EasyChair Preprint

№ 6621

eCSCDA: an Efficient System for Analyzing Contents of Computer Science Courses

Peerapon Kamlangpuech and Komate Amphawan

September 16, 2021

eCSCDA: An efficient system for analyzing contents of Computer Science Courses

Peerapon Kamlangpuech
Computational Innovation Laboratory,
Faculty of Informatics, Burapha University
Chonburi, 20131, Thailand
Email: peerapon.peetom@gmail.com

Komate Amphawan
Computational Innovation Laboratory,
Faculty of Informatics, Burapha University
Chonburi, 20131, Thailand
Email: komate@gmail.com

***Abstract*—This paper aims to introduce a new efficient system, called *eCSCDA*, to efficient analyze Computer Science (CS) course description. The primary task of the system is to identify similar (and dissimilar) contents amount two (or a group of) CS course descriptions which can help to know similar and different focuses on important contents to teach to students. Moreover, it can help to check for integrity and quality and to set up a standard of teaching contents of the course. In *eCSCDA*, text processing procedure is newly rearranged and developed. Besides, the linguistic rules and their derivation are newly updated and applied to extracted important keywords. Moreover, two synonym corpuses, terminology and word synonyms, are recently designed and collected to consider synonyms of the keywords hidden in the course descriptions. Last, to efficiently identify similar contents, two new matching techniques, sub-keyword and semantic matching techniques, are designed and applied together with exact and subset (superset) matching methods. Experiments were conducted on CS course contents gathered from nine Thai Universities to examine the effectiveness of our proposed system in comparison with previous system and related methodologies. From the results, it shows that *eCSCDA* is efficient to analyze the course contents and outperforms other related systems in various terms *e.g.* percentage of similar contents, precision, recall and F-measure, respectively.**

***Index Terms*—Content analysis, Course description, Computer Science course**

I. Introduction

Based on the rapid growth of computer technology, new concepts, tools, software, applications, programming languages and libraries are being developed each day. This led to the boom of learning in computer-related fields such as Computer Science (CS), Information Technology (IT), Computer Engineering (CE), Software Engineering (SE), Data Science (DS) and so on. With these emergences, institutes and universities should create news, revise and/or update their current curriculum in order to to keep up with the world. Some courses might be newly created in the curriculum meanwhile, some might be modernized with new contents. In addition, there are some ideas that core courses of the curriculum should be standardized. With these ideas, there are approaches to observe consistency between courses in the curriculum and that of *TQF:HEd* (*Thai Qualification Framework of Higher Education*) [1], [2], [3]. These approaches take contents of each course into account and then map them into a class of "Body of knowledge" by applying *semantic-based* and *structure-based ontology mapping*. Besides, Bloom's Taxonomy is applied to assess learning objectives of CS courses [4], [5], [6]. From these approaches, each of topic of teaching contents is considered and then its level of knowledge is extracted and mapped to Bloom's Taxonomy levels (*i.e. i*) recall, *ii*) comprehension, *iii*) application, *iv*) analysis, *v*) synthesis and *vi*) evaluation, respectively). This can assist to assess students' performance and report difficulties of varieties of causes hypothesized and solutions adopted. Moreover, there is an effort to calculate similarity among teaching contents [7] by considering career name, course name, course description and contents to solve students' mobility and credit validation.

From the above, focusing on checking the contents with Thai Quality framework is quite out of date since TQF:HEd is not updated. Moreover, one with focusing on finding similarity among teaching contents does not focus on the computer technology curriculum which have lots of special terminologies, reserved words and abbreviations. This may lead to losing of focusing on the important contents that should be considered and losing efficiency of similarity matching and extracting similar contexts from comparing two course contents. From these issues, a system, called *CSCDA* (Computer Science Course Description Analysis), is introduced [8]. The *CSCDA* takes two course descriptions (either on the same or different subjects) as input. It then performs text processing and extracts important contents (*i.e.* keywords) from each course description. Next, keywords of one course description are compared with ones from another course description to gain similar/dissimilar contents and to calculate level of similarity between the two course contents. These information can help to investigate and check for redundancy, integrity, popularity and quality of contents in the course descriptions. However, even *CSCDA* can well perform in analyzing course contents, but it still has not high precision and recall due to it applies only lexical similarity matching. Thus, there is room to improve the ability of the *CSCDA* system by considering semantic of contents of the course description.

