Identifying Pairs of Terms with Strong Semantic Connections in a Textbook Index

James Geller[1], Shmuel T. Klein[2] and Yuriy Polyakov[1]
[1]*Department of Computer Science, NJIT, Newark NJ, U.S.A.*
[2]*Department of Computer Science, Bar Ilan University, Ramat Gan, 52900, Israel*

Keywords: Ontology, Semantic Relationships, Textbook Index, Security Concepts, Semantically Correlated Terms.

Abstract: Semantic relationships are important components of ontologies. Specifying these relationships is work-intensive and error-prone when done by experts. Discovering domain concepts and strongly related pairs of concepts in a completely automated way from English text is an unresolved problem. This paper uses index terms from a textbook as domain concepts and suggests pairs of concepts that are likely to be connected by strong semantic relationships. Two textbooks on Cyber Security were used as testbeds. To show the generality of the approach, the index terms from one of the books were used to generate suggestions for where to place semantic relationships using the bodies of *both* textbooks. A good overlap was found.

1 INTRODUCTION

Ontologies are becoming increasingly popular, in a number of research areas, e.g., in Medical Informatics. The NCBO BioPortal (Musen et al., 2012) currently houses 443 ontologies (BioPortal, 2015). However, building them remains a hard problem for any ontology that is of a practically useful size. Many approaches have been reported, e.g., automated methods (Hindle, 1990; Hearst, 1992; Wiebke, 2004, Cimiano et al., 2005), and ontologies built by hand (Caracciolo, 2006).

Different variations and hybrid methods exist. One approach that deserves specific mention is to construct several small ontologies and then use alignment algorithms to combine them into a large one (Jain et al., 2013). Another approach is to leave the ontology building to experts, but to provide them with tools to make it easier (Geller et al., 2014).

Our general approach to ontology construction is to make maximal use of already existing semi-formal sources, as this is easier for the expert than to start with an "empty page," yet likely more successful than trying to extract concepts from completely unstructured text (Wali et al., 2013). In older work (An et al., 2007) we used tables of data hidden in the *Deep Web*, which are hard to access.

Thus, we have turned to another source of semi-formal data for ontology construction, namely textbook indexes. A review of the indexes of several textbooks indicates that their terms provide an excellent starting point for an ontology for the subject domain of each book. We note that our use of the word "concept" includes not only object concepts but a reification of tasks, processes, events, etc. The primary domain of our research is Cyber Security (Sections 4.2−4.6). Thus, encryption is a process, an attack should be viewed as an event, etc.

Many textbooks have sub-index-terms which can be translated into hierarchical relationships between main terms and associated subterms (that need to be validated by an expert). In previous work (Wali et al., 2013), we investigated several methods for finding IS-A relationships between concepts. That is not the topic of this paper.

The harder problem in ontology building is to insert the correct *semantic* (also called *lateral) relationships* into the ontology. These go beyond the basic hierarchical relationships (e.g., subclass, IS-A, kind-of, part-of) and beyond the local attributes of concepts.

In Cyber Security, examples of semantic relationships would be *defends, certifies, detects, etc.* Thus a virus scan program *detects* computer viruses. In an ontology with the concepts virus scan program and computer virus the *detects* relationship would point from the former to the latter. The problem of including semantic relationships in an ontology is not specific to Cyber Security.

Geller, J., Klein, S. and Polyakov, Y..
Identifying Pairs of Terms with Strong Semantic Connections in a Textbook Index.
In *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015) - Volume 2: KEOD*, pages 307-315
ISBN: 978-989-758-158-8

To restate the problem, given a backbone of concepts, connected by IS-A or similar hierarchical links, the question is what semantic relationships should be added to this backbone. Handing the backbone to a human expert and asking her to provide the semantic relationships is not practical.

If a textbook index contains $n = 700$ concepts, there could potentially be a relationship between any pair of those concepts. Given that there are $\binom{n}{2} = \frac{n(n-1)}{2}$ unique pairs of distinct concepts, an expert would need to review 244,650 pairs. Furthermore, many concepts are connected to multiple other concepts by the same or different semantic relationships.

