



ARTICLE

Prerequisite Relations among Knowledge Units: A Case Study of Computer Science Domain

Fatema Nafa^{1,*}, Amal Babour² and Austin Melton³

¹Department of Computer Science, Salem State University, Salem, MA 01970, USA

²Information Systems Department, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, 21589, Saudi Arabia

³Department of Computer Science, Kent State University, Kent, OH 44240, USA

*Corresponding Author: Fatema Nafa. Email: fnafa@salemstate.edu

Received: 03 November 2021 Accepted: 14 March 2022

ABSTRACT

The importance of prerequisites for education has recently become a promising research direction. This work proposes a statistical model for measuring dependencies in learning resources between knowledge units. Instructors are expected to present knowledge units in a semantically well-organized manner to facilitate students' understanding of the material. The proposed model reveals how inner concepts of a knowledge unit are dependent on each other and on concepts not in the knowledge unit. To help understand the complexity of the inner concepts themselves, WordNet is included as an external knowledge base in this model. The goal is to develop a model that will enable instructors to evaluate whether or not a learning regime has hidden relationships which might hinder students' ability to understand the material. The evaluation, employing three textbooks, shows that the proposed model succeeds in discovering hidden relationships among knowledge units in learning resources and in exposing the knowledge gaps in some knowledge units.

KEYWORDS

Knowledge graph; text mining; knowledge unit; graph mining

1 Introduction

A textbook is one of the most fundamental resources in which a learner obtains knowledge about the external world. Learning is considered as a building block where new knowledge often relies on the understanding of other existing knowledge. Textbooks are the primary channels for delivering knowledge to learners. Each textbook is written in a certain way to present its content. However, some textbooks fail to present its content in a clear interconnected smooth way which may make them classified as ineffective. Although some textbooks may cover all the needed concepts about a specific topic, they may not be well-written, making the concepts more difficult to comprehend. For example, consider a group of first graders getting their first mathematics lesson. If the learner chooses to start with "fractions" without knowing the subtraction, addition, and multiplication, the learner will be unable to understand the lesson itself. The text's form is illustrated by the quality of content



knowledge. Not only does a very rich knowledge textbook help in understanding the concepts, but also knowledge association within the textbook is important in such a manner to sequentially build up the knowledge base for further instruction.

Important questions include the following: What are the domain concepts that a learner needs to learn? What should he/she start with to build knowledge? Keys to answering these questions include understanding what a lexical relation is, where a lexical relation occurs, and when related concept pairs join together that can extend to become sentences. Identifying lexical relations among text concepts is considered a way to help in understanding the structure among the concepts and building knowledge. An important difference in the textbook is the amount of knowledge. Some textbooks may contain all the needed knowledge; others may assume knowledge known from previous knowledge.

Using a lexical database is an open research problem [1]. In order to ease the cognitive burden for the learner, this study proposes using an English WordNet as a lexical database to be the previous known knowledge and fill in missing knowledge in a textbook. Because the current English WordNet does not include computer science concepts, the current WordNet has been expanded to include computer science-specific concepts. Then, it is utilized to add the lexical relationships among the concepts and increase the connectivity among them. There are three parts to the main contribution of this paper: extraction of prerequisites relations between concepts; measuring the dependencies between extracted relations using the statistical model and extending the English WordNet to include most computer science concepts.

The rest of the paper is organized as follows. The related work is presented in [Section 2](#). [Section 3](#) references the model terminologies used in this article. [Section 4](#) provides information about enhancing knowledge graphs from a lexical database as well as describing the technical steps in detail. [Section 5](#) presents an example study. Lexical databases for enhancing knowledge are given in [Section 6](#). The experiment steps of the model are explained in [Section 7](#). [Section 8](#) concludes the paper and adds possible future work.

2 Related Work

To date, several kinds of relations such as semantic relations, grammatical relations, and negation have been extracted to illustrate the relation among concepts in different research areas, different domains and different purposes. Given the relation extraction and the methods used are very broad areas, the inclusion criteria must be explicitly explained. In the biomedical domain, Yao et al. [2] surveyed most of the biological relation extraction methods. In the area of information science domain, Allahyari et al. [3] investigated the current tools and algorithms used to extract the entity for both structured and unstructured text data. In the natural language processing (NLP) domain, Rim et al. [4] described the relation extraction methods and classification tasks on scientific papers. Authors of [5] pointed out the importance of relation extraction in the social sciences domain which extracted labels that describe relations between entities in social networks. Thus, in all scientific and academic fields, easy accessibility of relation extraction and domain knowledge have become the most beneficial goals.

Many existing studies have investigated different methods to extract different types of relations. Those methods can be classified as follows (but not limited to): clustering-based, classification-based, template-based, and ontology-based methods. The clustering-based approach uses hierarchical

clustering models to extract taxonomic relations [6,7]. Authors of [8,9] surveyed a comprehensive review of the most important supervised and semi-supervised classification methods. In the template-based methods, a standard algorithm is used with predefined template schemas to extract the relation [10]. Several research works also addressed the use of a lexical database such as WordNet to improve the accuracy of relation extraction. Authors of in [11] explored the possibility of enriching the content of existing ontologies. Authors of [12] indicated that the development of a lexical database is able to check the similarity of data records and detect the correct data region with higher precision. Also, WordNet is used to extract relations among free text concepts [13].

