Validating Learning Outcomes of an E-Learning System Using NLP

Eiman Aeiad and Farid Meziane

School of Computing, Science and Engineering
University of Salford, Salford, M5 4WT, UK
E.Aeiad@edu.salford.ac.uk,F.Meziane@salford.ac.uk

Abstract Despite the development and the wide use of E-Learning, developing an adaptive personalised E-Learning system tailored to the needs of individual learners remains a challenge. In an early work, the authors proposed APELS that extracts freely available resources on the web using an ontology to model the leaning topics and optimise the information extraction process. APELS takes into consideration the leaner's needs and background. In this paper, we developed an approach to evaluate the topics' content extracted previously by APELS against a set of learning outcomes as defined by standard curricula. Our validation approach is based on finding patterns in part of speech and grammatical dependencies using the Stanford English Parser. As a case study, we use the computer science field with the IEEE/ACM Computing curriculum as the standard curriculum.

Keywords: NLP, Keyword Extraction, Key Phrases, Dependency Relation

1 Introduction

E-learning is a modality of learning using Information and Communication Technologies (ICTs) and advanced digital media [14]. It offers education to those who cannot access face to face learning. However, the needs of individual learners have not been addressed properly [1]. A system to address this problem was proposed by Aeiad and Meziane [1] in the form of an E-Learning environment that is Adaptive and Personalised using freely available resources on the Web (the APELS System). The aim of APELS is to enable users to design their own learning material based on internationally recognised curricula and contents. It is designed to first identify learners' requirements and learning style and based on their profile, the system uses an ontology to help in extracting the required domain knowledge from the Web in order to retrieve relevant information as per users'requests. A number of modules were developed to support this process [1]. The contents of the retrieved websites are then evaluated against a set of learning outcomes as defined by standard curricula and this constitutes to the main focus of this paper. We used a set of action verbs based on Bloom's taxonomy[2] to analyse the learning outcomes. Bloom's taxonomy classifies action verbs into six levels representing the following cognitive skills: Remembering, Understanding, Applying, Analysing, Evaluating and Creating. For example, action verbs such as define, describe and identify are used to measure basic levels of cognitive skills in understanding, while action verbs such as carry out, demonstrate, solve, illustrate, use, classify and execute are used to measure basic levels of the applying cognitive skills. In addition, we used the Stanford Parser, an implementation of a probabilistic parser in Java which comprises a set of libraries for Natural Language Processing (NLP) that together make a solid unit capable of processing input text in natural language and produces part-of-speech (PoS) tagged text, context-free phrase structure grammar representation and a typed dependency representation [6]. In this paper, a case study using the IEEE/ACM Computing Curriculum [12] will be used to illustrate the functionality of APELS.

The rest of the paper is structured as follows: the next section reviews some related work by outlining different approaches for extracting keywords and keyphrases. Section 3 presents a revised architecture of the APELS system and the knowledge extraction module in details. Section 4 illustrates the functionality of the APELS system using examples from the ACM/IEEE Computer Society Computer Science Curriculum. and the system evaluated. Finally, conclusion and future developments are given in section 5.

2 Project Background

Personalised E- learning systems have attracted attention in the area of technology-based education, where their main aim is to offer to each individual learner the content that suits her/his learning style, background and needs. Previously, we designed APELS that extracted information from the WWW based on an ontology and tailored to individual learners' profile [1]. However, the suitability of the contents of the selected websites should be evaluated to ensure that they fit the learner's needs and this will be addressed in this paper. Matching the content to learning outcomes of curricula, is very important when assessing the suitability of the selected websites. Learning outcomes are statements of what a student is expected to know, understand and/or be able to demonstrate after the completion of the learning process [7]. Each learning outcome contains an action verb followed by usually a noun phrase that acts as the object of the verb. Together, the action verbs and noun phrases are referred to as Keywords or key phrases. These are used in academic publications to give an idea about the content of the article to the reader as they are a set of representative words, which express the meaning of an entire document. Various systems are available for keywords extraction such as automatic indexing, text summarization, information retrieval, classification, clustering, filtering, topic detection and tracking, information visualization, report generation, and web searches [3]. Automatic Keyword Extraction methods are divided into four categories: statistical methods [5], machine learning methods [13], linguistic methods [9] and hybrid methods [10].