Thus, a method is needed to show to the expert only a very small, well-chosen percentage of all possible concept pairs, such that the concepts of each pair are highly likely to be connected by a meaningful semantic relationship. But how can an algorithm guess this? The answer we suggest is that if two concepts appear often close to each other in the actual *body* of the textbook, then there is a good chance that such a semantic relationship might exist. Note that we are not determining whether it exists or what it might be. This decision will be made by a domain expert. We are only eliminating from the view of the expert the large majority of pairs that are not likely to be semantically connected.

The crucial question is to define what *close to each other in the actual body of the textbook* means. In this paper, we suggest using text units that are semantically meaningful, which, for this investigation, we have chosen as whole sentences. Other possibilities exist, such as paragraphs.

The remainder of the paper is organized as follows. In Section 2 we are reviewing related literature. Section 3 describes our theory and methodology. Section 4 describes preliminary tests and then experiments with two Cyber Security textbooks. Section 5 discusses the results, and Section 6 contains Conclusions and Future Work.

2 RELATED WORK

Building ontologies by hand is difficult and time consuming. For example, Caracciolo (2006) reports on building an ontology called LoLaLi. To avoid these difficulties, many attempts have been made to derive ontologies directly from text. To make this task easier, additional sources of information may be used. Maedche and Staab (2000) combined text mining with the use of a dictionary to create a domain-specific ontology. Our work uses a listing of index terms from a textbook instead of a dictionary. Work on using a book index in ontology building was reported by Pattanasri et al. (2007).

Examples of automated methods for building ontologies are, e.g., Hindle (1990), Hearst (1992), Wiebke (2004) and Cimiano et al. (2005). Hindle's work is based on the clustering approach and a corpus of text. Hearst used a linguistic pattern matching method by identifying a set of lexico-syntactic patterns to find semantic relationships between terms from large corpora of text. Wiebke reported on a set-theoretical approach for creating inheritance hierarchies. Cimiano et al.'s work is based on the use of Formal Concept Analysis.

The application area of this work is the creation of an ontology for Cyber Security. Several previous efforts have gone into building such an ontology. A summary appears in (Geller et al., 2014). However, we mention work by Vigna et al. (2003), Geneiatakis and Lambrinoudakis (2007), Herzog et al. (2007), Fenz and Ekelhart (2009) and Meersman et al. (2005). In previous work (Wali et al., 2013) we extended the ontology of Herzog. In this paper, we are focusing on the issue of existence and strength of semantic (non-taxonomic) relationships between Cyber Security concepts.

In the medical ontology domain, Lee et al. (2004) describe a method for the automated identification of the *treatment* relationship. Katsurai et al. (2014) use tagged images to identify semantic relationships. The use of co-occurrence information in extending ontologies was reported by Novalija et al. (2011). Their method assumes an additional glossary, which our primary book did not provide.

3 METHODOLOGY

The problem we deal with can formally be defined as follows. We are given a large more or less homogeneous text T, for which an index I has been provided. The goal is to generate pairs of index terms (*a,b*) that are most strongly correlated with each other. If an automated way of compiling such a list can be found, it could serve as input to experts working on establishing semantic links among the concepts of the ontology.

The challenge is to find the correct terms, which discriminate between the given semantic contexts; it is thus not sufficient for a pair of terms to co-occur frequently. We suggest to follow the methodology of (Bookstein and Klein, 1990), and to first partition the given text T, for each index term *a*, into two

disjoint subsets, the first, which we shall denote by T_a, and its complement $T_{\bar{a}} = T - T_a$.

The set T_a will be the set of terms appearing in the proximity of the term *a* under consideration, and may thus be considered as those terms most strongly connected with *a*.

