Knowledge Graph based Learning Guidance for Cybersecurity Hands-on Labs

Yuli Deng, Duo Lu, Dijiang Huang, Chun-Jen Chung, Fanjie Lin
Arizona State University
Tempe, Arizona
{yuli.deng,duolu,dijiang,cchung20,flin11}@asu.edu

ABSTRACT

Hands-on practice is a critical component of cybersecurity education. Most of the existing hands-on exercises or labs materials are usually managed in a problem-centric fashion, while it lacks a coherent way to manage existing labs and provide productive lab exercising plans for cybersecurity learners. With the advantages of big data and natural language processing (NLP) technologies, constructing a large knowledge graph and mining concepts from unstructured text becomes possible, which motivated us to construct a machine learning based lab exercising plan for cybersecurity education. In the research presented by this paper, we have constructed a knowledge graph in the cybersecurity domain using NLP technologies including machine learning based word embedding and hyperlink-based concept mining. We then utilized the knowledge graph during the regular learning process based on the following approaches: 1. We constructed a web-based front-end to visualize the knowledge graph, which allows students to browse and search cybersecurity-related concepts and the corresponding interdependence relations; 2. We created a personalized knowledge graph for each student based on their learning progress and status; 3. We built a personalized lab recommendation system by suggesting more relevant labs based on students' past learning history to maximize their learning outcomes. To measure the effectiveness of the proposed solution, we have conducted a use case study and collected survey data from a graduate-level cybersecurity class. Our study shows that, by leveraging the knowledge graph for the cybersecurity area study, students tend to benefit more and show more interests in cybersecurity area.

CCS CONCEPTS

• **Information systems → Recommender systems**; • **Applied computing → E-learning**; *Interactive learning environments*;

KEYWORDS

Laboratory, Knowledge Graph, Cybersecurity

ACM Reference Format:
Yuli Deng, Duo Lu, Dijiang Huang, Chun-Jen Chung, Fanjie Lin. 2019. Knowledge Graph based Learning Guidance for Cybersecurity Hands-on Labs. In *ACM Global Computing Education Conference 2019 (CompEd '19), May 17–19, 2019, Chengdu,Sichuan, China.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3300115.3309531

1 INTRODUCTION

Using hands-on labs is a critical learning approach for cybersecurity education. Existing lab materials are mainly managed in a problem-centric fashion, in which instructors arrange learning and corresponding lab materials based on a specific topic in security area. However, the inter-lab dependencies are usually complicated and unclear, which hinders both students and instructors to manage learning and teaching materials in a coherent way. It is challenging to build an effective and adaptive learning schedule for students according to their personal background and learning targets: First, efficient cybersecurity education heavily relies on hands-on labs since it focuses more on practical problem-solving skills instead of theory and models. In addition, it is more difficult to organize lab materials than textbooks, let alone manage a complicated experiment environment with multiple hosts, switches, routers, and cables. Second, due to inherent diversities in knowledge and skill sets in cybersecurity education, it is difficult to personalize the learning process and keep track of individual student's learning progress. Third, for instructors, the knowledge sets and instructing materials must be kept up-to-date to cope with the emerging new vulnerabilities, attacks and defense solutions. As a result, it is a continuing process to provide improved learning guidance and plan for students to keep up with the evolving of cybersecurity technologies. The cybersecurity education issues above inspired us to design a new learning solution that can provide a personalized knowledge graph (KG) and guidance to effectively organize, index, recommend reading materials and hands-on labs for learners.

To address the said challenges, we propose CyberKG, a cybersecurity knowledge graph for college-level cybersecurity education, which includes both learning-related and domain-specific knowledge. CyberKG is built on ThoTh Lab [2] [3], a web-based learning platform for cybersecurity hands-on labs by using publicly available hands-on labs, e.g., SEED Labs [19] . Our contribution in this paper is given as following:

1) We built a knowledge graph of concepts and terminologies of cybersecurity based on large amount of public cybersecurity contents, such as Wikipedia and public available cybersecurity lab descriptions. Nodes of the knowledge graph and their dependency relationship are obtained by mining the public cybersecurity contents and security concepts from many cybersecurity glossaries fine-tuned with reading materials and hands-on lab instructions used in our offered security courses.