Thus, this paper aims to improve the efficiency of the *CSCDA* by introducing a new improved system, called *eCSCDA* (efficient *CSCDA)*. In *eCSCDA*, text processing procedure is revised in order to efficiently collect important words. Terminology detection is improved by doing twice,

once before and after word stemming & lemmatization. New linguistic rules and their derivations are applied to accurately extract keywords. Two synonyms corpuses are newly prepared to consider the semantic of contents. Last, two new matching methods are designed and applied together with two existing matching techniques used in the *CSCDA* system. Experiments were done on CS course descriptions gathered from nine Thai universities. Then, the percentage of similar contents, precision, recall and F-measure are applied to investigate efficiency of the proposed *eCSCDA* in comparison with *CSCDA* and *Word2Vec* [9]. From the results, it is shown that *eCSCDA* outperforms the others on all measures.

II. Related work

Measuring the similarity of texts based on considering words, sentences, paragraphs, and documents is an important task. It is widely applied in many tasks of information retrieval, automatic question-answering, machine translation, dialogue systems, document matching, plagiarism detection, text summarization, and so on. The similarity calculation is divided into 2 groups [10] as discussed as follows.

1) *Text distance*–describes the proximity between two texts, words, or phrases from the perspective of distance. There are three categories of text distance described as follows:
   - *Length Distance*–calculates similarity from the distance of two texts using numerical characteristics, *e.g.* Euclidean distance, Cosine similarity, Manhattan distance, etc.
   - *Distribution Distance*–computes similarity by investigating the distribution of texts such as JS divergence, KL divergence, and so on.
   - *Semantic Distance*–considers distance of texts at the semantic level.
2) *Text Representation*–represents the texts as numerical features where texts can be similar in lexically or semantically. Words in texts are lexically similar if they have the same character sequence. Meanwhile, they are semantically similar if they are used in the same way or same context. This technique can be divided into 4 categories.
   2.1) *String-Based*–measures similarity by considering string sequences and character composition which consisting of *i*) Character-Based–considers similarity between characters *e.g.* editing distance, LCS (longest common substring), and Jaro similarity; and *ii*) Phrase-Based–considers similarity phrase words *e.g.* Jaccard and dice coefficient.
   2.2) *Corpus-Based*–uses additional information collected in a corpus, *e.g.* textual feature or co-occurrence probability, to calculate similarity *e.g.* distributed representation, bag-of-words model, and matrix factorization methods.
   2.3) *Semantic Text Matching*–determines similarity of texts by their meaning *e.g.* Single semantic text matching–and Multi-semantic document matching.
   2.4) *Graph Structure*–calculates text similarity by regarding links between nodes of the graph *e.g.* Knowledge Graph–projects entities and relationships in the graph into a continuous space; and Graph Neural Network–captures dependency of the graph through message transmission between nodes.

From the various methods mentioned above. In this research, we applied String-Based (both on Character-Based and Phrase-Based), Semantic Text Matching (Single Semantic Text Matching), and Graph Structure (Knowledge Graph: is-a-part-of) methods. These methods allow us to calculate similarities between keywords from course descriptions.

III. Proposed system

In this section, components and details of computation of the proposed *eCSCDA* system are described. As in Fig. 1, the system consists of three main procedures as follows.

A. Input and preprocessing

Before getting input, *eCSCDA* prior prepares three corpus and linguistic rules for further computation. First, as in [8], $28,392$ terminologies in Computer domain were gathered from eight well-known sources and stored in terminology corpus. Second, terminology synonym corpus is created by considering each terminology of the terminology corpus and then searched for synonyms from three sources *i.e. Longdo Dictionary*, *google translation corpus* and *Cambridge Dictionary*, respectively. Note that we also tried to consider the other sources but most of them provide too many synonyms with different levels of relevance with the target terminology. Third, word synonym corpus is built by considering each word from *www.dictionary.com* and then looks for its synonyms in the same manner as above. Last, linguistic rules from [11], [12] are applied to extract important contents which are in the form of keywords and/or terminologies.