There are several ways to define proximity. For example, one could decide to extend the proximity of *a* to the entire sentence in which *a* appears. Another, definition, would be to choose an integer threshold $d_a > 1$, and define the proximity as the set of all the terms within a range of d_a terms before and after any occurrence of *a*. A third possibility would be the intersection of the two preceding choices. The threshold d_a could be chosen fixed, or depending on the global frequency of the term *a* in T, and/or on the size of the current sentence. To avoid the bias introduced by frequent terms, which appear in the context of *a* just because they appear in fact almost everywhere, we define, for each term $b \neq a$, its probability of occurrence in T_a and $T_{\bar{a}}$:

$$f_a(b) = \frac{\mathrm{freq}_a(b) + 1}{|T_a| + 1} \qquad f_{\bar{a}}(b) = \frac{\mathrm{freq}_{\bar{a}}(b) + 1}{|T_{\bar{a}}| + 1}, \quad (1)$$

where $\mathrm{freq}_x(b)$ denotes the frequency of occurrence of the term *b* in the set T_x, $|Y|$ is the size of the set *Y*, and the addition of 1 to numerator and denominator avoids a potential division by 0. As measure of the correlation strength with term *a*, we use the ratio of the above probabilities, multiplied by an increasing function of the frequency:

$$c_a(b) = \log_2(\mathrm{freq}_a(b) + 2)\frac{f_a(b)}{f_{\bar{a}}(b)}, \quad (2)$$

which is well defined, since $f_{\bar{a}}(b) \neq 0$.

The following intuition lies behind this definition. The ratio of the probabilities can be considered as a weight measuring the connection strength of *b* to *a*. However, a term b_1 appearing 3 times in T_a and 7 times in $T_{\bar{a}}$ would have the same ratio as a term b_2 for which these frequencies could be 30 and 70; yet, the evidence for b_2 is stronger, and this should be reflected. This leads to the idea of a multiplicative factor. Taking $\mathrm{freq}_a(b)$ itself would be too large in relation to the quotient, so we use its logarithm, which can be considered proportional to its information content. Adding 2 is a correction, to avoid negative or 0 values.

Note that the notion of correlation defined by the above procedure is not necessarily symmetric: a term *b* might be among the most strongly related to a term *a*, but this does not imply that *a* must also be in the set of those terms most strongly connected to *b*. This is obvious for text strings. The most strongly related term to the phrase once upon a is probably time, but in the opposite direction, there could be many terms that strongly relate to time. In this research, we are operating with concepts as opposed to text strings. However, a similar effect may be observed here. The concept steering wheel is strongly connected to the concept car (and truck, etc.) However, car is strongly related to many other *car part concepts* besides steering wheel.

A similar phenomenon has been noticed by (Choueka et al, 1983), where terms are sorted according to an *attraction factor* for the automatic detection of idioms. It is also similar to the one-sidedness of the Kullback-Leibler (1951) divergence, a well-known metric measuring how far two probability distributions are apart, and can be made symmetrical in a similar way: Define the connection strength of a pair of index terms (*a,b*) as

$$c(a,b) = c_a(b) + c_b(a). \quad (3)$$

The set of pairs of terms (*a,b*) in L^2, with $a \neq b$ is then sorted by non-increasing *c*(*a,b*), and the first *k* elements of the sorted list are presented to the expert. A simple implementation for all the term pairs (*a,b*) in L^2 would thus require a time complexity of $O(|L|^2 \log |L|)$ which may be reduced to $O(|L|^2 \log k)$ by building a maximum-heap according to the values *c*(*a,b*).

While we suggest that a high value of *c*(*a,b*) implies a high correlation between *a* and *b*, we do not claim that the converse implication is also true. Thus, a pair of terms can have a low score *c*(*a,b*) and yet be strongly connected, because we base the score on co-occurrence, which might be indicative of correlation, but is surely not the only criterion. For example, the style of an author may prefer a term over its synonym. While these two terms are correlated, they will rarely appear together. We thus are primarily interested in the top ranked pairs, which we expect to be connected, and make no specific prediction for the lower ranked pairs.

4 EXPERIMENTS

4.1 Preliminary Feasibility Test

To run a preliminary test of these ideas on a generally available text, we chose the King James Version of the Bible, consisting of 23,136 verses. The term *a* has been chosen as house or houses, which together appear 1942 times, in 1637 verses.

