A Bootstrapping Approach for Developing a Cyber-Security Ontology Using Textbook Index Terms

Arwa Wali
Department of Computer Science
New Jersey Institute of Technology
Newark, NJ, USA
e-mail: amw7@njit.edu

Soon Ae Chun
Columbia University &
CUNY College of Staten Island
Staten Island, NY,USA
e-mail: Soon.Chun@csi.cuny.edu

James Geller
Department of Computer Science
New Jersey Institute of Technology
Newark, NJ, USA
e-mail: geller@njit.edu

Abstract—**Developing a domain ontology with concepts and relationships between them is a challenge, since knowledge engineering is a labor intensive process that can be a bottleneck and is often not scalable. Developing a cyber-security ontology is no exception. A security ontology can improve search for security learning resources that are scattered in different locations in different formats, since it can provide a common controlled vocabulary to annotate the resources with consistent semantics. In this paper, we present a bootstrapping method for developing a cyber-security ontology using both a security textbook index that provides a list of terms in the security domain and an existing security ontology as a scaffold. The bootstrapping approach automatically extracts the textbook index terms (concepts), derives a relationship to a concept in the security ontology for each and classifies them into the existing security ontology. The bootstrapping approach relies on the exact and approximate similarity matching of concepts as well as the category information obtained from external sources such as Wikipedia. The results show feasibility of our method to develop a more comprehensive and scalable cyber-security ontology with rich concepts from a textbook index. We provide criteria used to select a scaffold ontology among existing ontologies. The current approach can be improved by considering synonyms, deep searching in Wikipedia categories, and domain expert validation.**

Keywords-security ontology; cyber-security; learning objects; textbook; index terms

I. INTRODUCTION

With a vast amount of educational resources available online that ranges from digital text to audios and to recorded video lectures as well as textbook resources, an ability to search appropriate multi-modal lecture materials that fit to the requested content and that fit to the needs and levels of user preferences is essential. One of the important areas that need awareness, training and education is cyber-security and information assurance. The growing number of cyber-attacks in the private sector has cost companies an estimated $5.9 million on average in 2011, up from $3.8 million in 2010 [1]. They include web-based denial of service attacks, phishing, malicious code, malicious insider attacks, and sophisticated industrial espionage attacks that target financial and enterprise systems for theft and disruption. Many governments are also targets of national security attacks (e.g. national intelligence data breaches, cyber terrorist attacks), or attacks on critical infrastructure (e.g. transportation systems, grid system attacks) by criminals or self-righteous groups as well as by foreign countries. In addition, attacks on personal devices are rapidly growing with the explosive increase in mobile and smart phone users [2]. This includes attacks on mobile operating systems and mobile apps in order to steal personal information, e.g. to access bank records or other sensitive data. For example, based on Juniper Networks (JNPR) research, the near-field communication (NFC) chips that are used in around 300 million smart-phones for mobile payments all over the globe, may lead to $50 billion worth of global NFC transactions. This will make all smart phones payment systems major targets for cyber attacks [3].

On the other hand, trained professionals with the skills to effectively counter and mitigate these cyber attacks are in great demand, emphasizing the need for expanding cyber-security education. One way to facilitate the education in cyber-security is to make available a self-paced learning environment that allows search for particular topics of interest that may be needed to complement traditional class-based learning. This learning paradigm embraces free and open online educational resources as MOOC (Massive Online Open Course) platforms, other independent online education sites such as Khan Academy, as well as numerous educational resources (tutorials or lecture notes) shared by volunteers on the web.

The key challenge in this open environment with vast resources is to identify and locate the right learning resources at the right time. To meet this challenge, our project includes the goal to develop a core security ontology to be able to semantically annotate the educational contents and link the resources to one another, based on their semantic relationships. The location where each learning resource resides should be transparent to the users. Each educational resource should be easily locatable through rich annotations and content tags, regardless whether it is a part of a textbook, a slide in a PowerPoint presentation or a video of a professor lecturing to her class. One of our goals is to develop a core security ontology to enable users (teachers and learners) to annotate the security-learning resources they encounter to aid later searches and semantically correct linking of the learning resources.

Constructing a domain ontology involves the specification of terms and relationships among them. The ontology can help domain experts and users to share information in their field, can support reuse of domain

978-0-7695-5008-4 2013

DOI 10.1109/ARES.2013.75

knowledge, and makes domain assumptions explicit. This helps students and teachers and allows building of computer systems that encourage learning and reasoning about this domain [4]. Above all, it provides a controlled vocabulary and a common semantics that can help interoperability among different computer systems. The ontology may also be used for annotating educational resources, providing common labels or tags to identify and search for them. However, the manual development of ontologies has been a bottleneck as it is labor-intensive and error-prone, often resulting in unscalable systems.