CompEd '19, May 17–19, 2019, Chengdu,Sichuan, China

ACM ISBN 978-1-4503-6259-7/19/05…$15.00
https://doi.org/10.1145/3300115.3309531

2) We constructed a web-based front-end to visualize the knowledge graph and index all hands-on labs we surfed in the public domain.

3) We built a lab recommendation system for our hands-on lab environment. This system can make recommendations by exploiting the similarity relationship between nodes in the knowledge graph and the association between various knowledge graph nodes and lab instructions.

4) We personalized the knowledge graph for each student to help instructors and students to track individual learning progress.

The remainder of this paper is organized as follows. Section 2 describes related efforts in education area to construct Knowledge Graphs. Section 3 explains the system architecture and the approaches used to construct CyberKG and how we emphasize it as a learning guide for students. Section 4 reports a case study with our experience in teaching cybersecurity at a senior undergraduate level course and discusses various facets of this system. Finally, a short discussion and conclusion of the paper are given in Section 5 and Section 6, respectively.

2 RELATED WORKS

Building a KG is a challenging task though efforts have been done in this area in recent years. There are two major approaches to develop the knowledge bases in education: the first approach primarily relies on individual professional expert, which involves manual work to a certain degree to determine the discrepancies among different professionals and then generate a corresponding consolidated graph. There have been research efforts to describe and categorize knowledge and skills in cybersecurity area by a large board of professionals: Cybersecurity Curricular Guidelines [1], NIST NICE [14], NSA CAE Knowledge Units [13], etc. The outcome of these efforts are well-organized categories in tree structures, which provides clear guidance for human learners when exploring the area. However, it turns out to be significantly challenging for machine learning purposes as these structures contain very limited semantic data that is readable to a machine. The other approach is to automate the generation process by gathering data from web pages and books which is achievable by computers rather without human interaction, e.g., Wikimindmap [15]. There are various solutions been proposed in the last decade of research about building the KG: Mahdisoltani et al. [7] have shown how to construct a knowledge base from Wikipedia in multiple languages; Nickel et al. [12] gave a comprehensive review on training statistical models for large KG's, and further used them to predict new edges in the graph. Recently, attention has been drawn on word embedding for various learning tasks. While a word can be understood by a human being when it appears in the context, its numerical model has to be constructed based on the complex contexts using neural network. In 2013, Mikolov et al. [9], [8] proposed word2vec which included two models: CBOW (Continuous Bag of Words) and Skip-gram to minimize the complexity in computation of continuous vector representation. According to the previous work done by Milne et al. [11], two pages from Wikipedia are defined to be most similar when they have more common information being shared. As for other researches, e.g., Tsai et al. [20] showed that using the Anchor texts of Wikipedia led to better performance in learning the phrase vectors. Grefenstette et al. [6] represented their work on constructing the specialized dictionary by using word2vec to train the Wikipedia data. Speer et al. [18] represented a knowledge graph - ConceptNet5.5, which combines several sources to acquire word embeddings by using distributional semantics, e.g. word2vec. All the related works described above focus on constructing KG for general knowledge. In this paper, we propose to construct a KG for cybersecurity area with an enhanced Word2Vec implementation.

3 SYSTEM DESIGN

Our proposed CyberKG system contains two-stage generation and utilization in its work-flow as shown in Figure 1. We first work out the process to generate the knowledge graph including text data processing, word embedding and the graph structure generation in sections 3.1 and 3.2. Then three applications closely related to personalized learning are built upon CyberKG, which includes lab material indexing and searching (Section 3.3), knowledge graph visualization (Section 3.4), and hands-on lab recommendation (Section 3.5).

3.1 Word Embedding and Similarity Calculation

For computer to understand natural language and the knowledge and concepts within, words need to be represented in a computer-readable manner. Traditionally, NLP systems treat words as discrete symbols which leads to data sparsity and usually means that we may need more data in order to successfully train statistical models. Word embedding is a set of language modeling techniques to represent word as a vector in a low dimensional space. Using vector representations makes natural language computer-readable, which allows us to perform powerful mathematical operations on words to detect their similarities. word2vec[8] is a two-layer neural network that embeds text. Its input is a text corpus and its output are a set of vectors, i.e., the feature vectors for words in that corpus. Our goal of using word2vec is to group the vectors of similar words together in a single vector space, which help us to connect highly related words (concepts) in our knowledge graph.