The APELS architecture given in Figure 1, is a revised version of the one proposed in [1]. It is modified based on our experience while developing this project. APELS is based on four modules: student profile, student requirement, knowledge extraction and content delivery. The student profile and the student requirement modules are similar to the ones presented in [1]. We updated the knowledge extraction module adding the learning outcomes validation in order to evaluate the topics' contents against a set of learning outcomes as defined by standard curricula and the details of this module are given in section 3.

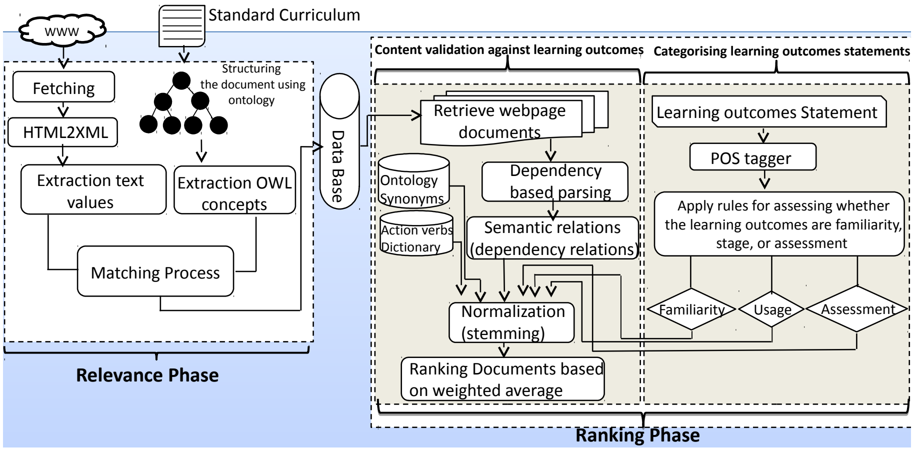
www
Standard Curriculum
Fetching
HTML2XML
Structuring the document using ontology
Extraction text values
Extraction OWL concepts
Matching Process
Relevance Phase
Data Base
Content validation against learning outcomes
Categorising learning outcomes statements
Retrieve webpage documents
Ontology Synonyms
Dependency based parsing
Action verbs Dictionary
Semantic relations (dependency relations)
Normalization (stemming)
Ranking Documents based on weighted average
Learning outcomes Statement
POS tagger
Apply rules for assessing whether the learning outcomes are familiarity, stage, or assessment
Familiarity
Usage
Assessment
Ranking Phase

Figure 1. Knowledge Extraction Module

3 The Knowledge Extraction Module

The Knowledge extraction module is responsible for the extraction of the learning resources from the Web that would satisfy the learners' needs and learning outcomes. The Module comprises two phases; the Relevance phase and the Ranking phase. The relevance phase uses an ontology to retrieve the relevant information as per users' needs. In addition, it transforms HTML documents to XML to provide the information in a friendly accessible format and easier for extraction and comparison. Moreover, we implemented a process called the matching process that computes the similarity measure between the subset of the ontology that models the learning domain and the values element extracted from the websites. The website with the highest similarity is selected as the best matching website that satisfies the learners' learning style.

After the matching process occurs, there is a further step that is required to evaluate that content adheres to a standard set of guidelines for studying the chosen subject. Hence, the learning outcome validation was added to ensure the selection of the most relevant websites that satisfy the learning outcomes set by standard curricula. This is the purpose of the ranking phase that is composed of two components (i) categorising learning outcomes statements and content validation against learning outcomes.