Most of the automatic domain ontology development research depends on extracting information (especially concepts) from unstructured text either manually, which is time consuming, or (semi-) automatically, which is a difficult task and needs extensive human review. Xuefeng et al. [5] demonstrated building an ontology based on extracting information from e-textbooks, since e-learning materials such as the electronic copies of textbooks are considered rich sources of domain concepts. However, their work was limited to a Chinese "Discrete Mathematics" textbook. Their automatic methodology for extracting domain concepts and developing an ontology for e-learning purposes employs different existing technologies and tools, such as ICTCLAS (Chinese Lexical Analysis System), GATE (General Architecture for Text Engineering), and JAPE (Java Annotation Pattern Engine) [5]. However, processing a whole textbook for extracting concepts requires a major effort in processing natural language text to identify domain concepts.

In this paper, we circumvent the need for extracting domain concepts from text by leveraging the index terms in the back of a textbook. Most text books have a back-of-the-book index of words (terms) with their page numbers. Such an index can help the reader to easily locate the information she needs. These index terms are usually domain concepts, such as people, places, or events relevant to the domain, and can be generated by any indexing software such as Cindex, Macrex and SkyIndex [6].

Since an index contains (presumably) mostly entries that are domain concepts, we avoid the need for extracting concepts from unstructured text. Instead, this paper focuses on how to integrate the concepts listed in the index into the hierarchical structure of a preexisting security ontology. We investigated several existing, partially developed security ontologies, and bootstrapped from one of these ontologies as a base, enriching it with domain concepts from the textbook index. In other words, we focused on the task of deriving the hierarchical relationship of each concept in the index to the concepts in the existing security ontology.

The described methodology is completely domain-independent and can be applied in any educational topic area where an electronic textbook with an index as well as a preexisting ontology are accessible. We used different strategies for linking a concept from the index to the existing ontology, such as concept matching between the index terms and the ontology classes, and relationship derivation using linguistic matching and external open source knowledge bases such as Wikipedia. Even though this new approach for enhancing a domain ontology from a textbook's index can be applied to any domain, we are applying it to a security textbook to build a security domain ontology for indexing educational materials.

Section II gives some background on building domain ontologies, and introduces the existing security ontologies that we considered as a basis of the bootstrapping approach. In Section III, we present our criteria for choosing a preexisting ontology and compare the index terms from two different security textbooks. We present different strategies for merging the concepts from the textbook index into the chosen security ontology in Sections IV. Sections V, VI and VII present our preliminary research results, discussion, and conclusions, respectively.

II. RELATED WORK

A. Ontology Development

Many approaches for developing ontologies and identifying class hierarchies have been reported in the literature. The most difficult approach is to construct an ontology manually by collecting all concepts and defining relationships between them. This can be done by a top-down, a bottom-up, or a combination method. All three methods primarily depend on defining the more general terms and subsequent specialization of the concepts in either direction. LoLaLi is an example of an ontology that was manually built. This work addressed the importance of building a domain-specific ontology for teaching and learning purposes [7], using Semantic Web technologies, to support learning and reading electronic scientific publications. Different versions of the LoLaLi ontology have been implemented using various technologies. This included building a LoLaLi hierarchy using XML (eXtensible Markup Language) with a document type definition (DTD), and RDF (Resource Description Framework) with the ontology editor Protégé 1.8 and then with Sesame in connection with the RQL query language. Although building LoLaLi manually was expected to result in an ontology that is more accurate in its concepts and the relationships between them than achievable by building it automatically, the process was very time consuming.

The second method is building an ontology by automatic parsing of English text. There are several methods used for this approach, such as clustering, linguistic pattern matching, formal concept analysis, or ontology alignment. Hindle, one of the methods based on the clustering approach, is similarity-based. Hindle used the similarity between terms based on subject-verb-object relations in a corpus of text and showed reasonable success in determining semantic relations between words in the text [8]. Hearst et al. used a linguistic pattern matching method by indentifying a set of lexico-syntactic patterns for hyponymy to find semantic relationships between terms from large corpora of English text [9]. The third method of automatic parsing of English text is formal concept analysis. This method is used for extracting monotonic inheritance relations between objects from unstructured data. These relationships are described through a set of attributes that build formal contexts for objects. From these, a formal concept lattice is derived that can be converted into a concept hierarchy [10, 11].

The fourth method for developing a comprehensive ontology is by ontology alignment. BLOOMS is an

example for building an ontology by alignment or ontology matching [12]. It uses a bootstrapping approach to extract the source information for the ontology from existing Linked Open Data (LOD) with Wikipedia categories and an Alignment API. To determine the correct subclass relationship between two concepts (T1, T2), this method first builds a BLOOM forest for each concept, using category hierarchy information from Wikipedia. Then, the decision that a concept is a subclass of another concept is based on pruning the common nodes in two forests and calculating the ratio of non-common nodes over total nodes in the forest for each forest such that the concept with the larger ratio is considered a superclass of the other concept.

In previous research, we used a methodology for automatic construction of a domain ontology, by combining WordNet concepts with domain-specific concept information extracted from Deep Web service pages [13]. The Google AJAX Search API and JSON have also been used to extract information from the web and process it for ontology construction [14].

Most of the previous approaches for developing domain ontologies based on unstructured data from the web or text corpora suffer from several problems. The methods that use the web as their text base are dependent on the layout of information on web pages that may be changing rapidly, thus the extraction algorithms may have to be adjusted repeatedly, raising issues of reliability and scalability.