The main input of the word embedding module is Wikipedia pages. The English version of Wikipedia database dump on May 1st, 2018 from [4] has been used. We develop a toolkit using Python to scrape Wikipedia pages for the categories in computer security section to acquire more accurate related information. The tool that we developed iterates through categories and stores a list of the corresponding information. All main pages in computer security and their related pages in 10 levels of subcategories have been scrapped. There are 7,143 pages obtained under the criteria after removing duplicates. With the processed database dump, we design and develop several tool-kits to train our word embedding model. As a result, there are 4,724,129 unique word embeddings been acquired which are represented in a computer-readable vector space, of which 1,472,477 are Wikipedia pages titles (concepts). For each keyword, the most similar words can be calculated through the cosine similarity between two vectors. For example, for "DDoS", the top ten similar words generated by our word embedding model are shown in Table 1.

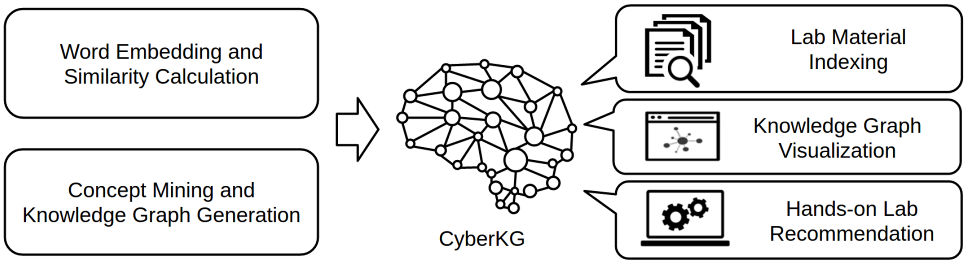

Word Embedding and Similarity Calculation
Concept Mining and Knowledge Graph Generation
CyberKG
Lab Material Indexing
Knowledge Graph Visualization
Hands-on Lab Recommendation

Figure 1: System architecture of CyberKG.

Table 1: Top ten similar words of "DDOS".

word	similarity	word	similarity
botnets	0.833809	honeypot	0.775258
phishing	0.767333	DDoS	0.751166
denial of service	0.708796	spoofing	0.641557
synflood	0.596164	malware	0.593467
attacks	0.584982	crimeware	0.549531

3.2 Knowledge Graph Generation

Knowledge graph, e.g., WordNet [10], is an abundant graph model, whose entity can be represented as a node and the link can be represented by the relationships between nodes. After gathering the word similarities from the previous section, we are able to generate a knowledge graph in our system.

Originally, the knowledge graph generation is handled by human experts. The first step is to do manual text analysis and get a list of concepts, represented as labeled points, and a list of links between these nodes. By combining the lists of concepts and links, a small knowledge graph from a single author is then generated, which is called an author graph. The next step is to combine graphs from various authors into one large graph by identifying common points with each other. When the texts of the nodes deal with the same subject, points with the same label are first identified. Then, human help is needed to identity synonyms for the same concept and connect these synonyms together. One way is to compare the neighborhoods of points. Computing the similarity between two concepts' neighborhood points help us to decide if these two concepts are identical. This method even helps us to detect homonyms, which means the same label but referring to different content.

In our case, each Wikipedia page represents a concept and its explanation (which contains knowledge). There are also hyperlinks within each Wikipedia page that links to other concepts. By analyzing the URL links within one Wikipedia page, we got a simple author graph. For example, on the DDoS page, there are hyperlinks that linked to Exploit, Trojan Horse, IDS, IPS, Computer Fraud, Botnet, Firewall and computer Virus. With 7,143 pages under computer security category in Wikipedia, we now have 7,143 single author graphs ready to be merged together. We utilize the similarities obtained during the word-embedding process described in section 3.1 to further connect these small graphs. Figure 2 showcases how we merge graph of 'Firewall' and 'DDoS' graph into one graph. Word pair like *Antivirus, Computer virus, Spyware, Trojan Horse* are connected together in Figure 2 as their similarity based on word embedding is high. We set the similarity lower limit to 0.8 (while 0 means no relationship and 1 means the two concepts share the same embedding) and connect all node pairs over this similarity threshold. After that, we get one unified and also highly connected knowledge graph ready for further utilization. The threshold 0.8 is used as the lower limit for the following reason according to our experiments: when 0.85 is applied, we get more than 2,000 unconnected nodes, which means these concepts under computer security category are not closely related compared to speaking language words, which is a sign for us to reduce the threshold. There still exist 673 disconnected nodes/small graphs that cannot be included in our main knowledge graph with a threshold as low as 0.7.