3.1 Categorising learning outcomes statements

The learning outcomes statements are analysed by selecting a set of action verbs based on the Bloom's taxonomy [2]. Each learning outcome contains an action verb associated with the intended cognitive level of the bloom's Taxonomy, followed by the object of the verb (specific subject material). The Stanford parser is used as the pre-processor of the input statement, which is a learning outcome. It takes the learning outcome statement, written in natural language, and marks it with the PoS tagger, builds tree representation of the sentence from the sentence's context-free-phrase-structure-grammar parse, and eventually builds a list of typed dependencies [6]. Here we used only the PoS tagger to analyse the learning outcome and followed by Nouns and Verbs Extractor to classify the learning outcomes. The PoS tagger is used to identify the nouns and verbs by tagging each word in the text (e.g. 'drink': verb, 'car': noun, 'lucky': adjective, 'vastly': adverb, 'on': preposition etc). It has been widely proposed by many authors[4,8] as the main task for analysing the text syntactically at the word level. After all the words in the learning outcomes statements are tagged, Nouns and Verbs Extractor is used to extract the nouns and verbs by selecting the pattern tags of the PoS. The current pattern tags of Stanford parser is defined as follow:

define/VB and/CC describe/VB variable/NN ./.

A set of rules are used to identify the learning outcomes statement by searching the pattern token in the tagged verb in the action verbs dictionary that have been manually defined based on the Bloom's Taxonomy. The rules that are used to assign learning outcomes based on action verbs in bloom's Taxonomy have the form:

```
if pattern token in tagged verb belongs to Level A,
  then learning outcome = "A"
```

The six levels representing of the cognitive skills defined in section 1. We associate a set of action verbs with each level which will be used to identify the level. The actions verbs associated with the Applying level for example are also given in section 1.

3.2 Content validation against learning outcomes

The evaluation of the topic's content will be against the identified learning outcomes statements. The Stanford typed dependencies representation is used to extract a topic name, an action verb and their relationship. Moreover, we adopted a rule based linguistic method to extract key phrases and keywords from text. The Stanford typed dependencies representation provides a simple description of the grammatical relationships in a sentence, establishing relationships between "head" words and words which modify those heads ("refer"). Furthermore, the Stanford dependency parser consists of three variables namely: Type dependency name, governing of the dependency and Subordinate of the dependency.

To extract action verbs, topic names and their relationships, two types of dictionaries are used. The action verbs dictionary that contains the action verbs that have been manually defined based on the Bloom's Taxonomy and the topic name synonym dictionary whose terms are retrieved from the ontology. The system checks the output of a typed dependency pattern to check if the governor of the dependency is an action verb and its subordinate a topic name or if the governor of the dependency is a topic name and its subordinate an action verb. Moreover, we used Porter's stemming Algorithm [11] to produce the roots of the words. Once the Stanford parser produces the typed dependency between a pair of words, these are analyzed to get the root of the word that will be looked up in the action verb dictionary and the topic name synonyms from the ontology. The other distinctive feature of Normalisation (stemming) is to reduce the size of the action verbs dictionary and topic name synonym as they contain all the different forms of the word.

The typed dependency parsing approach is only used to analyze the text in order to identify the potential relationship between the action verbs and topic names. This is not enough to fully validate the content against the learning outcomes. Hence, we used rule based linguistic methods to filter out the key phrases and keywords by using the linguistic features of the word (i.e., PoS tags) to determine key phrases or keyword from the text. These rules are employed to identify familiarity, usage, and assessment levels which are illustrated in the case study section.

4 Case Study and Evaluation

4.1 Description of the Case Study

The ACM/IEEE Computer Science Society Curriculum [12] was used to illustrate the functionality of APELS . The IEEE/ACM Body of Knowledge (BoK) is organized into a set of 18 Knowledge Areas (KAs) corresponding to typical areas of study in computing such as Algorithms and Complexity and Software Engineering. Each Knowledge Area (KA) is broken down into Knowledge Units (KUs). Each KU is divided into a set of topics which are then classified into a tiered set of core topics (compulsory topics that must be taught) and elective topics (significant depth in many of the Elective topics should be covered). Core topics are further divided into Core-Tier1 topics and Core-Tier2 topics (Should almost be covered). The software development fundamentals area for example is divided into 4 KUs. The Algorithm and design KU is divided into 11 Core-Tier1 topics. Learning outcomes are then defined for each class of topics. We will specifically look at designing an advanced programming module in C++ from the IEEE/ACM fundamental Programming Concepts KU using APELS. Moreover, we used the learning outcomes that have an associated level of mastery in the Bloom's Taxonomy, which have been well explored within the Computer Science domain based on the IEEE/ACM Computing curriculum. The level of mastery is defined in the Familiarity, Usage and Assessment levels. Each level has a special set of action verbs. The linguistic rules used in APELS include:

- **Rule 1:** At the "Familiarity" level, the student would be expected to know the definition of the concept of the specific topic name in the content text. Thus this rule is utilized to extract the key phrases when the topic name is followed by verb "to be" expressed as "is" and "are" such as in the phrases "variable is" and "algorithms are". In these kind of key phrases the noun "variable" does not depended on the verb to be "is". The PoS tag is used to identify the grammatical categories for each word in the content of the text. Then, the system will extract a noun followed by the verb "to be" by selecting the pattern token in the tagged noun followed by the pattern's token in the tagged verb. We first identify the token with the noun tag in the topic name synonyms from the ontology and check if it is followed by the token with the verb tag ("is" or "are" in this case).
- **Rule 2:** At the "usage" level, the student is able to use or apply a concept in a concrete way. Using a concept may include expressions made up of two words such as "write program", "use program" and "execute program". In these expressions, where words such as "write", "use" and "execute" are dependent on "program", the system is able to recognize these expressions automatically from the text using dependency relations.
- **Rule 3:** Students who take courses in computer science domain will have to apply some techniques or use some programs. Therefore, the content may include examples to illustrate the use these concepts. To search whether the content has terms such as ”example” or ”for example”. A PoS tagger is used to tag each word in the text. The system will then extract nouns by selecting the pattern token in the tagged noun. Finally, the system checks if the pattern token in the tagged noun matches with the word "example".
- **Rule 4:** at the "assessment" level, we have designed a special kind of rules because at this level there are two types of information that needs to be evaluated. First, the student is required to understand a specific topic and be able to use the topic in a problem solving scenario for example. In this case the system will apply rules 1 to 3 for the specific topic. Second, the student should be able to select the appropriate topic among different topics, hence the system apply again rules 1 to 3 for each topic.

4.2 Results and Evaluation

. The APELS system produced a list of websites for learning the C++ language with the highest accuracy rating [1]. Now, the validity of these selected websites will be assessed by matching their content to the targeted learning outcomes as described by IEEE/ACM curriculum. We selected one of the outcome "Define and describe variable" to be tested by our system. First, the system identified the learning outcome in the Familiarity level because it contains the action verbs "define" and "describe". Hence in this case, three kinds of grammatical patterns are implemented for keywords and key phrases extraction as shown in Table 2. One is the potential relationship between the action verb and the topic name, second if the potential syntactic structure of the sentences includes a noun phrase followed by a verb phrase and the third pattern of the PoS includes a noun phrase.

The system ranks the relevant documents using the weighted average method, which is used to calculate the average value of a particular set of occurrence of keywords and key phrases in a document with different weights. The weighted average formula is defined as follows:

$$WeightAvg(x) = w_1x_1 + w_2x_2 + w_3x_3 + ... + w_nx_n$$

Where W= weight, x= occurrence of keywords and key phrases.

The weights are determined based on the importance of each mastery level of the learning outcomes. The weights are chosen manually where sentence definition of the topic name taken from Rule 1 is worth 0.70. This is because understanding the definition of the concept is most important for the students before they do any further learning around it. For example, the learning outcome "define and describe variables", if the student does not understand the variable concept he/she might not be able to implement it in their work. The dependency relationship between action verbs and topic name and Rule 3 are both given a weighted 0.15. This is because both criteria have the same importance. To calculate a weighted average , each value must first be multiplied by its weight. Then all of these new values must be added together. Thus, the overall calculation for the website (www.cplusplus.com/doc/tutorial/variables) would be (43 * 0.15) + (8 * 0.70) + (16 * 0.15) = 14.45. It is crucial that each criteria are given the correct weights based on their importance. If more weights were given to less important criteria compared to important one, it would give inaccurate ranking for the WebPages. For example, although (www.cplusplus.com/doc/tutorial/variables) has highest dependency relation, (fresh2refresh.com/c-programming/c-variables) ranked first because it has the highest score for Rule 1 which is most important.