Regarding relation extraction in the area of education, researchers have investigated a specific type of relation named prerequisite relation [14–16]. Authors of [17] focused on mining the learning-dependency between knowledge units in textbooks to specify the importance of the structure and content of the textbook for learners. Work also measured the quality of the textbook [18] using one of the data mining models. Authors of [19] extracted knowledge from textbooks by harvesting geometry axioms for the mathematical domain. Cohen et al. [20] predicted the prerequisite structures in Wikipedia pages using a random walk method. The extraction of prerequisite relation which investigated the use of information visualization models for better understanding characteristics in textbooks was described in [21]. For enhancing the prerequisite relations between educational concepts extracted from a computer science textbook specific tool, a knowledge graph was built [22].

This work focuses on more general semantic relations among concepts and knowledge units, as well as measuring the knowledge in the textbook. The ontology-based method is proposed to enhance the extracted knowledge along with designing a specific domain (a CS-WordNet).

3 Terminologies

A Semantic Knowledge Graph (SKG) is a graph that captures the current state of learning knowledge. It shows textbook concepts and associations among the concepts. The associations can be found by reading the textbook. A directed graph presents $SKG = (C, E)$, where C is a set of the textbook concepts and E is a set of edges among the concepts. Each edge refers to a syntactic relation representing a piece of knowledge between a pair of concepts extracted from the textbook. A set of concepts can be connected as a subcomponent to form a Knowledge Unit.

A Knowledge Unit (KU) is a framework for identifying necessary knowledge concepts for a sub-topic in a specific area. It is a way to organize a knowledge base of content resources such as text, audio, video, or graphics. SKG may contain too many knowledge units. As shown in Fig. 2 for example, in the domain of computer science, consider t is a topic that has three sub-topics KU, where $t = \text{data structure}$, $KU_1 = \text{tree}$, $KU_2 = \text{stack}$, and $KU_3 = \text{list}$. To reach a good state of understanding, all KU should be connected by at least one connection between each two KU's. However, in some cases, a KU may not have connections to other KUs, thus rendering it totally isolated and making it difficult to understand. However, if the KU has connections to another knowledge unit, this will form a knowledge component and be simply understood by the learner. Fig. 1 illustrates an example of an SKG with three knowledge units: knowledge unit (KU_1), knowledge unit (KU_2), and knowledge unit (KU_3).

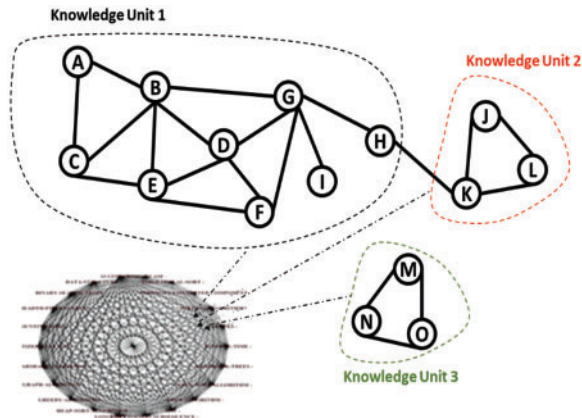


Figure 1: Semantic knowledge graph with three knowledge unit

This work focuses on more general semantic relations among concepts and knowledge units, as well as measuring the knowledge in the textbook. This study proposes using the ontology-based method to enhance the extracted knowledge, along with designing a specific domain (CS-WordNet).

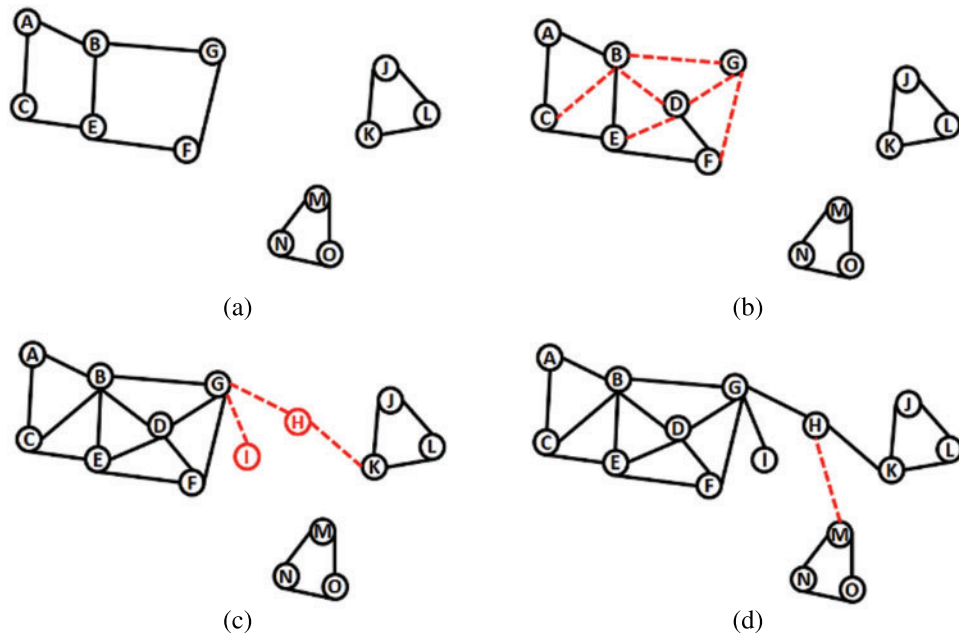


Figure 2: Knowledge among SKG concepts. (a) SKG_0 in the initial learning time. (b–d) Knowledge among SKG_1 , SKG_2 , and SKG_3 in other learning times