At this preliminary stage, we used only $c_a(b)$, that is, this experiment was based on the understanding of the directionality of relationships; moreover, we wanted to assess the strength of using the pure ratio, without the logarithmic term. The context was set to the entire versein which the term occurs. As expected, the terms with highest probabilities are similar for the two sets: the, of, and, house, to, in, And,... for T_a and the, and, of, to, And, in, that,... for $T_{\bar{a}}$. However, sorting the list by the ratio c_a and restricting it to terms occurring at least five times yields the following list.

ratio	*term*
24.43	courts
16.11	wing
15.65	dedicated
14.09	beams
13.73	timber
13.2	build

Note that the first term on the list, courts, appears there because of the expression courts of the house of the LORD; obviously, the term LORD appears just as often in the same verses, but LORD also appears in many other contexts, so it is not typical of a context generated by the term house, while courts does qualify as highly related term.

In a second test, the chosen term was David, a proper name. Details of this experiment are omitted due to space reasons.

4.2 Textbook Preprocessing and Cleaning

Encouraged by the results of the preliminary test, we next applied it to the textbook *Introduction to Computer Security* by Goodrich and Tamassia (2011). We had access to the textbook as a PDF file, however, extensive cleaning was required. The goal of this preprocessing is to create two separate files. One file contains the complete body of the textbook organized into separate sentences. The other file contains an alphabetical listing of all index terms. Due to the creativity of book authors in how they are *structurally* expressing their ideas, and due to the highly technical nature of the content, it is difficult to create a single program or script that will perform the cleaning. Thus, the cleaning process was performed in a combination of manual steps and regular expressions.

4.3 Processing the Index

To indicate the wide variety of ontological and textual issues in the index, the following examples will suffice. Many index terms are followed by index-internal references to other index terms. The index term is sometimes an acronym that is followed by the keyword "see" and the expansion of the acronym. In some cases, the acronym is the target of a "see reference" and not the source. Some internal references are genuine synonyms. Synonym relationships are important elements of ontologies.

It is desirable to extract IS-A relationships from a textbook index to support the creation of the ontology backbone. (This is not the issue in this paper.) A number of terms in the index are followed by subterms, which could be related by IS-A relationships to the main term.

However, the semantics of the relationship between an index term and a subterm is often oblique. Goodrich et al. are "relatively parsimonious and disciplined" in their assignment of subterms in the index. Many of their term-subterm pairs indeed express an IS-A relationship. In other textbooks, their authors included more but less well defined relationships between terms and subterms.

Another issue with subterms is that sometimes the subterm by itself is dependent, which means that it is expected that the subterm is read together with its main term. In other cases the subterm is independent and defines a concept on its own.

A number of issues in processing an index are textual, as opposed to ontological. For example, many of the elements of L are multi-word index terms. Some of those are just lists of juxtaposed words, separated by blanks. However, other terms appear with dashes between words. The complicating factor is that some words have hyphens (-) in the middle of the word because the word is at the right margin of the textbook page and does not fit in its entire length. These hyphens are textually not distinguishable from dashes between words. In a few cases, the authors of the index were not consistent whether a term should be dashed or not. Both formats appeared in the index. Some acronyms look like words. Non-ASCII characters appear in foreign names.

As examples of internal references in the index, we note that Internet Protocol refers to IP. HIPAA refers to Health Insurance Portability and Accountability Act. Homeograph attack appears as synonym of Unicode attack.

The term biometric is followed by the two subterms identification and verification. The

intended reading is the concatenation, as in biometric identification. For the term firewall, three subterms exist: application-layer, stateful, and stateless. The intended reading is now the *reverse* concatenation as in stateless firewall. While humans can easily distinguish between these options, this is not the case for a knowledge-poor algorithm.

The term operating systems has nine subterms, including concepts, kernel and filesystem. Clearly, concepts cannot stand by itself, because the meaning of operating systems concepts is very different from the meaning of concepts. On the other hand, filesystem can stand by itself. The relationship between kernel and operating systems is *not* an IS-A relationship, but a PART-OF.