Pattanasri et al. [15] developed a textbook ontology using textbook metadata such as the index and the table of content. They used the index at the back of the book and built an index ontology using an OWL representation from the heading-subheading relationships of index entries. The table of contents was used to build a textbook ontology that represents the hierarchical organization of textbook segments. The concepts in each ontology are cross-referenced with page numbers to refer to corresponding textbook segments or slide page numbers, so that these ontologies help the learners formulate queries to search inside of lecture materials. The index ontology was developed using a three level structure of index entries, i.e., by defining the heading of the index entry as a topic concept. Topics are connected to subheading entry terms by subtopicOf relationships, and sub-subheading entry terms are likewise defined with subtopicOf relationships. Page numbers are linked to concept nodes. However, the index ontology is only usable for searching the specific e-learning materials that it was built for, which means it is not reusable. In addition, since this research only considers three levels of index headings, which is an arbitrary limitation, there is not enough structure in the resulting ontology. Hence it lacks completeness and comprehensiveness, especially in relationships among domain concepts. The resulting structure is akin to a topic hierarchy with mapping tables, with pointers to a limited set of the learning resources. It is doubtful whether this approach can easily scale to tagging a large number of learning resources.

To the best of our knowledge, our research is the first that combines a back of the textbook index with an existing security ontology as a bootstrapping structure to build a more complete security domain ontology.

B. Security Ontology

There are several preexisting security ontologies; some of them are dedicated to a specific security subdomain such as attacks and intrusions [16, 17] or SIP-VoIP based services [17, 18]. Others are general security ontologies that attempt to cover the whole computer security domain, such as Herzog et al.'s security ontology [19], Fenz and Ekelhart's security ontology [20], and the NRL security ontology by Kim et al. [21].

Different methods were used to extract ontology concepts for the general security ontologies. Herzog's group developed their security ontology manually according to established ontology design principles [22]. Their ontology was built based on the classic components of risk analysis, assets, threats, vulnerabilities, and countermeasures. This ontology contains a detailed domain vocabulary to answer any specific queries, support machine reasoning and provides natural language definitions for the domain terms. The ontology was implemented using Web Ontology Language (OWL) [19].

The security ontology by Fenz and his group was proposed based on the security relationship model described in the National Institute of Standards and Technology Special Publication 800-12 [23]. Their security ontology includes low cost risk management and threat analysis. It also reuses existing taxonomies for importing concept definitions and relationships, such as the German IT Grundschutz Manual [24] and The United Nations Standard Products and Services Codes. They used around ten resources to automatically extract concepts for high-level threats. The ontology concepts were coded in graphical, textual or description logic (DL) formats and then transformed into the OWL format [25].

The NRL security ontology was developed by importing seven specific security domain ontologies and it is focused on security functional aspect resources. Three ontologies were imported from existing DAML security ontologies [26] while the other four were implemented using OWL. It is not clear whether the development methodologies used for these mission-driven ontologies are automated.

The cited security ontologies are still in active development. None of them is considered comprehensive enough to cover all existing security concepts. In addition, the existing ontology concepts are limited in their attributes, and some of the expressions used in defining them are not appropriate [27]. The goal of our research is to create a "close to complete" security ontology that allows sharing security knowledge and supports web services queries for learning purposes. In addition, the algorithms used to build the security ontology are reusable for building ontologies in other domains.

III. OUR APPROACH

A. Selection of Security Core Ontology

We considered three different preexisting cyber-security-related ontologies, Herzog et al.'s [19], Fenz et al.'s [20], and Kim et al.'s [21]. These ontologies have 463, 635, and 75 concepts, respectively. We compared these ontologies to choose one for our bootstrapping process to build a more enriched security ontology. As discussed, our goal is to be able to annotate and search the

learning materials in computer security courses using a comprehensive ontology. We found the ontology by Herzog et al. most appropriate, since it has well defined concepts and relationships that can be used for annotations. The ontology by Fenz et al. has more concepts, so it would appear to be a better choice. However, the concept names also reflect relationships between security terms, which makes the class names very long and thus less useful for annotating learning materials or improving search queries. Moreover, this ontology concentrates on security hardware-related terms (assets) more than on software terms. See Figure 1 for partial hierarchies of these two ontologies.

The ontology by Kim et al. depends on integrating and importing many different security subdomain ontologies such as credentials, security assurance, security algorithms, etc. Besides that it has a very limited number of concepts, and importing many ontologies into one not only makes the ontology hard to extend but also requires a sufficient amount of time to correct and validate the hierarchy.