Next, to feed input to the *eCSCDA* system, two (a group of) CS course descriptions (in English) should be in the form of 2-tuple $\langle s, cc \rangle$ (see Fig. 2) where s is the course name, and cc is teaching contents of the course.

B. Keyword Extraction

When any two course descriptions c_x and c_y (or a group of course descriptions $c_u, c_{u+1}, \ldots, c_v$) are input, their teaching contents cc_x and cc_y (or $cc_u, cc_{u+1}, \ldots, cc_v$) are first considered. Text-processing is performed on cc_x (also for the cc_y) by applying *i*) sentence tokenization, *ii*) word tokenization, *iii*) lowercase conversion *iv*) error correction, *v*) stopword removal, and *vi*) POS tagging, respectively. With these processes, the teaching content cc_x is divided into a set of topics, defined as $cc_x = \{tp_{1,x}, tp_{2,x}, \ldots, tp_{n,x}\}$. Then, each topic $tp_{i,x}$ is decomposed into a sequence of words with its tag to describe its duty in the topic, denoted as $tp_{i,x} =< (w_1^{tp_{i,x}}, tag), (w_2^{tp_{i,x}}, tag), \ldots, (w_n^{tp_{i,x}}, tag) >$.

Next, each n-gram of words of the topic $tp_{i,x}$ is considered to search for CS terminology hidden in the topic. This can help to recognize important contents. In this procedure, terminology

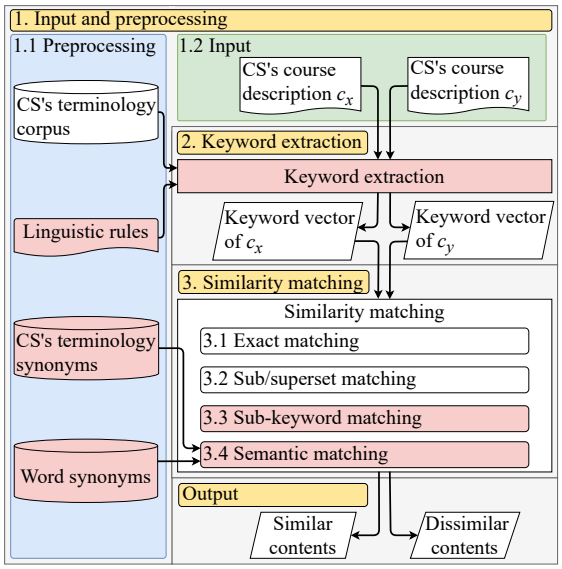

Fig. 1: The framework of the *eCSCDA* system

matching (comparing the n-gram with any terminology prior collected in the terminology corpus where the n-gram is thus grouped together and tagged as "TE" (terminology) if it matches with a terminology) is performed twice, once before and once after doing word stemming & lemmatization. Last, each n-gram of words of the topic $tp_{i,x}$ is reconsidered and the linguistic rules of [12] are thus applied to extract important keywords where a keyword might be in the form of *i*) Terminology, *ii*) Noun, *iii*) Adjective + Terminology, *iv*) Adjective + Noun, *v*) Noun + Noun, *vi*) Noun + Terminology, *vii*) Terminology + Terminology, *viii*) Terminology + Noun, *ix*)JJ + Noun + Terminology, or *X*) Noun + Terminology + Terminology, respectively. After this, a list of keyword of each topic $tp_{i,x}$ is collected as $K^{tp_{i,x}} = \{k_1^{tp_{i,x}}, k_2^{tp_{i,x}}, \ldots, k_n^{tp_{i,x}}\}$, but note that in practice, each topic mostly contains only one or two keywords.