Returning to the term filesystem, it appears as one word in the index. However, as part of the term Encrypting File System, it becomes a two-word term. The term interface appears in its hyphenated form as inter-<LF><CR>face because of the narrow column it is in, but it does not stand for any kind of face. GOT stands for Global Offset Table. Transforming GOT to lower case would change the meaning to the get() operation. The Merkle-Damgård construction contains a non-ASCII å. As a first approximation we chose to eliminate many such complicating factors from our index, e.g., by using the ASCII equivalent Merkle-Damgard construction. The resulting index contained 760 terms.

4.4 Processing the Body

Working with the body of the text required additional adjustments. As a technical publication, the textbook contains many formulas, some with non-ASCII characters such as π. As the focus was on complete English sentences with a Subject – Verb – [Object] structure, most titles and captions were eliminated. Most numbers do not carry conceptual weight, with a few exceptions such as 3.14159265…, thus numbers were eliminated. Periods appear as sentence end-markers, in abbreviations, in ellipses, etc. Hyphenation/dashes cause similar problems as in the index. The hyphens cannot be automatically removed as they are legitimate in many cases, such as in man-in-the-middle attack. Thus, file-system may become file system or filesystem, but maninthemiddle attack is erroneous, and term inter-face (with inter- at the end of the line) cannot become inter face. Question marks and exclamation points were treated just as periods. URLs were treated like formulas and deleted. Names containing diacritical marks were reduced to ASCII equivalents. U.S. was replaced by US and other similar simplifications were performed. Special characters such as ‘”,;:$*, etc. were eliminated. Code examples, the *micro tables of content* at the beginning of chapters and the exercises and chapter notes at the end were deleted.

Bullet lists caused a considerable processing effort as the authors were remarkably “variable” in using them. In some cases, bullets were followed by one or more complete sentences terminated by periods. Those sentences could be maintained while eliminating the bullet symbols (●). In other cases, bullets were each followed by a head term, a colon, and sentence without a final period. Following our focus on complete sentences, these were deleted. In other cases bullets were followed by incomplete sentences. There were also numbered paragraph lists and many numbered and unnumbered subsection titles. Lastly, we found a few “true editorial mistakes” in the textbook. In the end of this cleaning process, 6019 sentences remained. Each one of those sentences was stored as a single line in an ASCII file.

4.5 Results of Pilot Study on Textbook 1

To investigate the viability of our approach, we performed a pilot study on Goodrich et al. Our algorithm generated a list of pairs of index terms, sorted according to Section 3, still using only the asymmetric $c_a(b)$, but with the logarithmic factor.

To improve the quality of results and as a side effect also the efficiency of computing them, we pruned the index and eliminated any terms that appeared fewer than k times in the body of the textbook for some predetermined constant k. The idea was that we wanted to concentrate on the main domain concepts, whereas terms with frequencies under a certain threshold may rather be sporadic uses by the author of the text. We used $k = 14$.

Three experts in Cyber Security were chosen from the faculty of NJIT. Each one has taught security-related classes and is an active researcher. They were given a randomized list of a total of 102 output pairs of our algorithm from the top, middle and bottom of the result list, i.e., they were given pairs of concepts considered highly connected, weakly connected and “in between” by our algorithm, in randomized order.

The experts were asked to perform the high/medium/low ranking of the strength of connection between pairs of concepts. The experts were permitted to drop any pairs involving terms for which they were not sure of the meaning. To

quantify the results of this experiment we computed the inter-rater reliability using Cohen's Kappa (Cohen, 1960). Cohen's Kappa (κ) measures the agreement between two raters in a classification task, involving a number of mutually exclusive classes.

Table 1: Comparison Program with Experts.

First Evaluator	Second Evaluator	κ
Program	Expert 1	12.02%
Program	Expert 2	-3.04%
Program	Expert 3	11.76%
AVG	PROG	6.91%

Table 2: Comparison between Experts.

First Evaluator	Second Evaluator	κ
Expert 1	Expert 2	9.92%
Expert 1	Expert 3	19.31%
Expert 2	Expert 3	1.60%
AVG	HUMAN	10.28%

Table 1 shows the inter-rater reliability between the algorithm and the human experts in percent. We note that it is legal for Cohen's Kappa to be negative. To put these low agreements into perspective, it is however necessary to compare them with the inter-rater reliability between pairs of human experts. Table 2 shows these results.