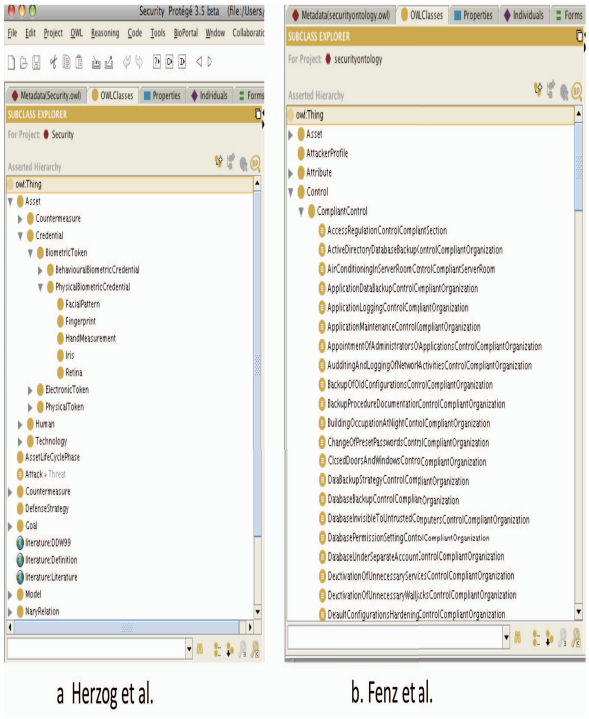

Figure 1. Concepts in two ontologies

The two criteria we used in choosing the ontology by Herzog's group as a bootstrap structure were annotation power, and coverage. Annotation power means that the ontology contains security concepts that are useful for annotating the learning objects, e.g., *Anti-VirusSoftware*, and *FileAccessControl*. In contrast, an ontology that uses long concept names such as *HandlingOfDrivesForRemovableMediaControlCompliant Organization*, is less useful for annotation or search. "Coverage" addresses the question whether the ontology contains concepts from sufficiently many relevant security subareas. Although the number of concepts in the ontology by Fenz et al. is larger than the number of concepts in Herzog et al.'s ontology, it focuses on hardware security concepts, ignoring other important subareas of security such as vulnerabilities, or control. To determine coverage, we compared the index terms in the introductory computer security textbook by Goodrich and Tamassia with the main classes in both ontologies, Asset, Control, Goal, Vulnerability, etc. If the index term x occurs "under" the concept y in an ontology (i.e. x⊆ y), then x is covered by the ontology's concept y. The security ontology of Herzog's group covered 221 terms under all main concepts, with different degrees of coverage for each one. In the ontology of Fenz's group, 192 terms were classified, mostly under the *Asset* concept and its subconcepts, e.g., *movable asset*.

B. *Selection of Textbook's index:*

Another issue we faced was selecting a security index for automating the security ontology bootstrapping process. Two textbooks were the candidates, Introduction to Computer Security by Goodrich and Tamassia [28], and Counter Hack Reloaded: A Step-by-Step Guide to Computer Attacks and Effective Defenses by Skoudis and Liston [29]. There are many differences between these two indexes as seen in Figure 2 and Figure 3.

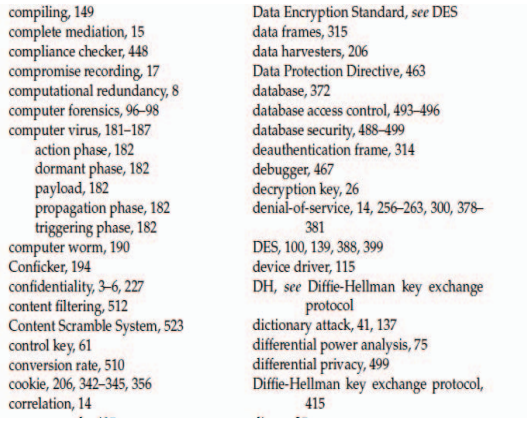

Figure 2. Part of Goodrich's security textbook index

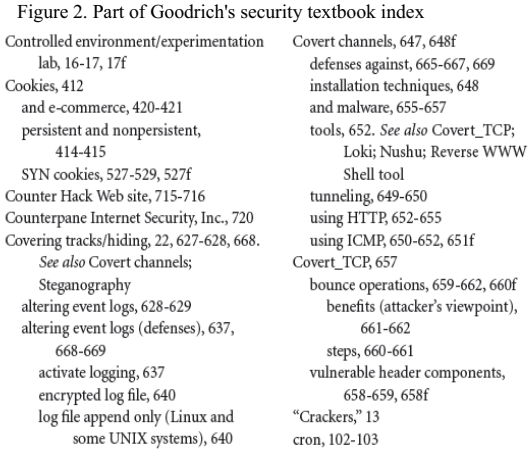

Figure 3. Part of Skoudis' security textbook index

First, Skoudis' index contains many phrases instead of single word terms and is organized in three levels with main terms, subterms and subsubterms. There are many self-contained concepts, but without clear, meaningful relationships between the subterms and the main terms. Goodrich's index has more simple terms rather than phrases, and it organizes security terms with only one level of related subterms. Secondly, after the sequences of page numbers for each main term in Goodrich's index, there might be a synonym or abbreviation for that term, if one exists, while no clear relation is apparent in Skoudis' index between a term and the collection of terms following the page numbers. Thirdly, Goodrich's index terms are mainly security terms or computer terms. In contrast to that, Skoudis' index also includes people's names and long phrases that are not useful. Based on these distinctions, Goodrich's index was chosen as the main input for our research to build the security ontology.

IV. BOOTSTRPPING SECURITY ONTOLOGY

In the following, we describe the method for automatic ontology bootstrapping used to enrich the existing ontology (specifically, the ontology by Herzog et al.) with security terms from the textbook index. A flow chart of the method is shown in Figure 4. Some of the key modules are explained below.