4 Semantic Knowledge Graph Measuring

By auto reading the textbook, a Semantic Knowledge Graph (SKG) is generated. The proposed model uses a model by Newman [23], where the Newman model is used to evaluate the state of the knowledge in SKG at a specific learning time. By a derivative of the model, the knowledge in SKG may increase or remain stable. The knowledge for an SKG can be measured using Eq. (1). It takes a value

between 0 and 1, where 0 means that there is no knowledge in the SKG, while 1 means that SKG is full of knowledge. In other words, each concept in SKG can be reached from any other concepts, where m is the number of unit knowledge in SKG and p^i is the probability of concepts having i connections in a specific knowledge unit. It can be determined by Eq. (2), while ck represents the actual degree of concept c .

$$SKG_0(C) = \sum_{j=1}^m \sum_{i,k=1}^n p^i c^k \tag{1}$$

$$p^i = \frac{\text{number of concepts has degree } i}{\text{total numbe of concepts in KU}} \tag{2}$$

As an example, p^1 means the number of concepts in the knowledge unit has a degree of 1 divided by the total number of concepts in the knowledge unit, p^2 means the number of concepts in the knowledge unit has a degree of 2 divided by the total number of concepts in the knowledge unit, and so on. The amount of knowledge in each knowledge unit in SKG can be calculated by Eq. (3), where SKG_0 is the derivative of SKG_0 and SKG_0'' is the derivative of SKG_0' [6–9].

$$KUj = 1 + \frac{(SKG_0'(1))^2}{SKG_0'(1) - SKG_0''(1)} \tag{3}$$

4.1 Text Layout Gain Component Evaluation

It can be thought of as a probability that knowledge learned at a specific time from the SKG has been formed and joined to other knowledge to form a giant knowledge unit. Assuming that all the knowledge units in SKG forms a gain component, the state of the knowledge in SKG can then be measured by Eq. (4) which is reproduced in the context of this paper [23,24].

$$GKU = SKG_0(1) + \frac{SKG_0'(1) SKG_0(1)}{1 - SKG_0'(1)} \tag{4}$$

4.2 Non-Gain Component Evaluation

Assuming that the knowledge units in SKG forms a non-gain component, the state of the knowledge in SKG can then be measured by Eq. (5) which is reproduced in the context of this paper [24,25].

$$NGKU = \frac{GKU}{SKG_0(1)} \tag{5}$$

5 Example Study

Consider that SKG shown in Fig. 2 represents topic t and KU_1 represents a sub-topic. To calculate the amount of knowledge in KU_1 , where the number of concepts $n = 9$, first, Eq. (1) is applied to create the mathematical function of SKG as follows. $SKG_0(C) = p^5 c^5 + p^4 c^4 + p^3 c^3 + p^2 c^2 + p^1 c^1$. To calculate the value of p^1, p^2, \dots, p^5 , Eq. (2) is applied as follows:

$$p^1 = \frac{\text{number of concept have degree 1}}{\text{total number of concepts in KU1}} = \frac{2}{9} = 0.22$$

$$p^2 = \frac{\text{number of concepts have degree 2}}{\text{total number of concepts in KU1}} = \frac{1}{9} = 0.11$$

$$p^3 = \frac{\text{number of concepts have degree 3}}{\text{total number of concepts in KU1}} = \frac{3}{9} = 0.33$$

$$p^4 = \frac{\text{number of concept have degree 4}}{\text{total number of concepts in KU1}} = \frac{2}{9} = 0.22$$

$$p^5 = \frac{\text{number of concept have degree 5}}{\text{total number of concepts in KU1}} = \frac{2}{9} = 0.22$$

The mathematical function for SKG_0 can be found by substituting the obtained values in Eq. (1).