Except for Expert 1 and Expert 3, who showed a slightly better agreement (still low at about 20%) the experts' agreements are of the same order of magnitude as the algorithm/expert agreements. Thus, even though the results in Table 1 were disappointing, Table 2 makes it clear that the task of assigning semantic strength of connection to pairs of concepts is difficult for experts as well. An additional analysis of the pilot study results indicated the following phenomena.

1. The program discovered idioms. Thus, the co-occurrence of IP and spoofing (23 times) was primarily due to the fact that IP spoofing is a technical term by itself. We accommodated this insight by collecting statistics of term pairs appearing immediately next to each other.
2. The program discovered terms that frequently co-occur in Cyber Security text, even though they do not have a strong relationship with each other. For example, the terms availability and integrity are not strongly related to each other, but often co-occur in statements of desired features of computer systems.
3. The program discovered pairs of concepts that are closely connected by IS-A relationships, but ranked them low. For example Linux and operating system appeared weakly connected. This is counter-intuitive. We hypothesize that after the fact was made known to the reader that Linux IS-A operating system, it was assumed as known and the more concrete term Linux was used most of the time. Another interpretation is that Linux and operating system are indeed not connected by a *Cyber Security relationship*, but by a "general purpose relationship."
4. The assumption of directionality of pairs was NOT supported relative to human experts.

4.6 Two-Book Study with Symmetric Similarity

To address the insights won in the pilot study, a further study was designed, with the strength of relationship measured by the symmetric distance in equation (3). The second major change was that instead of comparing the judgments of experts with the output of the program, we chose a *second textbook* (Solomon, 2006) *but used the index terms from the first textbook* (Goodrich et al.). We argued that if the algorithm discovers semantic relationships in one textbook then it should discover the same relationships in other textbooks on the same subject.

The results of this study were indeed considerably better. However, the number of output pairs for the second textbook was much smaller than for the first textbook. Thus, Cohen's Kappa could not be used, as it assumes that both "classifiers" see the same input data.

Instead, we divided the output of our algorithm for Goodrich et al.'s book into three equal partitions (top, medium and bottom). The top partition ended up having 57 rows. Similarly, the output of our algorithm for Solomon had a top partition of 13 rows. We then used *Mean Average Precision* to determine the degree of overlap.

Mean Average Precision (MAP) is one of the most widely used measures in Information Retrieval (IR) to measure system effectiveness (Turpin and Scholer, 2006) for ranked lists. MAP provides a single metric to gauge the quality of a ranked list, which is considered as a sequence of retrieved items ordered by relevance. MAP computes the average *precisions*, AP (in the sense of the technical terms used in IR) over a number of queries that a system executes and then derives their arithmetic mean. For our case we are limiting ourselves to a single query.

To calculate the average precisions in each query, the precision at a certain cut-off point in the ranked list is computed, and then all precision values

are averaged. For example, if the cut-off point is the *n*th position in the ranked list, the precisions for item sets: $\{i_1\}$, $\{i_1, i_2\}$, $\{i_1, i_2, i_3\}\ldots\{ i_1, i_2, i_3, \ldots, i_n\}$ will be computed, where i_k is the *k*th item in the ranked list. The average precision is computed with:

$$\mathrm{AP} = \frac{\sum_{k=1}^{n} \left(P(k) \times \mathrm{Rel}(k)\right)}{N} \quad (4)$$

N is the number of correct items, *n* is the number of retrieved items, and *k* is the rank in the sequence of retrieved items. *P*(*k*) is the precision at the cut-off *k* in the list. Rel(*k*) is an indicator function, which =1 if the item at rank *k* is correct, 0 otherwise.