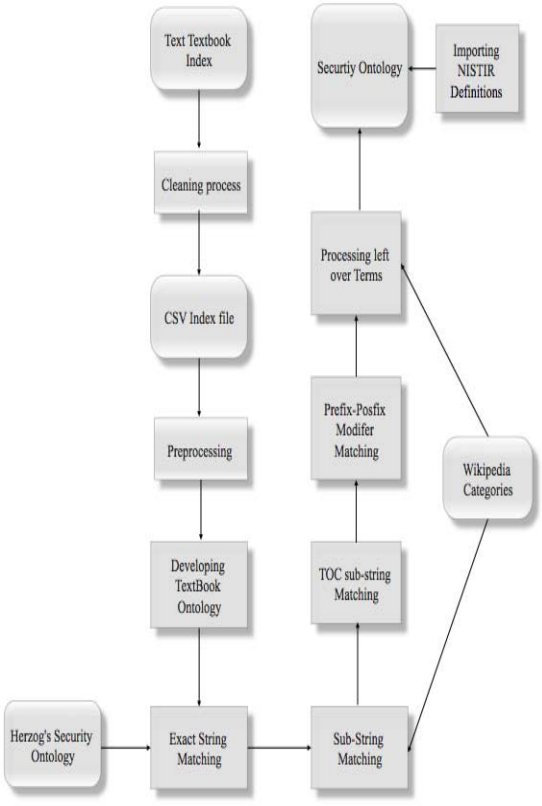

Figure 4. Flow chart of bootstrapping security ontology with textbook's index terms

A. Preprocessing the index file

Before ontology bootstrapping with the index terms, preprocessing of the index data was performed by defining common relationships between the terms, such as hasAbbreviation, hasSynonym, hasLocation (page numbers in the textbook), and hasSubTerm. These relationships help in the following processes to extract the learning objects and to set their properties and semantic relationships. For example, when a security term, "*CBC*" appears together with the key word "see" followed by an acronym or a second term, e.g. "*See cipher-block chaining mode*," then there is a strong relationship between these two terms. Establishing these two terms as Equivalent or as related by hasAbbreviation simplifies the process of building the ontology. Figure 5 shows an intermediate result of preprocessing.

	A	B	D	E	F	G	H	I	J	K	L
1	hasName	hasSubTerm	hasAbbreviation	hasPageNum	hasPageNum	hasPageNum	hasPageNum	hasPageNum	hasPageNum	hasPageNum	hasPageNum
2	A.A.A.			9							
3	access control			4	140						
4	access control list		ACL	20-21	40	140-143	171	453	493		
5	access control matrix			19							
6	access point		AP	314							
7	accountability			516							
8	ACK scan			323							
9	ACK storm			254							
10	action attribute set			448							
11	ActiveX			355	366	383					
12	adaptive chosen-plaintext at			389							
13	address space			124							
14	address space layout randor		ASLR	158	170						
15	AddRoundKey			400							
16	Adobe Flash			353	354						
17	Advanced Encryption Standa		AES	27	77	100	102	293	296	319	38

Figure 5. Pre-Processing the index file by defining properties and relationships

B. Developing security ontology from index file

Bootstrapping the cyber-security ontology with terms extracted from Goodrich's index has been achieved using the Protégé API, by defining all main terms as classes and the subterms as subclasses. Abbreviations and synonyms are represented in the ontology as object properties or equivalent classes. In addition, data type properties such as term locations and number of occurrences of a term in the textbook were defined for each class. Having this information is important for guiding the learning process. For example, a term that occurs on many pages is considered relatively more important for learning purposes. If a term appears earlier in the book that indicates that it is relatively more basic.

C. Exact string matching

The main method of this research is bootstrapping, i.e. expanding or enriching an existing security ontology to include most of the security concepts (terms) from a textbook index, also including the semantic relationships between them. As mentioned previously, the ontology by Herzog et al. was selected as a basis for developing our security ontology. Given a term from the textbook index, we process it as follows. First we compare it with the concepts in the ontology. If there is an exact match, we assume that the concept already exists in the ontology. (While there are homonyms in the English language, this

problem is reduced in a narrow domain such as computer security.) We used Porter's stemming algorithm for this similarity matching to ensure that word variations are taken into account. For example, Goodrich's term "Vulnerabilities" is considered a match with "Vulnerability" in Herzog's ontology. In addition, once a match has been determined for the concept, we add additional information for the matched concept that was discovered in the preprocessing stage, such as the hasLocation property to store page numbers, or IsEquivalentTo or hasSubClass, etc. If the index term has a subterm under it that was extracted in the preprocessing stage, and none of the subclasses under Herzog's concept contains it already, then we add the subterm with its properties and instances to the ontology.

D. Substring matching

The previous steps did not lead to a sufficient number of matches. The next method we used was substring matching. We compared an index term with each concept in the ontology, such that if any of them is a part of the other concept (after applying Porter's stemming algorithm), we added Goodrich's term as a subclass of Herzog's ontology concept, after checking it against its Wikipedia categories, to boost our confidence that the subclass relationship holds.