$$SKG_0(C) = 0.22 c^5 + 0.22 c^4 + 0.33 c^3 + 0.11 c^2 + 0.22 c^1$$

The derivative of $SKG_0(C)$ is:

$$SKG'_0(C) = (5 * 0.22) c^4 + (4 * 0.22) c^3 + (3 * 0.33) c^2 + (2 * 0.11) c^1 + (1 * 0.22)$$

$$SKG'_0(C) = 1.1 c^4 + 0.88 c^3 + 0.99 c^2 + 0.22 c^1 + 0.22$$

The derivative of $SKG'_0(C)$ is:

$$SKG''_0(C) = (4 * 1.1) c^3 + (3 * 0.88) c^2 + (2 * 0.99) c^1 + (0.22)$$

$$SKG''_0(C) = 4.4 c^3 + 7.04 c^2 + 1.98 c^1 + 0.22$$

By considering $C = 1$, the result of $SKG_0 = 1.1$, $SKG'_0 = 3.32$, and $SKG''_0 = 9.24$, where SKG'_0 and SKG''_0 represent SKG_1 and SKG_2 , respectively. By substituting the values in Eq. (3), the amount of knowledge in SKG is as follows: $KU = 1 + \frac{(3.32)2}{3.32 - 9.24} = 1 + \frac{11}{5.92} = 0.85$. For calculating the amount of knowledge in KU_1 , assuming that KU_1 is a giant component, then by substituting the values in Eq. (4), the amount of knowledge that can be gained from KU_1 is as follows: $GKU = 1.1 + \frac{3.32 * 1.1}{1 - 3.32} = 0.45$. Assuming that KU_1 is not a giant component, then by substituting the values in Eq. (5), the amount of knowledge that can be gained from KU_1 is as follows: $NGKU = \frac{0.45}{1.1} = 0.40$. The result shows that the amount of knowledge in SKG which represents topic t (0.85) can be considered good and it can be used as a good reference for learners. Whereas in the assumption that KU_1 is a giant component, the amount of knowledge (0.45) in KU_1 representing sub-topic s is not good enough for the learners and more knowledge needs to be added using other resources. Also, in the assumption that KU_1 is a non-giant component, the amount of knowledge (0.40) refers to the basic amount of knowledge that can be given to the learner about sub-topic s .

6 Enhancing Knowledge Using a Lexical Database

One of the possible ways to increase knowledge in knowledge units is by adding lexical relations among SKG concepts using a lexical database.

6.1 WordNet

WordNet is a reliable a lexical database that has been used as a source of knowledge in different areas. It can be defined as a lexical database of English words including nouns, verbs, adjectives and adverbs that are grouped into a set of synsets, each expressing a unique concept. Synsets are linked using conceptual-semantics [26]. WordNet's structure makes it a useful tool for computational linguistics and natural language processing tasks [26,27]. In WordNet, word forms can be connected

to each other through a variety of relations such as antonymy (e.g., Good to Bad and vice-versa). Word-meaning nodes are also connected by relations like hypernymy (Car is a Vehicle) and meronymy (Car has an Engine). Although these relations such as hypernymy and meronymy are directed, they can be directed both ways depending on what relation is formed. For example, the connection between CAR and ENGINE can be from CAR to ENGINE, because the CAR has ENGINE but also from ENGINE to CAR because ENGINE is a part of a CAR. Because there are no inherently preferred directions for these relations, WordNet is treated as an undirected graph.

It includes the following relation types [26]: *Synonymy*: It is considered a fundamental relation because WordNet uses sets of synonyms called Synsets to represent multiple senses. *Antonymy*: It is an association of opposing words. It is symmetric between word forms. *Hyponymy*: It is a sub-relation, and its inverse is hypernymy or super-name. It is a transitive relation between synsets. It is a lexical relation because it relates one synset to another. *Meronymy*: It is a part of a relation, and its inverse is Holonymy, known as a whole name. *Troponymy*: It is a kind-of type relation.

This paper particularly focuses on a WordNet ontology-based approach for finding hidden semantic/lexical relations among the concepts in the semantic knowledge graph.

6.2 Computer Science WordNet

When a WordNet is used, lexical connections might be added among the concepts. However, in the SKG, not all domain concepts become connected. In this paper, the domain is computer science concepts due to the WordNet limitations.

There has been related work in the extending of English WordNet for CS majors such as: Computer Memory domain area [28] and extensions in the domain of Psychology and Computer Science [29]. Different models have been used in previous studies in the construction of Computer Science CS ontologies, including the seven-step method [29], the model method [30], the skeleton method [31], and the enterprise modeling method [32,33]. All those methods are semi-automatic and are not available online. In order to efficiently and effectively use CS WordNet, this study proposes using the same relations that have been used in the English WordNet in the process of the addition of CS concepts. Fig. 3 illustrates the steps of a WordNet extension with CS concepts. For example, consider concepts from the Computer Sciences domain related to Algorithm topic as follows: Tree and Binary-tree. If the general ontology to add a relation between those concepts was reviewed, there would be no connection between them, but in CS WordNet there is a connection between those concepts where Tree is a Hyponymy to Binary-tree.

6.2.1 Choosing of CS Textbook

Algorithm and Data Structure textbooks [34–36] have been chosen as CS textbooks. Automatic extraction of CS keywords (concepts) and the semantic relationship are among them using NLP. In this step, two main parts have been completed.

6.2.2 Keyword Extractions

In this part, a list of concepts from a textbook content are obtained, after which the obtained list is stored in an Excel file. The concepts are used to retrieve all the synsets in any part of speech. Using the concept level, which here is known as either a low-level concept or a high-level concept, the categorization of these levels is based on calculating the likelihood ratio for each concept. After the likelihood ratio has been taken for each concept, the concepts that have a higher value are meant to

be more related to the CS domain. Each concept is represented with a synset, where each part of the synset can be only a noun as a Part of Speech (POS).