3.3 Lab Material Indexing

Within ThoTh Lab, our virtual lab platform, we create a cybersecurity lab repository that is available to instructors and students in our university. We implement our lab design and material from labs of computer science courses within our school and other high-quality open sources labs like SEED Labs from Syracuse University. Instructors are able to upload their own new lab materials into the lab repository at any time. All labs in our lab repository are tagged with keywords by matching the lab material with concepts available in our knowledge graph. For example, keywords our system identified in "Local DNS Attack Lab" from SEED lab include *DNS, bind9, cache, hostname, IP address, LAN, pharming, RFC, rndc, sudo, Ubuntu, Wireshark*. Some of these concepts, like *sudo, Ubuntu* are not directly related to DNS attack, but these are necessary knowledge for each student to finish this lab successfully. Instructors may also edit these concepts before adding them to our lab repository if they think some important concepts were skipped by our system.

We now get one lab to N concepts mapping in CyberKG, which allows us to index labs based on nodes in the knowledge graph, and vice versa. As each lab covers at least one node in the knowledge graph, given any two Lab material A and B, we may obtain their related knowledge graph nodes as the set S_A and S_B. A similarity of these two articles can be calculated as follows:

$$sim(A, B) = \frac{|S_A \cap S_B|}{|S_A \cup S_B|}.$$

General speaking, the more overlapping between two labs' knowledge graph coverage, the more similar these two labs are. This similarity will then be used as the input of the recommendation module described in Section 3.5. Learning material is another component in CyberKG. We currently linked each node in CyberKG to its Wikipedia page, which can serve as basic reading material for students. In order to expand the reading material repository, we

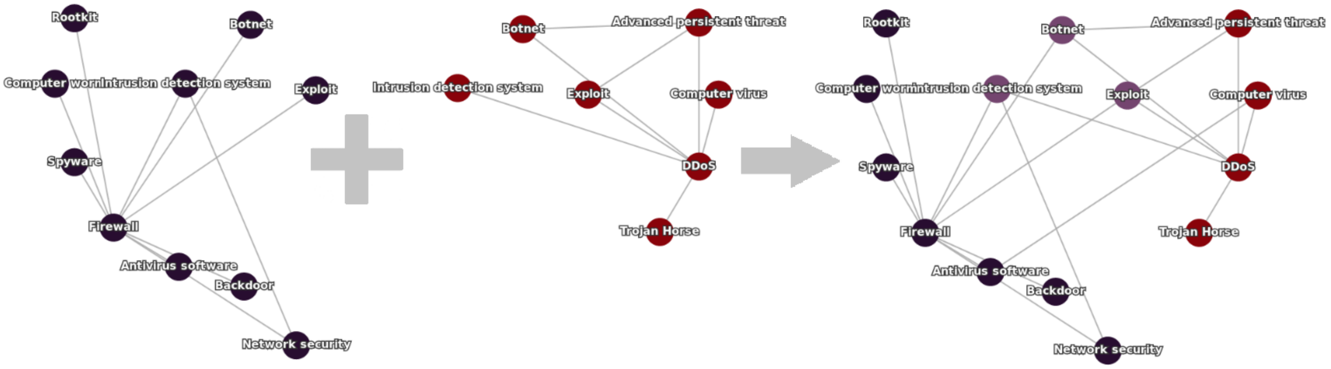


Figure 2: Merge two small graph together based on overlap and word embedding similarity. (Firewall Graph on the right, DDoS graph in the middle and merged result on the right.)

are working on indexing research papers available online with our knowledge graph.