The Wikipedia category information API function was used to extract all categories of Goodrich's term. Then, if any of these categories was equal to Herzog's concept it is compared with, then we have increased confidence that Goodrich's term is a subclass of Herzog's concept. In the case that Goodrich's term has two or more of Herzog's concepts as its Wiki categories then all of Herzog's concepts are defined as superclasses for it.

E. TOC substring matching and prefix –postfix modifier matching:

For the terms that could not be categorized by the similarity bootstrapping methods described in the previous two sections, we used the table of contents (TOC) in Goodrich's textbook. We compared an index term with the title headings and subtitles of Goodrich's textbook TOC, in such a way that if any of them was equal to or a substring of the concept in the bootstrapping ontology, we established the appropriate subclass relationship. (We applied Porter's stemming algorithm also at this step.)

Prefix and postfix modifiers (as in the compound noun N+N format) in the index term were also used to identify subclass relationships. That is, if any modifier in the beginning or at the end of any of Goodrich's terms exists as an ontology concept, then we plugged in Goodrich's compound term as a subclass of that concept.

Protégé does not allow two concepts with the same name in one ontology file. Therefore, if we have two concepts that are functioning as prefix and postfix modifiers for any of Goodrich's concepts, then we consider both of them as potential superclasses for that concept. For example, if we have "Email Worm" as one of Goodrich's concepts and we have two concepts in the ontology, "Email" and "Worm," then both of them are interpreted to be superclasses of "Email Worm." Naturally, there will be many cases where human validation is necessary to ensure the correct classification.

F. Processing the leftover terms

The remaining index terms in Goodrich's textbook that were not classified after all the previous bootstrapping steps are processed using the Wikipedia API by extracting Wiki categories for the concepts in two steps. First, we test for equality between Wiki categories for Goodrich's term and the security ontology concept. If they are equal, then we have an indication that this term belongs to this concept, and we add it as subclass as before. Secondly, if none of the concepts are equal to any of Goodrich's term categories that we extracted from Wikipedia, then the combination of a superclass and its subclasses is checked as substring of any category name. If that test succeeds, then we have again an indication that this term belongs to this concept, and we add it as subclass for that concept. For example, the categories of the index term "Mydoom" are "Email Worms," and "Windows Viruses." Because we have a concept name in the ontology called "EmailWorm," "Mydoom" is classified under this concept.

G. Importing security concept definitions

Finally, to build a comprehensive security ontology that can be used for tagging learning materials for any educational purpose, we used a glossary of key information security terms that is provided by the National Institute of Standards and Technology (NIST) Interagency Report (NISTIR) [30] and imported the definitions of the included security concepts into the ontology.

The pseudo-code for deriving the hierarchical relationship of a security index term to the security class in the bootstrapping ontology is shown below in Table 1.

I. RESULTS

The Goodrich textbook index [25] contains 724 terms, either main terms or subterms for the main terms. The bootstrapping methods we discussed in section IV for automatically categorizing these security terms to expand the existing ontology were applied, and the numbers of concepts resulting from each module were recorded as shown in Table 2. The first bootstrapping method was the exact match case where the properties and instances of matching concepts were added. The second method was to plug in the terms with matching substrings of the ontology concept and add their information and properties. The third method was to add TOC substring matches and prefix and postfix modifier matches. The last step was to import NISTIR definitions for all security concepts in the ontology. The statistics of recognized concepts from Goodrich's index are displayed below in Table 2.

TABLE 1: PROCESSING OF INDEX TERMS AS PSEUDO-CODE

```
IndexSecurityOntology(Ontology Goodrich, Ontology Herzog)
{  For each Goodrich class i ,  Herzog class j do
    //Exact matching
    If (stemming (i)= stemming (j))
            Attach all properties of class i to class j
            Define an instance for class j with Goodrich locations
            properties
            Import NISTIR definition if it exists
    //Substring matching
    Elseif (stemming (i) ⊆ stemming (j) or stemming (j) ⊆ stemming (i))
            Categories []= WikiAPICategories( i's name)
            If (any of Categories = j)
                Plug in i as a subclass of j and attach its properties
                Define an instance of j with Goodrich locations properties
                Import definition if it exists.
    //TOC matching
    Elseif (j ⊆ TOC title x or TOC title x ⊆ j)
            Plug in i as a subclass of j and attach its properties
            Define an instance of j with Goodrich locations properties
            Import NISTIR definition if it exists
     // Prefix- postfix modifier matching
     Elseif (i appears at the beginning or at the end of  j)
              If  (i already exists in the ontology)
                        Assert j is superclass i
               Else
                        Plug in i as a subclass of j and attach its properties
                        Define an instance of j with Goodrich locations
                        properties
                        Import NISTIR definition if it exists.
     //processing left over terms
    Elseif (i's Wiki categories = to any j) or  ((j + j's superclass) ⊆ i's
Wiki categories)
            Plug in i as a subclass of j and attach its properties
            Define an instance of j with Goodrich locations properties
            Import NISTIR definition if it exists
     End if
  End for
//Add subclasses of Goodrich concepts
 For each Goodrich class i ,  Herzog class j do
     If  (i has subclasses Y[ ] that do not exist under j)
            Plug in subclasses Y[ ] of  j
            Define instances of Y[ ] with Goodrich locations properties
            Import Y[ ]'s NISTIR definitions if they exist.
     End if
  End for
}
```