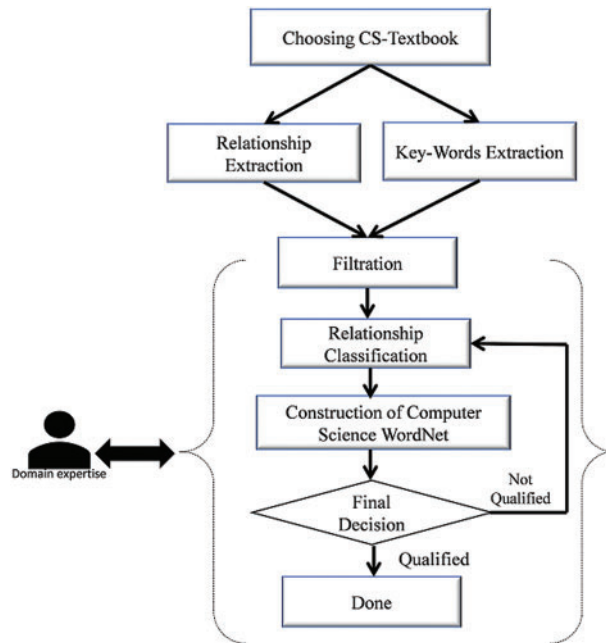


Figure 3: WordNet computer science construction methodology

6.2.3 Relationship Extraction

In this part the main relationships extracted are the WordNet relationships between CS concepts, where they are the WordNet relationships between any two concepts C_i and C_j . For the CS WordNet structure, each concept is represented with a synset, each item of which can be only a noun as a part of speech (POS). The sense is based on the precise meaning of the concept. The classification of the senses is based on using a free online dictionary of Algorithms and Data Structures from the National Institute of Standards and Technology (NIST). This paper will go through examples for the concepts that exist and the concepts that do not exist in English WordNet and will show how the relationships and the sense of the CS concepts are classified throughout.

6.2.4 Filtration

The filtration step involves the removal of common words like “a”, “of”, or any other common words not related to the CS domain.

6.2.5 Relationship Classification

In this step, domain experts judge the extracted relationships to decide the relevant and non-relevant relationships, the results of which are saved in an Excel file to facilitate future work.

6.2.6 Relationship Classification

This step used more technical details to build the CS concept senses. The sense of the CS concept is based on the precise meaning of the concept, as concepts usually have several senses. The task of

selecting which concept sense most accurately represents the sense of a particular use of a concept in the linguistic research is known as Word Sense Disambiguation (WSD) [37]. A base algorithm was used as Lesk's algorithm for WSD. This algorithm has a major performance problem, as the dictionary used may not include sufficient vocabulary to identify related senses. This study used ontology to look up the concepts and the gloss of those concepts instead of the dictionary. The gloss is a short description that makes it easier to distinguish one particular sense from other similar senses for the synonyms in the set. In the process of determining CS senses for the concepts, a full detail for each sense of each concept was prepared by obtaining the gloss and the relationships (Hypernyms, Hyponyms, and Menonym). The Lesk algorithm was slightly modified to create baseline algorithm. In the extension of the CS WordNet, different types of senses for CS concepts were found. These can be classified as a semantic sense (SS) and a geometry sense (GS), where a semantic sense represents the semantic meaning of the concepts which are not related to the CS meaning. The geometric sense represents the meaning of CS concepts based on different CS areas. Fig. 4 shows an example of semantic and geometric senses: $\{S_1 = \text{general}\}$, $\{S_2 = \text{CS/Data Structure}\}$, $\{S_3 = \text{CS/Operating System}\}$, and $\{S_4 = \text{CS/Algorithm}\}$, where $S_{1,1}$ refers to the first sense of the first concept, $S_{2,1}$ the first sense of the second concept and so on. Each concept may be associated with one or more synsets. This will lead to ambiguity in analyzing the content and each concept in the synset can be associated with five parts of speech. Each part of speech POS is associated with many senses. It is important to note that this work focused only on the noun concepts and not the other POS.

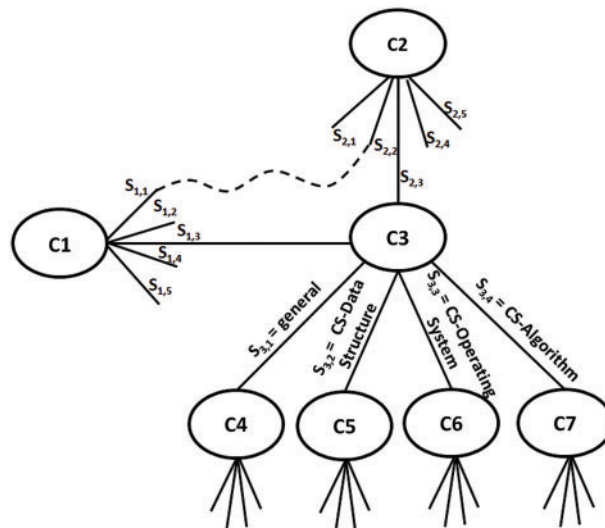


Figure 4: An example of semantic and geometric senses

7 Experiment

7.1 Content Material

An experiment to evaluate the amount of knowledge in knowledge units was conducted on three highly adapted textbooks TXT_i used in Computer Science classes at many universities. TXT_1 is “Introduction to Algorithms” [34], TXT_2 is “Data Structures and Algorithms” [35] and TXT_3 is “Algorithms” [36], respectively. The applied lexical database is WordNet version 3.0, in addition to the created CS-WordNet.