Table 1. Ranking of WebPages based on Weighted Average

WebPages	**Occurences**			**Weighted**	**Average**		**Total**
	Dependency Relation	Rule 1	Rule 3	*0.15	*0.70	*0.15	
fresh2refresh.com/c-programming/c-variables	16	15	15	2.4	10.5	2.25	15.15
www.cplusplus.com/doc/tutorial/variables	43	8	16	6.45	5.6	2.4	14.45
en.wikibooks.org/wiki/C- Programming/Variables	32	4	20	4.8	2.8	3	10.6
microchip.wikidot.com/tls2101:variables	37	2	3	5.55	1.4	0.45	7.4
www.doc.ic.ac.uk/ wjk/C++Intro/RobMillerL2.html	11	2	25	1.65	1.4	3.75	6.8
www.penguinprogrammer.co.uk/c-beginners-tutorial/variables	11	3	1	1.65	2.1	0.15	3.9

5 Conclusion and future work

Validating the website content against the learning outcome would add a great value to the system making it more specific. In the future work, we need to finalise the system so that the leaner will be able to adapt and modify the content and learning style based on the interactions of the users with the system over a period of time. The information extracted by the system will be passed to a Planner module that will structure it into lectures/tutorials and workshops based on some predefined learning times.

References

1. Eiman Aeiad and Farid Meziane. An adaptable and personalised e-learning system based on free web resources. In Chris Biemann, Siegfried Handschuh, André Freitas, Farid Meziane, and Elisabeth Métais, editors, *Proc. 20th Inter. Conf. on Applications of Natural Language to Information Systems*, LNCS, pages 293–299, Passau, Germany, 2015. Springer.
2. Benjamin S. Bloom, Max D. Engelhart, Edward J. Furst, Walker H. Hill, and David R. Krathwohl. *Taxonomy of Educational Objectives: Handbook 1, The Cognitive Domain.* Allyn & Bacon, Boston, 1956.
3. David B Bracewell, Fuji Ren, and Shingo Kuriowa. Multilingual single document keyword extraction for information retrieval. In *Proc. Inter. Conf. on Natural Language Processing and Knowledge Engineering*, pages 517–522. IEEE, 2005.
4. Eric Brill. A simple rule-based part of speech tagger. In *Proc. 3rd Conf. on Applied Natural Language Processing*, pages 152–155, Stroudsburg, PA, USA, 1992. ACL.
5. Jonathan D. Cohen. Highlights: Language- and domain-independent automatic indexing terms for abstracting. *Journal of the Association for Information Science and Technology*, 46(3):162– 174, 1995.
6. Marie-Catherine De Marneffe and Christopher D Manning. Stanford typed dependencies manual. Technical report, Stanford University, 2008.
7. ECTS Users' Guide. Computer science curricula. http://ec.europa.eu/education/programmes/socrates/ects/doc/guide_en.pdf, 2005.
8. Mark Hepple. Independence and commitment: Assumptions for rapid training and execution of rule-based pos taggers. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, pages 278–277, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.
9. Anette Hulth. Improved automatic keyword extraction given more linguistic knowledge. In *Proc. of the 2003 Conf. on Empirical Methods in NLP*, pages 216–223, 2003.
10. B. Krulwich and C. Burkey. Learning user information interests through extraction of semantically significant phrases. In *Proceedings of the AAAI spring symposium on machine learning in information access*, pages 100–112, 1996.
11. Martin F Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
12. ACM/IEEE Societies. Computer science curricula. http://www.acm.org/education/CS2013-final-report.pdf, 2013.
13. Yasin Uzun. Keyword extraction using naive bayes. Technical report, Bilkent University, Computer Science Dept., Turkey, 2005.
14. Tim L Wentling, Consuelo Waight, Danielle Strazzo, Jennie File, Jason La Fleur, and Alaina Kanfer. The future of e-learning: A corporate and an academic perspective. *http://learning.ncsa.uiuc.edu/papers/elearnfut.pdf*, 2000.