C. Content matching

To recognize similar and dissimilar contents of c_x and c_y, each keyword list, $K^{tp_{i,x}}$ of topic $tp_{i,x}$ extracted from previous step, is considered. It is then compared to any keyword list $K^{tp_{u,y}}$ of $tp_{u,y}$ of course description c_y by the four matching techniques as follows:

1) exact matching – the keyword $K^{tp_{i,x}}$ is exactly the same as the keyword $K^{tp_{u,y}}$ (this also includes the case that $K^{tp_{i,x}}$ is equal to the paraphrase of words in $K^{tp_{u,y}}$),
2) subset matching – the keyword $K^{tp_{i,x}}$ is a subset of the keyword $K^{tp_{u,y}}$ (or $K^{tp_{i,x}}$ is a superset of the keyword $K^{tp_{u,y}}$),
3) sub-keyword matching – a part of the keyword $K^{tp_{i,x}}$ is exactly the same as a part of the keyword $K^{tp_{u,y}}$, and
4) semantic matching –the keyword or a part of the keyword $K^{tp_{i,x}}$ has the same semantic as the keyword or a part of the keyword $K^{tp_{u,y}}$ (Thanks to the terminology and word synonym corpuses), respectively.

From above, if the keyword $K^{tp_{i,x}}$ match with the keyword $K^{tp_{u,y}}$ by exact or subset matching, it can be concluded that the topic $tp_{i,x}$ of the course description c_x is similar to the topic $tp_{u,y}$ of the course description c_y. On the other hand, for matching on sub-keyword or semantic matching, it can be identified that topic $tp_{i,x}$ relates the topic $tp_{u,y}$.

Note that if the topic $tp_{i,x}$ matches with $tp_{u,y}$ by sub-keyword matching, the remaining words of $tp_{i,x}$ should be reconsidered and compared with the remaining words of $tp_{u,y}$ by semantic matching (also for semantic matching and then sub-keyword matching). If all of both keyword lists matches by these two cases, it can be concluded that the topic $tp_{i,x}$ is exactly the same as the topic $tp_{u,y}$.

When, the keyword $K^{tp_{i,x}}$ match with the keyword $K^{tp_{u,y}}$ by one of the four cases above, the matching score between c_x and c_y is set as 1 if the matched keyword is not equal to (or being subset of) the course name of c_x and c_y , calculated as follows :

$$match(tp_{i,x}) = \begin{cases} 1 & , \{\exists tp_{u,y} \in c_y | K^{tp_{i,x}} \text{ matches with } \\ & \quad K^{tp_{u,y}}, K^{tp_{i,x}} \not\subset s_x, K^{tp_{i,x}} \not\subset s_y\} \\ 0 & , \text{otherwise} \end{cases}$$

Last, after considering all topics in the course description c_x, the percentage of similar contents between the two course descriptions c_x and c_y can be calculated by

$$per\_sim(c_x, c_y) = \frac{\sum_{i=1}^{n} match(tp_{i,x})}{n} \quad (1)$$

where n is the number of topics in in c_x.

D. Example

Let's consider two course descriptions on "Probability and Statistics" from Burapha University (BUU) and King Mongkut's University of Technology Thonburi (KMUTT)) as shown in Fig. 2(a). First, the course from BUU is divided into 10 and that of KMUTT is also decomposed into 12 topics by applying text-processing as shown in Fig. 2(b). Second, terminologies hidden in each course description are recognized and labeled as "TE" as shown in the red highlight of Fig. 2(c). Third, by applying linguistic rules and their derivations, the two words, { ('descriptive', 'JJ'), ('statistics', 'TE')} of the topic $t_{1,BUU}$ are grouped and identified as a keyword, meanwhile, the tag of each word is still retained for matching procedure (see Fig. 2(d)). Fourth, matching of keywords from the course descriptions is performed, for example, the keyword { descriptive (JJ) statistics(TE)} of the topic $t_{1,BUU}$ matches with the keyword {statistics(TE)} of the topic $t_{1,KMUTT}$ by subset matching. Then, when we look at the matched word, {statistics(TE)} , it is a subset of course name ("Probability and Statistics") where does not indicate important content. It is then eliminated. The topic $t_{2,BUU}$ is also matched with $t_{1,KMUTT}$ by semantic matching, but it is also eliminated since the match keyword is a subset of the course name. For the keyword { probability(TE) principle(NN)} of the topic $t_{3,BUU}$, it matches with { probability(TE) theory(TE)} of the topic $t_{2,KMUTT}$ by two cases : *i*) the word 'probability(TE)'