7.2 Results

This section presents the information about the created CS-WordNet and the analysis of the amount of knowledge gained in each of the three books before and after using the lexical database. [Table 1](#) shows the number of concepts and a breakdown of the total number of relations in each of the three textbooks before and after using the lexical databases (WordNet, CS-WordNet). As can be seen, using WordNet and CS-WordNet contribute to adding lexical relations to the syntactical relations among the textbooks concepts which may help in increasing the amount of knowledge among the knowledge units.

Table 1: Total number of concepts and lexical relations in the three textbooks

	TXT ₁	TXT ₂	TXT ₃
Number of concepts	2384	769	5088
Number of syntactic relations from the textbook	3886	3154	4085
Number of lexical relations using WordNet	150	115	95
Number of lexical relations using CS-WordNet	4250	4100	3000
Total number of relations	8286	7369	7180

A breakdown of the total number of the lexical relations among the concepts in the created CS-WordNet in each of the three books is shown in [Table 2](#).

Table 2: Breakdown of the created lexical relations CS WordNet in the three textbooks

	TXT ₁	TXT ₂	TXT ₃
Total number of relations	4250	4100	3000
Synonymy	1300	1815	2123
Hyponymy	1716	1314	727
Troponymy	1234	971	150

[Fig. 5](#) shows the analysis of the amount of knowledge gained from the three textbooks, considering the found relations from the textbooks, using WordNet and CS-WordNet, respectfully. In the experiment, this study considers the learning time as reading the whole textbook. As shown, the amount of knowledge in TXT₁ is 85%; this can be considered a good amount of knowledge. By considering the graph as a giant component to analyze the knowledge inside each knowledge unit, the amount of knowledge is 75%, which may still be considered good. While considering the graph as a non-giant component, the amount of knowledge is 3%, which is considered as ineffective. Then, after using WordNet, the amount of knowledge stayed stable at 85%, while after considering the graph as a giant component, the amount of knowledge increased from 75% to 80%. Thus, even the amount of the knowledge increased but the non-giant component remained at 3%. Then, after using CS-WordNet, the connectivity among knowledge units increased from 80% to 98%; this made those knowledge units ready with rough knowledge to be introduced as an introduction topic or an advanced topic. Thus, decreasing the size of the non-giant component to be 2%. 75% of those knowledge units could be used as an introduction topic, rather than an advanced one. This resulted in increasing the size of the non-giant component to be 3%. Then the external reference was used to enhance the knowledge for the textbook, but at the level of the whole knowledge remained stable at 85%, whereas the connectivity

inside the knowledge units increased from 75% to 80%, as shown in the figure for Textbook1. It is clear that the external reference was used but not domain specific where a lot of concepts are still isolated. After that, CS WordNet was used. Then, connectivity among knowledge units increased from 80% to 98% which made those knowledge units ready with rough knowledge to be introduced as an introduction topic or an advanced topic. Thus, it led to decreasing the size of the non-giant component to be 2%. Textbook2 has a knowledge level of 78%, where the knowledge inside each knowledge unit is 85%; after that using the non-giant component measuring the isolation inside those knowledge units is 2%. Then an external reference was used to enhance the knowledge for Textbook2, which increased from 78% to 85%, as shown in Fig. 5. Then CS WordNet was used to increase the knowledge unit connectivity from 85% to 90%; this decreased the size of the non-giant component to be 1%. Textbook3 has a knowledge level of 70%, where the knowledge inside each knowledge unit is 70%; then using the non-giant component measure, the isolation inside those knowledge units is 3%. Then an external reference was used to enhance the knowledge for the textbook, which increased from 70% to 85%; also, the size of the non-giant component decreased to be 2%, as shown in the figure. Then, CS WordNet was used to increase the knowledge unit connectivity from 85% to 90%; this decreased the size of the non-giant component to be 1%. The connectivity inside the knowledge units increased from 80% to 98%, thus decreasing the size of the non-giant component to be 1%.