TABLE 2 THE NUMBER OF RESULTING CONCEPTS FROM EACH MODULE

Module	Number of recognized concepts	Ratio
Exact string matching	88	12.15 %
Substring matching	52	7.2 %
TOC substring matching and Prefix/postfix modifier matching	88	12.15 %
Processing left-over terms	35	4.8%

While the total number of concepts at the starting point in Herzog's ontology was 463, the resulting expanded ontology contains 638 concepts. The resulting ontology is represented in OWL format as seen in Figure 6.

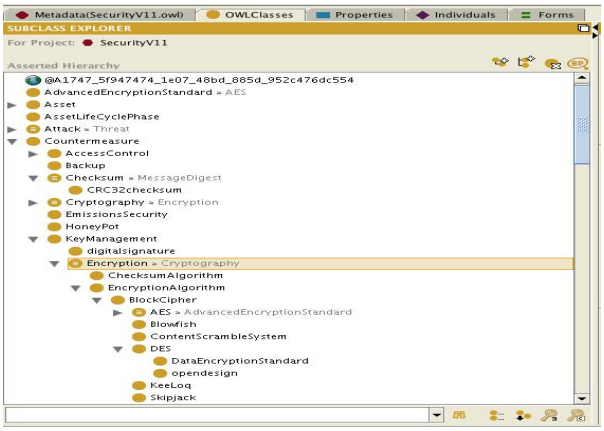

Figure 6. Top Level of Resulting Ontology

II. DISCUSSION

The remarkable differences between the two textbook indexes in structure and content make it difficult to build one algorithm that can be applied to both indexes to bootstrap a security ontology. Besides that, any index is likely to need heavy cleaning before attempting an automatic bootstrapping of the ontology. However, the number of terms matched by our approach was considerably smaller than the number of terms listed in the index. We have the following interpretation of this limited result. First, the Wiki API is not good enough to find the correct classes for the concepts, since it gives either no category information or too many ambiguous categories for the index terms, which leads to too many terms not being classified. For example, the term "ACK scan" has no Wiki categories and "Apple" has too many ambiguous categories. We plan to extract the textual sentences that the index page number(s) where a term occurs refer to, and consider them to derive the correct relationship information.

Secondly, many Goodrich terms refer to an article in Wikipedia, but none of their categories appear as concepts in the security ontology. For example, "Wi-Fi Protected Access" has a Wikipedia article, and it has three categories, Cryptographic protocols, Computer network security, and IEEE 802.11, but none of them exists among the base ontology concepts. Although there is a "Cryptography" class, we would need a subclass named "Cryptographic protocols" under it to achieve correct classification. Lastly, there are better ways of using the TOC, taking into account the locations of the terms in the textbook, to recognize more concepts. This is a topic of future research.

III. CONCLUSION

In this paper, we have developed several bootstrapping strategies to automatically classify the index terms of a textbook into a security ontology. The resulting enriched security ontology will be used for annotating and searching for learning materials in cyber-security education and training. While initial results are encouraging, more work

is needed to increase the coverage of the bootstrapping method.

To increase the quantity of security concepts in the ontology, we will deepen our Wikipedia category extraction to the second and third level by finding categories of categories. We need to find a better way to disambiguate the ambiguous article results. We also intend to define better methods for concept classification and to extract more semantic relationships between concepts to achieve a comprehensive and complete ontology for the security domain. In future research, we will examine term synonyms from WordNet.

We have identified a human security expert who will perform a quantitative evaluation of the correctness of our results. In addition, more efforts will have to be expended on developing a software tool that minimizes the effort and time of the human expert. Finally, this new approach will be applied to a second textbook in security. It is hoped that the methods developed here can be applied to any technical textbook, not just to computer security.

ACKNOWLEDGMENT

This work is partially funded by NSF grant DUE1241687. We gratefully acknowledge Pearson and Addison-Wesley for making the electronic textbooks available. The work by Chun was partially conducted while she was on a sabbatical leave at the Network Security Lab at Columbia University.

REFERENCES

[1] K. Rawlinson and M. Doss, “HP Research: Cybercrime Costs Rise Nearly 40 Percent, Attack Frequency Doubles, Security intelligence solutions key to mitigating impact,” 2012; http://www8.hp.com/us/en/hp-news/press-release.html?id=1303754#.UbkFErvLgQB.

[2] C. Thompson, “New Malware Attacks Smartphone, Computer to Eavesdrop,” 2013; http://www.cnbc.com/id/100431624.

[3] D. Goldman, “Smartphone cyberattacks to grow this year,” 2013; http://money.cnn.com/2013/01/08/technology/security/smartphone-cyberattacks/index.html.

[4] N. F. Noy and D. L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology," Stanford.

[5] W. Xuefeng, D. Pingan, and C. Guangzuo, "Automatic Extraction of Course Ontology from Chinese Textbook," in *Computational Intelligence and Software Engineering (CiSE), 2010 International Conference on*, 2010, pp. 1-4.