3.4 Knowledge Graph Visualization

With the Knowledge Graph represents in a graph data structure, our next step is to represent the graph in an interactive GUI to empower instructor and students to use it. Since ThoTh Lab itself is a purely web-based lab environment [21], we want to integrate our CyberKG system into the Web UI seamlessly. In this project, we utilized *Echarts*, which is a web-based visualization library that features a plethora of APIs to creating interactive and dynamic content on the web. We first visualized our graph using three different ways. First, a full knowledge graph is presented to the user. As shown in Figure 3 (a). The user may zoom in and hang over nodes in the graph to highlight nodes' neighborhood and gray out unconnected nodes, as shown in Figure 3 (b). Furthermore, the user may click on one node to generate a tree graph using the selected node as root, as shown in Figure 3 (c), leaves in this graph can be further expanded. We also add the search function to help the user locate concept nodes and index function to show the related labs for each node. The color of the nodes represents the lab it belongs to.

We also develop a personalized knowledge graph according to a learner's knowledge gained through the lab experience. The personalized knowledge graph is represented as a subset of the cybersecurity knowledge map. When a learner accomplishes a lab, finish the assigned reading material, and get the pass from an exam or quiz, the personalized knowledge graph is automatically updated. The graphical UI allows the learner to view straightforwardly what has already been covered in his/her learning progress.

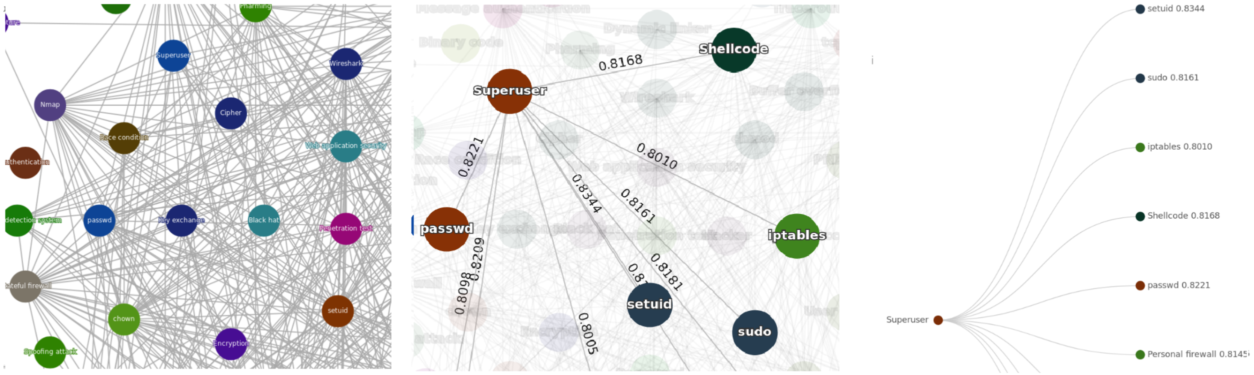
3.5 Recommendation of Hands-on Labs

Traditional education recommendation systems derive the user preferences from predefined features like user age, sex, educational background, previous grades and/or pre-course survey results and etc. Our system utilizes the concepts in CyberKG and in the lab materials to recommend labs that suit the needs for instructor or students.

There are two types of students who use our Lab system. The first type is those who are taking a course which uses our lab platform as an instructional tool. Instructors of such courses need to create syllabus and lab planning for the class at the beginning of each semester. Our system provides instructors with adequate lab materials within our lab repository. An instructor may provide a list of concepts he/she wants to cover during the course run within CyberKG, and our system will return labs related to these concepts based on the concept-lab indexing generated in Section 3.3. During the course run, our system is also able to identify students at-risk or challenged based on their previous lab grades, quiz results, and lab activities to make extra lab practice suggestions. Such suggestions turns out to be simple and straightforward, that contains only one lab, which is either the lab with highest similarity (defined as $sim(A, B)$ in Section 3.3) to the lab which the student was not able to finish or a lab that covers concepts the student lost most points in their exam or quiz.

The second type of students is graduate students who use our virtual lab platform as a self-tutoring platform for cybersecurity study. They are the target audience of our recommendation module. For these students, we first create an entry-survey to check their background in the cybersecurity domain. Then, each student is asked to pick either a set of concepts/knowledge they want to cover or a lab within a lab repository they want to finish independently as their personal learning goal.