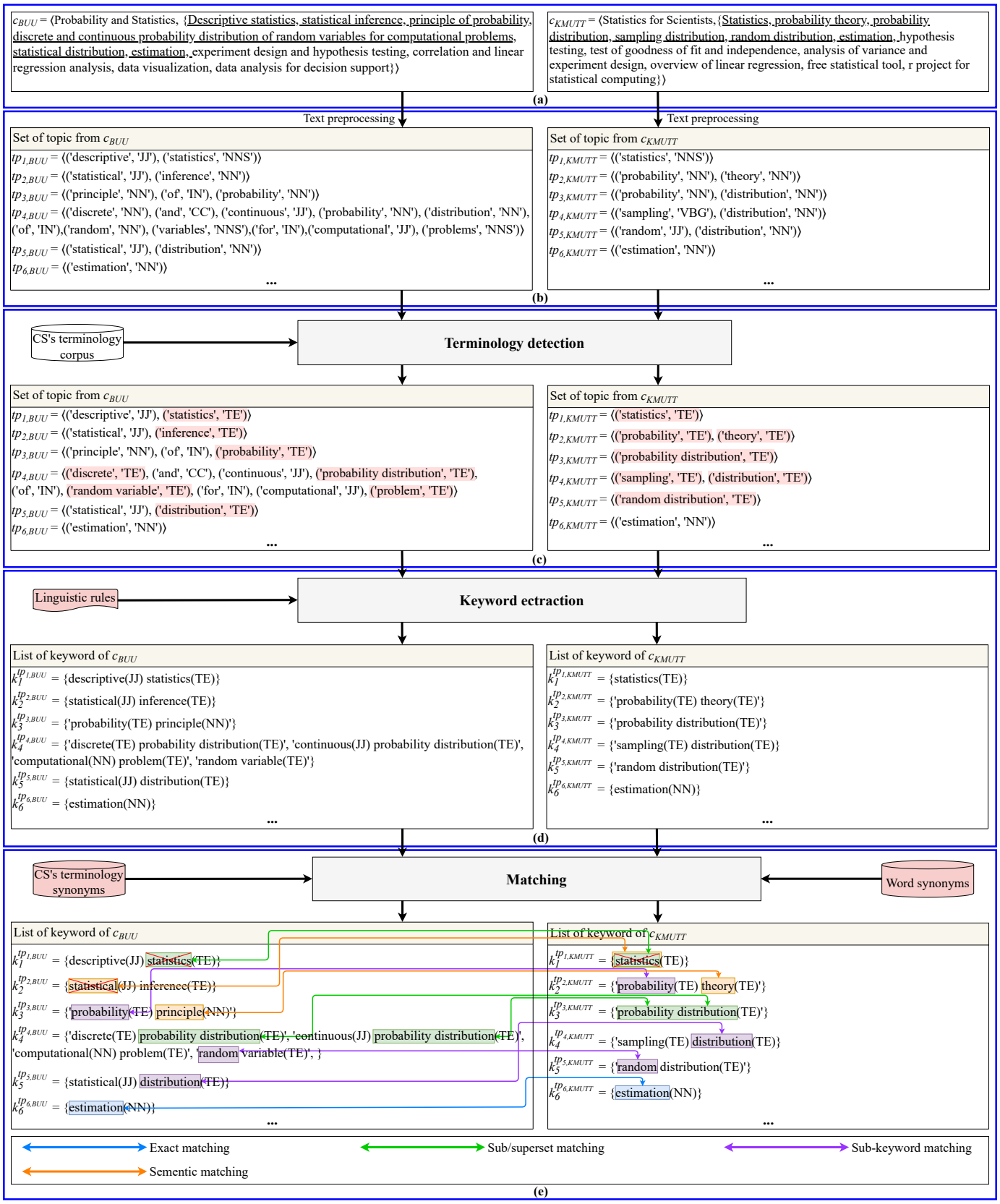

Fig. 2: Example of *eCSCDA* system on considering contents of "Probability and statistics" from BUU and KMUTT