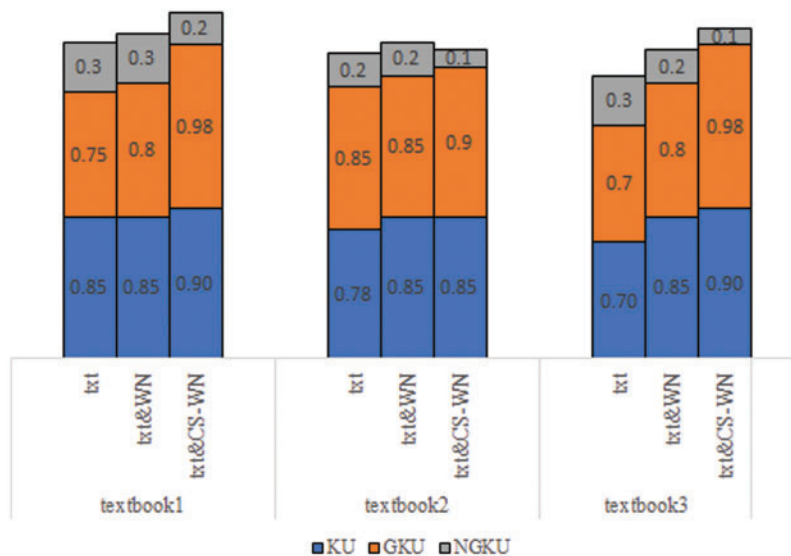


Figure 5: Scaling of the KU in the three textbooks

To evaluate the quality of the proposed model, the gold standard for any analysis is human judgment. The results of the proposed model were compared with ground truth (GT). 150 Computer Science undergraduate students from Kent State University were asked to read Chapter 6 (Heapsort Topic) in *Introduction to Algorithms* [34], extract the main concepts, and add syntactic relations among them from what they read to construct SKG manually. The results of the obtained SKG were scaled using Minmax Scaler 3 and compared with the results obtained from the proposed model. The evaluation results are summarized in Table 3. As shown, all the measured values were obtained by counting the TP, FN, FP, and TN. The accuracy accesses 86%, with recall 83% and precision 97% and F-measure 89%. According to these results, the performance of the model is significantly great.

Table 3: Evaluation results

TP = 57	FN = 12	FP = 2	TN = 28
Accuracy = 0.86	Recall = 0.83	Precision = 0.97	F-measure = 0.89

8 Conclusion

This paper presented a model which was used to analyze CS corps. The study identified the knowledge units of a textbook that are not well-written, which may affect both the quality of the knowledge acquired and the time needed to learn this knowledge unit. Adding a lexical database could enhance the quality of the learner knowledge as well as improve the knowledge base quality for the specific domain. At this point, it can be clearly stated that this model can fit any new textbook in the Computer Science domain, as well as those in other domains. For future research, researchers of this study will investigate the use of Computer Science ontologies (CSO) [38] and compare the result obtained by WordNet with this ontology. In addition, the use of the framework to evaluate online learning resources will be investigated and a free online tool will be designed to be tested and used by the other institutes.

Acknowledgement: Thank you for the editors of this journal and the anonymous reviewer.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Kurilovas, E., Juskeviciene, A. (2015). Creation of web 2.0 tools ontology to improve learning. *Computers in Human Behavior*, 51, 1380–1386. DOI 10.1016/j.chb.2014.10.026.
2. Yao, Y., Ye, D., Li, P., Han, X., Lin, Y. et al. (2019). DocRED: A large-scale document-level relation extraction dataset. *arXiv preprint arXiv:1906.06127*.
3. Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D. et al. (2017). A brief survey of text mining: Classification, clustering and extraction techniques. *arXiv preprint arXiv:1707.02919*.
4. Rim, K., Tu, J., Lynch, K., Pustejovsky, J. (2020). Reproducing neuralensemble classifier for semantic relation extraction in scientific papers. *Proceedings of the 12th Language Resources and Evaluation Conference*, Marseille, France.
5. Cai, D., Shao, Z., He, X., Yan, X., Han, J. (2005). Mining hidden community in heterogeneous social networks. *Proceedings of the 3rd International Workshop on Link Discovery*, Chicago, IL, USA.
6. Smirnova, A., Cudré-Mauroux, P. (2018). Relation extraction using distant supervision: A survey. *ACM Computing Surveys*, 51(5), 1–35. DOI 10.1145/3241741.
7. Riedel, S., Yao, L., McCallum, A., Marlin, B. M. (2013). Relation extraction with matrix factorization and universal schemas. *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, Georgia, USA.
8. Kumar, S. (2017). A survey of deep learning methods for relation extraction. *arXiv preprint arXiv:1705.03645*.