The CyberKG system first estimates the concept node coverage of a student based on his/her entry-survey results and update these concepts as mastered in his/her personal knowledge graph. We define the set of mastered concepts C_M and the concepts covered by the student's learning goal as C_G. After that, CyberKG is able to generate a set of paths P_{MG} between C_M and C_G using the knowledge graph. Each path P in P_{MG} contains a set of concepts C_p. Combine all C_p together we got C_P. It is assumed that C_P includes all the concepts a student needs to learn and practice in our lab system in order to achieve his/her learning goal. The last step is to find a set of labs L that covers all concepts in C_P. Currently, our system will generate L where each lab in L got high $sim(A, B)$ with another lab in L. This results in a set of labs that shares a lot of concepts between them. When students start working on such labs, they will have the chance to consolidate their current C_M while learning the new concepts. L becomes our recommendation to a user. Each time a lab is finished and graded, we update C_M and



(a) (b) (c)

Figure 3: Web-UI for Knowledge Graph: (a)Part of KG; (b)Mouse hang-over ’Superuser’;(c)Mouse click on ’Superuser’, which generate a tree using ’Superuser’ root;

regenerate L to see if there is an update needed in the suggested recommendation.

An example of the recommendation process for one user is shown below. At the beginning, our system estimates that his knowledge coverage C_M contains Linux command line, Linux Network and Firewall, and he picks the learning goal C_G containing only SSL Session Hijack. Then CyberKG generated C_P for him as shown in Figure 4. Based on C_P, a L of five labs were recommended to him: (1) Linux web service lab, which covers two concepts in C_P (blue squares), (2) Linux firewall lab, which covers two concepts in C_P (green squares), (3) Packet Sniffing lab, which covers three concepts in C_P (red squares), (4) IP and port scanning lab, which covers three concepts in C_P (purple squares), and (5) SSL Session Hijacking Lab, which covers four concepts in C_P (yellow squares).

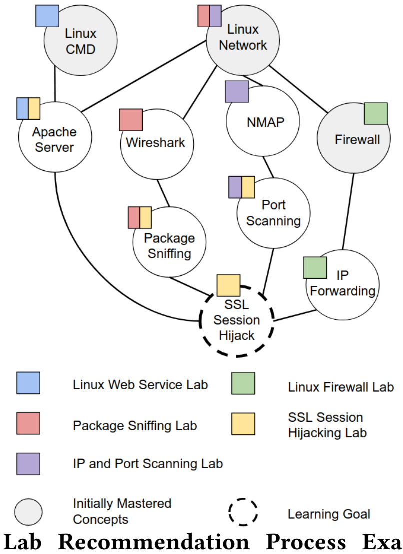


Figure 4: Lab Recommendation Process Example. Best viewed with color.

Another scenario is when a self-learning student uses CyberKG system without providing any personal data and goal. In such case, CyberKG will not give any recommendation at first. Instead, it will obtain lab activity data when users start doing their first few labs and record their knowledge gained through the lab experience to generate C_M for the students. Once enough labs are completed and basic concepts are covered in the user’s personalized knowledge graph, CyberKG system will start providing future lab recommendation based on lab similarity ranking calculated and sorted by $sim(A, B)$. By doing these recommended labs, the user will quickly consolidate the knowledge they have acquired and steadily expand their personalized knowledge graph.

4 CASE STUDY

An experiment using CyberKG was conducted in a graduate-level network security class during Fall 2017 at Arizona State University. This class involves three hands-on labs for computer networks security. 23 graduate students took the course, and all of them finished the pre-survey before the first lab to provide an estimation of their network knowledge backgrounds. During the semester, each student was asked to finish three labs in the virtual lab platform. They were also asked whether they wanted to volunteer in our research practice, and nine students participated. These nine students set their own learning goals on our knowledge graph and then got the recommendation of labs as a outcome of the CyberKG system. They continued to work on these labs, and 8 of them finished all recommended labs. All students’ activities during the labs were recorded in the browser end and inside the virtual machine they used. At the end of the semester, all 23 students were asked to finish an exit survey, where those nine volunteers got extra questions to answer.

In the exit survey, the student satisfaction on our hands-on virtual lab platform has been analyzed and they were also asked about their opinion on CyberKG system. The following questions were asked in the exit survey: (Answer on a scale of 1 to 5, 1 being totally disagree, 5 being fully agree.)