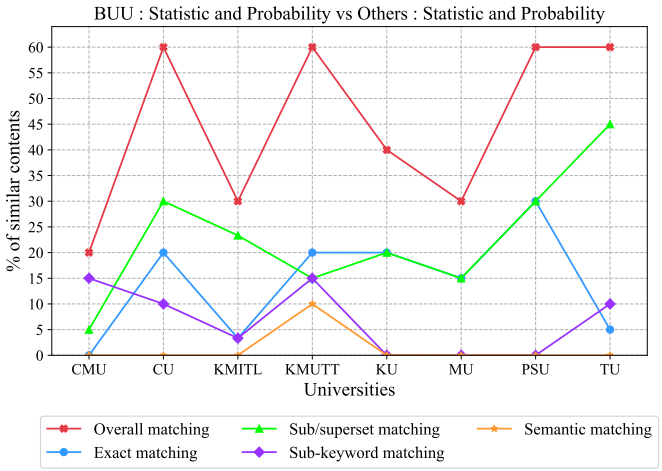

Fig. 3: Percentage of similar contents of *eCSCDA* on "Probability and Statistics" course of BUU against that of others

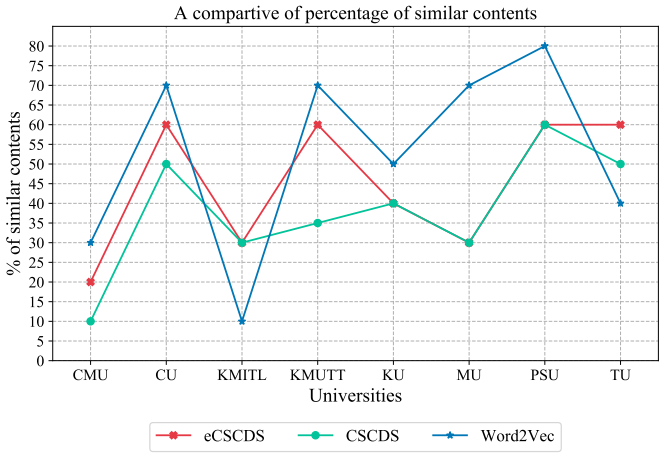

Fig. 4: Percentage of similar contents of *eCSCDA* against *CSCDA,* and *Word2Vec* on "Probability and Statistics"

matched by sub-keyword matching and *ii* the word 'principle(NN)' matched with 'theory(TE)' by semantic matching. With these matches, it can be concluded that the topic $t_{3,BUU}$ and $t_{2,KMUTT}$ is similar. Last, after matching all topics, the percentage of similar contents of BUU and that of KMUTT is calculated as the number of labeled topics of BUU divided by total number topic of BUU, computed as $\frac{4}{10} = 0.4$ (40%). On the other hand, the percentage of similar contents of KMUTT and BUU is the number of labeled topics of KMUTT divided by total number topic of KMUTT, calculated as $\frac{5}{12} = 0.417$ (41.7%), respectively.

IV. Experimental Results

Experiments were conducted on 550 CS course description collected (only English part) from 9 Thai universities having CS curriculum (*i.e.* Burapha University (BUU) 66 courses, Chiang Mai University (CMU) 63 courses, Chulalongkorn University (CU) 47 courses, King Mongkut's Institute of Technology Ladkrabang (KMITL) 87 courses, King Mongkut's University of Technology Thonburi (KMUTT) 40 courses,

TABLE I: Percentage of similar contents of *eCSCDA* against *CSCDA,* and *Word2Vec* on all courses of BUU

BUU(46) vs.	Percentage of similar contents		
	eCSCDA	*eCSCDA*	*Word2Vec*
CMU (15)	**34.82**	31.09	41.28
CU (16)	**28.45**	21.97	48.67
KMITL (19)	**32.10**	25.27	48.10
KMUTT (16)	**43.06**	34.87	46.91
KU (20)	**37.94**	27.94	47.94
MU (11)	**32.27**	34.86	47.46
PSU (22)	**35.38**	26.68	51.41
TU (21)	**32.27**	23.35	38.44
Avg	**34.54**	28.25	46.28

Kasetsart University (KU) 69 courses, Mahidol University (MU) 35 courses, Prince of Songkla University (PSU) 60 courses, and Thammasat University (TU) 85 courses).