[6] Wikipedia, "Index (publishing).

[7] C. Caracciolo, "Designing and Implementing an Ontology for Logic and Linguistics," *Literary & Linguistic Computing,* vol. 21, pp. 29-39, 2006.

[8] D. Hindle, "Noun classification from predicate-argument structures," in *Proceedings of the 28th annual meeting on Association for Computational Linguistics* Pittsburgh, Pennsylvania: Association for Computational Linguistics, 1990.

[9] M. A. Hearst, "Automatic acquisition of hyponyms from large text corpora," in *Proceedings of the 14th conference on Computational linguistics - Volume 2* Nantes, France: Association for Computational Linguistics, 1992.

[10] P. Wiebke, "A Set-Theoretical Approach for the Induction of Inheritance Hierarchies," *Electron Notes Theor Comput Sci,* vol. 53, pp. 13-13, 2004.

[11] P. Cimiano, A. Hotho, and S. Staab, "Learning concept hierarchies from text corpora using formal concept analysis," *J. Artif. Int. Res.,* vol. 24, pp. 305-339, 2005.

[12] P. Jain, P. Hitzler, A. P. Sheth, K. Verma, and P. Z. Yeh, "Ontology alignment for linked open data," in *Proceedings of the 9th international semantic web conference on The semantic web - Volume Part I* Shanghai, China: Springer-Verlag.

[13] Y. An, J. Geller, Y. Wu, and S. Chun, "Automatic Generation of Ontology from the Deep Web," in *Database and Expert Systems Applications, 2007. DEXA '07. 18th International Workshop on*, 2007, pp. 470-474.

[14] K. Netti, "Automatic construction of ontology by exploiting web using Google API and JSON," *JOURNAL OF COMPUTING,* vol. 3, pp. 40-46, 2011.

[15] N. Pattanasri, A. Jatowt, and K. Tanaka, "Context-aware search inside e-learning materials using textbook ontologies," in *Proceedings of the joint 9th Asia-Pacific web and 8th international conference on web-age information management conference on Advances in data and web management* Huang Shan, China: Springer-Verlag, 2007.

[16] G. Vigna, C. Kruegel, E. Jonsson, J. Undercoffer, A. Joshi, and J. Pinkston, "Modeling Computer Attacks: An Ontology for Intrusion Detection," in *Recent Advances in Intrusion Detection*. vol. 2820: Springer Berlin Heidelberg, 2003, pp. 113-135.

[17] M. Bajec, J. Eder, A. Souag, C. Salinesi, and I. Comyn-Wattiau, "Ontologies for Security Requirements: A Literature Survey and Classification," in *Advanced Information Systems Engineering Workshops*. vol. 112: Springer Berlin Heidelberg, pp. 61-69.

[18] D. Geneiatakis and C. Lambrinoudakis, "An ontology description for SIP security flaws," *Comput. Commun.,* vol. 30, pp. 1367-1374, 2007.

[19] A. Herzog, N. Shahmehri, and C. Duma, "An Ontology of Information Security," IGI Global, 2007, pp. 1-23.

[20] S. Fenz and A. Ekelhart, "Formalizing information security knowledge," in *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security* Sydney, Australia: ACM, 2009.

[21] R. Meersman, Z. Tari, A. Kim, J. Luo, and M. Kang, "Security Ontology for Annotating Resources," in *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE*. vol. 3761: Springer Berlin Heidelberg, 2005, pp. 1483-1499.

[22] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing?," *International Journal of Human-Computer Studies,* vol. 43, pp. 907-928, 1995.

[23] NIST, "An Introduction to Computer Security - The NIST Handbook. Technical report, NIST (National Institute of Standards and Technology), ," *Special Publication 800-12,* October 1995.

[24] "BSI. IT Grundschutz Manual.," 2004.

[25] "United Nations. United Nations Standard Products and Services Code," 2006.

[26] D. Fensel, K. Sycara, J. Mylopoulos, G. Denker, L. Kagal, T. Finin, and M. Paolucci, "Security for DAML Web Services: Annotation and Matchmaking," in *The Semantic Web - ISWC 2003*. vol. 2870: Springer Berlin Heidelberg, 2003, pp. 335-350.

[27] C. Blanco, J. Lasheras, R. Valencia-Garcia, E. Fernandez-Medina, A. Toval, and M. Piattini, "A Systematic Review and Comparison of Security Ontologies," in *Proceedings of the 2008 Third International Conference on Availability, Reliability and Security*: IEEE Computer Society, 2008: pp. 813-820.

[28] M. T. Goodrich and R. Tamassia, *Introduction to Computer Security*, 1 ed.: Addison-Wesley, 2010.

[29] E. Skoudis and T. Liston, *Counter Hack Reloaded: A Step-by-Step Guide to Computer Attacks and Effective Defenses*, 2 ed.: Pearson Education, Inc., 2006.

[30] NIST Interagency Report*: Glossary of Key Information Security Terms*, R. Kissel, Ed.: National Institute of Standards and Technology, U.S. Department of Commerce, 2012, p. 222.