Q1: The virtual lab platform is convenient to access.

Q2: Doing labs in virtual lab platform is easier compare to doing labs in a physical lab.

Q3: Personal knowledge graph in the virtual lab platform is accurate at the beginning of the class.

Q4: Personal knowledge graph is accurate at the end of the class.

Q5: I regularly check my personal knowledge graph.

The extra questions for research volunteers:

Q6: The recommendation a reasonable recommendation for me.

Q7: The connection/similarity between labs recommended to me is noticeable.

Q8: The recommendation system is easy to use.

Q9: Compare to labs required by the course, I find the labs recommended to me more interesting.

Q10: I want to keep on using the system as a self-guidance tool after this class.

Figure 5 shows the average score of each question in the exit survey. While the estimation of student knowledge coverage on the quiz is not accurate (Q3), it improved at the end of the class based on the user activity log (Q4). Among the 9 volunteers who utilized the CyberKG system for learning recommendation, 6 of them agreed that the recommendation is highly related to the topic they pick (in Q7). The survey results also show that majority of students confirmed the usefulness of the recommendation for hands-on labs (Q6), and students' personal preference on lab content have been satisfied by using our system during the semester (Q9).

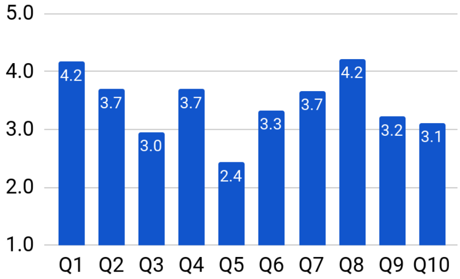


Figure 5: Average score of each question in the exit survey.

5 DISCUSSION

Our current knowledge graph generation module relies heavily on word2vec model to accurately represent words with vectors and calculate similarity among these vectors. However, there are known limitations in this task. For example, it is known that word frequency information in the embedding space affects cosine similarity greatly. As a result, we need to consider word frequency in the training process, especially for those cybersecurity terminologies that do not appear frequently. Another challenge we are facing is how to evaluate the generated knowledge graph. In English language domain, there are several datasets that contain similar word pairs defined by human experts, including Rubenstein and Goodenough dataset[17] and WordSim353 dataset[5]. These datasets can be used as an evaluation baseline for NLP processing modules in English language domain. But such dataset is absent in the cybersecurity domain. As a result, we can only rely on our own domain knowledge to check the results and fine-tune model parameters based on our own judgment. However, word embedding using unsupervised learning methods like word2vec is still the mainstream method on natural language dataset, as these datasets are way too large for human experts to supervise the learning process. One possible solution to these challenges is to construct an ontology with a group of experts in cybersecurity. A few examples of such ontology emerge in recent research works[16]. The difference between ontology and our current knowledge graph is that the edge in the ontology is well defined during the construction, while the links in a knowledge graph may be meaningless. This makes the merging of ontology in the same domain simple and straightforward. Our next step is to add a human defined ontology module in our system to evaluate and trim the knowledge graph generated by CyberKG, then build a feedback loop and editing tool for users to give them the opportunity to help us improve CyberKG.

There are also several lessons we learned during the case study. First, students from all backgrounds take cybersecurity class. This is not surprising at all, but we were still shocked by the huge gap between students. At the beginning of the class, some students have no background on computer network, while some students already mastered most concepts we are going to cover in all labs. This makes the entry survey/quiz and estimation important as these students definitely need different kind of instruction from the very beginning. Second, it is extremely hard to stop students from cheating during online labs. Common cheats we found during the case study were searching answer online (as SEED Lab we used are widely used) and doing the labs together. Third, online support availability is important to keep students motivated. Students may do the lab any time anywhere, but when they encounter a problem that they struggle, they'll need help right away or they'll procrastinate. One solution to this is group lab, which enable students to help each other. Lastly, we want to further investigate CyperKG's impact on students' learning outcomes [22].