In the experiments, one teaching description must be assigned as an *initial course description* and another one (or a group of ones) is set to be a *comparable course description*. Thus, a teaching description of a course from BUU is regarded as *an initial course description* and then compared with ones (on the same course) belonging to other universities. Four measures are applied to investigate the efficiency of the *eCSCDA* system in the term of number of detection and accuracy of matching similar/dissimilar contents defined as *i*) percentage of similar contents (*per_sim* calculated as in Eq. 1), *ii*) $precision = \frac{TP}{TP+FP}$, *iii*) $recall = \frac{TP}{TP+FN}$ and *iv*) $F\text{-}measure = 2 \times \frac{precision \times recall}{precision+recall}$, where TP is the number of correct matching on topics of descriptions given by the system, FP is the number of wrong matching, and FN is the number of mismatching. Last, a comparative study is conducted by comparing the *eCSCDA* system with other related systems *i.e. CSCDA* and *Word2Vec*, respectively.

Fig. 3 shows the percentage of similar contents (*per_sim*) of the "Probability and Statistics" course of BUU in comparison with that of other universities. As shown in the red line, the value of *per_sim* of comparing the course of BUU with another one is between 20 and 60% of the total number of contents of BUU (and $\approx$ 45% on average). This can be inferred that all universities have different perspectives and focuses on the teaching contents. Moreover, the figure also indicates the level of similar contents by the value of *per_sim* identified by each matching technique. With this, there are some contents that are described by using the same words (as shown in the blue line, the *per_sim* identified by exact matching which is $\approx$ 14 on average). Meanwhile, the most similar contents are just related to each other as shown by the value of *per_sim* recognized by subset, sub-keyword, and semantic matching. The average of the summation of *per_sim* identified by these three matching techniques is $\approx$ 31% of $\approx$ 45%. This can let us know that the teaching contents are pretty the same but it might be different on writing style or else which can be further analyzed. Meanwhile, Fig. 4 illustrates the comparison

TABLE II: Precision, recall, and F-measure of *eCSCDA* against *CSCDA,* and *Word2Vec* on all courses of BUU

BUU(46) vs.	Precision			Recall			F-measure		
	eCDCDA	*CSCDA*	*Word2Vec*	*eCDCDA*	*CSCDA*	*Word2Vec*	*eCDCDA*	*CSCDA*	*Word2Vec*
CMU (15)	**0.97**	0.97	0.38	**1.00**	0.98	0.46	**0.98**	0.96	0.36
CU (16)	**0.80**	0.72	0.38	**0.83**	0.79	0.56	**0.81**	0.71	0.41
KMITL (19)	**0.82**	0.80	0.45	**0.91**	0.77	0.57	**0.85**	0.78	0.46
KMUTT (16)	**0.88**	0.78	0.33	**0.95**	0.83	0.40	**0.91**	0.79	0.33
KU (20)	**0.91**	0.85	0.32	**0.95**	0.88	0.51	**0.93**	0.86	0.35
MU (11)	**0.86**	0.65	0.40	**0.94**	0.91	0.40	**0.89**	0.75	0.36
PSU (22)	**0.95**	0.87	0.41	**1.00**	0.98	0.53	**0.97**	0.91	0.44
TU (21)	**0.81**	0.77	0.51	**0.85**	0.79	0.50	**0.83**	0.77	0.47
Avg	**0.88**	0.80	0.40	**0.93**	0.87	0.49	**0.90**	0.82	0.40

of the percentage of similar contents calculated from *eCSCDA*, *CSCDA* and *Word2Vec*, respectively. It is shown that for some cases *eCSCDA* has the same percentage of similar contents as *CSCDA* but there are some that *eCSCDA* can give higher. The reason is that in some cases the teaching contents are similar only by using the same words or being subset (or superset) of each others. On the other hand, it is also shown that *eCSCDA* is better than *Word2Vec* in some cases but it gives higher precision, recall and F-measure for all cases.

