNet will be marked as potential issue for future study. An approach that is considered appropriate and may be used is a Breadth-First Search (BFS) or Depth-First Search (DFS) theory in the synsets extraction process. These theories will be used to find neighboring words around the query word which are related to each other through WordNet hypernymy-hyponymy relation [16].

ACKNOWLEDGMENT

This research was supported by a grant Tabung Bantuan Pendidikan Khas (TBPK), Universiti Malaysia Terengganu (Vot: 53057) and National Science Fellowship (NSF) under Ministry of Science, Technology and Innovation (MOSTI) Malaysia.

REFERENCES

1. Booch, G., J. Rumbaugh and I. Jacobson, 2005. Unified Modeling Language User Guide, The (2nd Edition) (Addison-Wesley Object Technology Series). Addison-Wesley Professional
2. Thomasson, B., M. Ratcliffe and L. Thomas, 2006. Identifying novice difficulties in object oriented design. In the Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education.
3. Unhelkar, B., 2003. Process Quality Assurance for UML-Based Projects. Addison-Wesley Professional
4. Noor Maizura Mohamad Noor, Noraida Haji Ali and Noor Syakirah Ibrahim, 2010. A New Framework to Extract WordNet Lexicographer Files for Semi-Formal Notation: A Preliminary Study. In the 4th Internationatiol Symposium on Information Technology 2010 (ITSim'10), pp: 1027-1031.
5. Wang, T. and G. Hirst, 2009. Extracting synonyms from dictionary definitions. In the Recent Advances in Natural Language Processing.
6. Miller, G.A., 1995. WordNet: A Lexical Database for English. Communication of The ACM, 38(11): 39-41.
7. R. Beckwith, G. A. Miller and R. Tengi, 1998. Design and Implementation of the WordNet Lexical Database and Searching Software in WordNet: an electronic lexical database, C. Fellbaum, Editor. 1998, MIT Press: Massachusetts, pp: 105-127.
8. Ibrahim, N.S., N.H. Ali and N.M.M. Noor, 2011. A Study on the Structure of WordNet Lexicographer Files. In the Universiti Malaysia Terengganu 10th International Annual Symposium (UMTAS) 2011.
9. Nor Fazlida, S., Mohd, Z. Abdullah Mohd, I. Sufian and S. Zarina, 2005. Designing an understanding and debugging tool (UDT) for object-oriented programming language. In the Proceedings of the 4th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering Data Bases.
10. Ali, N.H., Z. Shukur and S. Idris, 2007. Assessment System for UML Class Diagram Using Notations Extraction. International Journal of Computer Science and Network Security, 7(8): 181-187.
11. Howe, D.C., 2009. Creativity Support for Computational Literature, in Department of Computer Science. 2009, New York University: New York, pp: 250.
12. Howe, D.C., 2008. RiTa: Creativity Support for Computational Literature. In the SIGCHI 2008, pp: 4.
13. Ali, N.H., N.S. Ibrahim, N.F.M. Sani and N.M.M. Noor, 2011. Analyze Semantic of Object-Oriented Model Using RiTa.WordNet. Journal of Computing, 3(5): 61-66.
14. Rennie, J., 2003. Text Classification, pp: 1-47.
15. Paice, C.D., 1994. An evaluation method for stemming algorithms. In the Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, pp: 42-50.
16. Sun, K.T., Y.M. Huang and M.C. Liu, 2011. A Wordnet-Based Near Synonyms and Similar-Looking Word Learning System. Educational Technology & Society, 14(1): 121-134.