9. Pawar, S., Palshikar, G. K., Bhattacharyya, P. (2017). Relation extraction: A survey. *arXiv preprint arXiv:1712.05191*.
10. Romano, L., Kouylekov, M., Szpektor, I., Dagan, I., Lavelli, A. (2006). Investigating a generic paraphrase-based approach for relation extraction. *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy.
11. Schutz, A., Buitelaar, P. (2005). Relext: A tool for relation extraction from text in ontology extension. *International Semantic Web Conference*, Berlin, Heidelberg, Springer.
12. Boyce, S., Pahl, C. (2007). Developing domain ontologies for course content. *Journal of Educational Technology & Society*, 10(3), 275–288.
13. Hasan, A. M., Noor, N. M., Rassem, T. H., Noah, S. A. M., Hasan, A. M. (2020). A proposed method using the semantic similarity of WordNet 3.1 to handle the ambiguity to apply in social media text. *Information Science and Applications*, pp. 471–483. Singapore, Springer.
14. Pal, S., Arora, V., Goyal, P. (2020). Finding prerequisite relations between concepts using textbook. *arXiv preprint arXiv:2011.10337*.
15. Alzetta, C., Miaschi, A., Dell’Orletta, F., Koceva, F., Torre, I. (2020). PRELEARN@ EVALITA 2020: Overview of the prerequisite relation learning task for Italian. <http://ceur-ws.org/Vol-2765/paper164.pdf>.
16. Agrawal, R., Gollapudi, S., Kenthapadi, K., Srivastava, N., Velu, R. (2010). Enriching textbooks through data mining. *Proceedings of the First ACM Symposium on Computing for Development*, London, United Kingdom.
17. Agrawal, R., Gollapudi, S., Kannan, A., Kenthapadi, K. (2012). Data mining for improving textbooks. *ACM SIGKDD Explorations Newsletter*, 13(2), 7–19. DOI 10.1145/2207243.2207246.
18. Agrawal, R., Gollapudi, S., Kannan, A., Kenthapadi, K. (2011). Identifying enrichment candidates in textbooks. *Proceedings of the 20th International Conference Companion on World Wide Web*, Hyderabad, India.
19. Agrawal, R., Gollapudi, S., Kannan, A., Kenthapadi, K. (2011). Enriching textbooks with images. *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, Glasgow, Scotland, UK.
20. Han, K., Yang, P., Mishra, S., Diesner, J. (2020). WikiCSSH: Extracting computer science subject headings from wikipedia. *ADBIS, TPD and EDA 2020 Common Workshops and Doctoral Consortium*, Cham, Springer.
21. Yu, H., Li, H., Mao, D., Cai, Q. (2020). A relationship extraction method for domain knowledge graph construction. *World Wide Web*, 23(2), 735–753. DOI 10.1007/s11280-019-00765-y.
22. Stahr, M., Yu, X., Chen, H., Yan, R. (2020). Design and implementation knowledge graph for curriculum system in university. EasyChair Preprint. No. 4288.
23. Müller, B., Reinhardt, J., Strickland, M. T. (1995). *Neural networks: An introduction*. Springer Science & Business Media.
24. Li, M., Wang, B. H. (2015). Generating function technique in complex networks. *Paper Presented at the Journal of Physics: Conference Series*, IOP Publishing.
25. Dorogovtsev, S. N., Goltsev, A. V., Mendes, J. F. (2008). Critical phenomena in complex networks. *Reviews of Modern Physics*, 80(4), 1275. DOI 10.1103/RevModPhys.80.1275.
26. Miller, G. A. (1995). WordNet: A lexical database for English. *Communications of the ACM*, 38(11), 39–41. DOI 10.1145/219717.219748.
27. Keszocze, O., Soeken, M., Kuksa, E., Drechsler, R. (2013). Lips: An ide for model driven engineering based on natural language processing. *2013 1st International Workshop on Natural Language Analysis in Software Engineering (NaturaLiSE)*, San Francisco, CA, USA, IEEE.
28. Kremizis, A., Konstantinidi, I., Papadaki, M., Keramidas, G., Grigoriadou, M. (2007). Greek WordNet extension in the domain of Psychology and Computer Science. *Proceedings of the 8th Hellenic European*

- Research Computer Mathematics and its Applications Conference (HERCMA)*, Economical University, Athens, Citeseer.
29. Blitsas, P., Grigoriadou, M. (2011). Greek WordNet and its extension with terms of the computer science domain. *International Conference on Universal Access in Human-Computer Interaction*, Berlin, Heidelberg, Springer.
 30. López, M. F. (1996). *CHEMICALS: Ontología de elementos químicos*. Politechnische Universität von Madrid, Madrid.
 31. Uschold, M., King, M. (1995). *Towards a methodology for building ontologies*. Citeseer, Artificial Intelligence Applications Institute, University of Edinburgh.
 32. Siau, K. (2004). Informational and computational equivalence in comparing information modeling methods. *Journal of Database Management*, 15(1), 73–86. DOI 10.4018/JDM.
 33. Noy, N. F., McGuinness, D. L. (2001). Ontology development 101: A guide to creating your first ontology. *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880*.
 34. Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C. (2009). *Introduction to algorithms*. MIT Press and McGraw-Hill.
 35. Goodrich, M. T., Tamassia, R., Mount, D. M. (2011). *Data structures and algorithms in C++*. John Wiley & Sons.
 36. Sedgewick, R., Wayne, K. (2011). *Algorithms*. Addison-Wesley professional.
 37. Kumar, M., Mukherjee, P., Hendre, M., Godse, M., Chakraborty, B. (2020). Adapted lesk algorithm based word sense disambiguation using the context information. *International Journal of Advanced Computer Science and Applications*, 11(3). DOI 10.14569/IJACSA.2020.0110330.
 38. Salatino, A. A., Thanapalasingam, T., Mannocci, A., Birukou, A., Osborne, F. et al. (2020). The computer science ontology: A comprehensive automatically-generated taxonomy of research areas. *Data Intelligence*, 2(3), 379–416. DOI 10.1162/dint_a_00055.