6 CONCLUSION AND FUTURE WORK

This paper describes our efforts towards creating a knowledge graph to represent concepts and their relationships in the cybersecurity domain. This work is intended to provide an organized knowledge that incorporates information from a large variety of data sources including Wikipedia pages and instruction materials, which includes all relevant concepts within the domain for educational usage. We then applied such knowledge graph into an e-learning virtual lab platform to test it. When using the knowledge graph as a recommendation/guidance tool for students, our case study proves that our prototyped system is able to meet students' expectation in making the desired recommendation.

In future work, we want to incorporate more unstructured data into our system, including but not limited to textbooks, internet web pages, and online video transcripts. We also plan to incorporate cybersecurity ontology which is intended to support our knowledge graph generation. We also need to come up with innovative solutions to the other challenges discussed in Section 5. Further experiments and in-class studies are necessary for system validation. Our ultimate goal is to build a knowledge graph that will serve as the backbone of the cybersecurity education domain, which would evolve and grow with additional cybersecurity lab sets as they become available, being fully adaptive to different learners who want to utilize it.

ACKNOWLEDGEMENT

All authors are gratefully thankful for research grants from NSF DGE-1723440, NSFC 61628201 and 61571375.

REFERENCES

[1] ACM. 2017. Cybersecurity Curricular Guidelines. https://cybered.hosting.acm.org/wp/

[2] Yuli Deng, Dijiang Huang, and Chun-Jen Chung. 2017. ThoTh Lab: A Personalized Learning Framework for CS Hands-on Projects. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. ACM, 706–706.

[3] Yuli Deng, Duo Lu, Chun-Jen Cheng, Dijiang Huang, and Zhen Zeng. 2018. Personalized Learning in a Virtual Hands-on Lab Platform for Computer Science Education. In *2018 IEEE Frontiers in Education Conference (FIE)*.

[4] Wikimedia Foundation. 2018. Wikimedia Downloads. https://dumps.wikimedia.org/

[5] Evgeniy Gabrilovich. 2002. The WordSimilarity-353 Test Collection. http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/

[6] Gregory Grefenstette and Lawrence Muchemi. 2016. Determining the Characteristic Vocabulary for a Specialized Dictionary using Word2vec and a Directed Crawler. *arXiv preprint arXiv:1605.09564* (2016).

[7] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M Suchanek. 2013. Yago3: A knowledge base from multilingual wikipedias. In *CIDR*.

[8] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[9] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[10] George A. Miller. 1995. WordNet: A Lexical Database for English. *Commun. ACM* 38, 11 (Nov. 1995), 39–41. https://doi.org/10.1145/219717.219748

[11] David Milne and Ian H Witten. 2008. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 509–518.

[12] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2016. A review of relational machine learning for knowledge graphs. *Proc. IEEE* 104, 1 (2016), 11–33.

[13] NIETP. 2018. CAE REQUIREMENTS AND RESOURCES. https://www.iad.gov/NIETP/CAERequirements.cfm

[14] NIST. 2018. NICE Cybersecurity Workforce Framework. https://www.nist.gov/itl/applied-cybersecurity/nice/

[15] F Nyffenegger. 2009. WikiMindMap.

[16] Leo Obrst, Penny Chase, and Richard Markeloff. 2012. Developing an Ontology of the Cyber Security Domain. In *STIDS*.

[17] Herbert Rubenstein and John B. Goodenough. 1965. Contextual Correlates of Synonymy. *Commun. ACM* 8, 10 (Oct. 1965), 627–633. https://doi.org/10.1145/365628.365657

[18] Robert Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge.. In *AAAI*. 4444–4451.

[19] Syracuse University. 2018. SEED Labs. http://www.cis.syr.edu/~wedu/seed/labs.html.

[20] Chen-Tse Tsai and Dan Roth. 2016. Cross-lingual wikification using multilingual embeddings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 589–598.

[21] L. Xu, D. Huang, and W. Tsai. 2014. Cloud-Based Virtual Laboratory for Network Security Education. *IEEE Transactions on Education* 57, 3 (Aug 2014), 145–150. https://doi.org/10.1109/TE.2013.2282285

[22] Zhen Zeng, Yuli Deng, I-Han Hsiao, and Dijiang Huang. 2018. Understanding StudentsâĂŹ Engagement behavior in Virtual Hands-on Lab: Findings from a Computer Network Security Course. In *2018 IEEE Frontiers in Education Conference (FIE)*.