Next, Table I shows a comparative study on the percentage of similar contents calculated by the three methods. From 66 courses from BUU, there are only 46 courses teaches by other universities where 15 of 46 are identical with CMU, 16 with CU, 19 with KMITL, 16 with KMUTT, 20 with KU, 11 with MU, 22 with PSU and 21 with TU, respectively. When looking at the results, it can be seen that teaching courses of BUU are mostly similar to that of KMUTT ($\approx 43\%$ on average) and then follow by MU ($\approx 37\%$), PSU ($\approx 34\%$), and so on. On the other hand, BUU has least similar contents to CU ($\approx 28\%$) which can let us know that both universities have a lot of different contents. It is then can be deeply analyzed for causation of these differences such as different focuses and/or writing styles, lack of updates, etc. Moreover, with the looking at efficiency, it can be seen that our *eCSCDA* can give a higher percentage of similar contents than *CSCDA* $\approx 6\%$ on average but it can give less than *Word2Vec* $\approx 12\%$ on average. This can express that *eCSCDA* outperforms *CSCDA*. However, even though *eCSCDA* give less number of similar contents *Word2Vec* than but its can give the highest values on precision, recall, and F-measure in comparison with the others. With all the results, it can be concluded that *eCSCDA* can efficiently match similar contents hidden in the course descriptions. Thanks to the terminology and word synonyms corpus with the new matching techniques that can give more similar contents.

V. Conclusion

In this paper, a new system, called *eCSCDA* (*efficient Computer Science Course Description Analysis* system) is introduced to improve the performance of *CSCDA* system for analyzing the course descriptions of Computer Science courses. In the new system, new linguistic rules and an efficient keyword extraction technique are applied to precisely identify important contents (*i.e.* keywords and/or terminologies) from a course description. Moreover, two corpuses, terminology and word synonyms, are settled and collected. Then, two matching methods, semantic and sub-keyword matching, based on the new corpuses are designed and applied to improve the task of matching similar contents occurring in course descriptions. From the experiments on 550 Computer Science course descriptions, the results show that the new improved *eCSCDA* outperforms the previous related systems (*i.e. CSCDA* and *Word2Vec*) in the terms of precision, recall, F-measure and percentage of similar content matching, respectively.

References

[1] C. Nuntawong, C. S. Namahoot, and M. Brückner, "A semantic similarity assessment tool for computer science subjects using extended wu & palmer's algorithm and ontology," in *Information Science and Applications*, 2015, pp. 989–996.

[2] C. S. N. Chayan Nuntawong and M. Brückner, "Home: Hybrid ontology mapping evaluation tool for computer science curricula," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 9, no. 2-3, pp. 61 – 65, 2017.

[3] C. Nuntawong, C. S. Namahoot, and M. Brückner, "A web based cooperation tool for evaluating standardized curricula using ontology mapping," in *Cooperative Design, Visualization, and Engineering*, 2016, pp. 172–180.

[4] C. W. Starr, B. Manaris, and R. H. Stalvey, "Bloom's taxonomy revisited: Specifying assessable learning objectives in computer science," *SIGCSE Bull.*, vol. 40, no. 1, p. 261–265, 2008.

[5] S. Masapanta-Carrión and J. A. Velázquez-Iturbide, "A systematic review of the use of bloom's taxonomy in computer science education," in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 2018, p. 441–446.

[6] A. Pawar and V. Mago, "Similarity between learning outcomes from course objectives using semantic analysis, blooms taxonomy and corpus statistics," *ArXiv*, vol. abs/1804.06333, 2018.

[7] G. O. M. O. N. P. V. Saquicela, F. Baculima and M. Espinoza, "Similarity detection among academic contents through semantic technologies and text mining," in *Proceedings INFOBAE Cuba*, 2018, pp. 1–12.

[8] P. Kamlangpuech and K. Amphawan, "A new system for analyzing contents of computer science courses," in *2020 7th International Conference on Advance Informatics: Concepts, Theory and Applications (ICAICTA)*, 2020, pp. 1–6.

[9] X. Rong, "word2vec parameter learning explained," 2016.

[10] J. Wang and Y. Dong, "Measurement of text similarity: a survey," *Information*, vol. 11, no. 9, p. 421, 2020.

[11] B. Chaisoongnoen, K. Amphawan, and A. Bunpeng, "Supplementary book suggestion for computer science courses," in *2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA)*, 2018, pp. 84–90.

[12] B. Chaisoongnoen and K. Amphawan, "An improvement of supplementary book suggestion system," in *The 9th International Conference on Smart Media and Applications (SMA 2020)*, 2020.