Table 3 shows the top 13 pairs that are given as output by our algorithm for the Solomon textbook. Column 1 contains the row number. Columns 2 and 3 are the two resulting terms. Column 4 is a binary column that indicates whether this pair appears in the first 57 rows of the Goodrich et al. output. As the first four pairs of Solomon all appear in Goodrich et al, the average precision is computed as follows:

$$\mathrm{AP} = (1/1+2/2+3/3+4/4+5/7+6/13)/6 = 0.86 \quad (5)$$

Thus the Average Precision comes out to 86%. While this result is not directly comparable to Cohen's Kappa in the previous study, it is considerably stronger than the results found there. Notably, the first four strongly related pairs of Solomon are contained in the top results from Goodrich et al. Two lines in Table 3 require special consideration. DOS is an acronym for denial of service. Thus, this example indicates synonym discovery by the algorithm. Unix and DES were considered unrelated by our domain expert (YP). Presumably, the cut-off at "13" was chosen too low down in the table to cover only strongly related pairs of concepts.

Table 3: Relationship ranking for Solomon.

	TERM 1	TERM 2	Y/N
1	plaintext	ciphertext	Y
2	biometric	authentication	Y
3	public key	cryptography	Y
4	password	dictionary attack	Y
5	cryptography	RSA	N
6	HTTP	HTML	N
7	spoofing	IP	Y
8	public key	RSA	N
9	signature	certificate	N
10	TCP	IP	N
11	denial of service	DOS	N
12	Unix	DES	N
13	plaintext	one time pad	Y

5 DISCUSSION

This research has gained new insights into the difficulty of defining semantic relationships in an ontology in a way that is agreed to by domain experts. In the pilot study (Section 4.4) experts were specifically *not* asked what the relationship between pairs of terms is. Automatically deriving the relationships between concepts from text is the final goal. This research was limited to asking the question which pairs of concepts are strongly enough connected to even attempt to derive them. Yet the results show that experts widely disagree on this question. However, the results of comparing the relationships derived automatically from two textbooks with *one* index were highly encouraging.

We note that our processing of the body of the textbooks implied a loss of important structural information, due to the attempt to concentrate on whole sentences as units of mental semantic processing. It was observed that there were several cases where an index term appeared in a short subsection title of a textbook. Presumably, the following subsection was primarily about this index term. *However, surprisingly, the index term never appeared in the subsection.*

Many of the ontological and textual problems were "solved by hand." Ideally it should be possible to automate the process to the point that a *universal index processing program* takes an index as input and produces (disconnected) parts of an ontology as output. As noted above, this might be impossible without additional knowledge sources. Thus, an automated program cannot determine from the term "firewall" and the subterm "stateless" whether it is "stateless firewall" or "firewall stateless," only one of which is grammatically correct in English. The *universal index processing program* would need to consult the text of the book to determine the order.

6 CONCLUSIONS AND FUTURE WORK

Deriving semantic relationships automatically from English text is hard. However, using textbooks with well-defined index terms makes some aspects of this task easier. While inter-rater reliability was found to be very low, good results were obtained when working with two textbooks even though *we used*

the index of one of them only applied to the body of both of them. Future work will involve:

1) Applying the relationship discovery algorithm to longer relevant text.
2) Performing symmetric experiments by using the index of Solomon's textbook with the text of Goodrich et al.'s textbook.
3) Integrating the two indexes into one index and repeating the experiments.
4) Running experiments with human subjects asking them to specify *what the relationship is between two strongly connected concepts.*
5) In the intermediate term, implementing the *universal index processing program.*

Overall, we are keeping the longer term goal in mind of finding what the actual relationships are, between the discovered pairs of concepts. Sentences with pairs of strongly connected index terms are likely to contain additional words that are indicative of possible relationships. As noted, cases were observed where an index term appeared in a subsection title but nowhere in the subsection. Thus, one needs to assume that human readers mentally concatenate the index term with one or more sentences of that subsection. This connection needs to be recovered at all levels of the section hierarchy.

ACKNOWLEDGEMENTS

This work is partially funded by NSF grant DUE1241687. We thank Pearson and Addison-Wesley for making their textbooks available, and Dr. Curtmola and Dr. Rohloff, our domain experts.

REFERENCES

An, Y. J., Geller, J., Wu, Y., Chun, S. A.. 2007, Automatic Generation of Ontology from the Deep Web. *Proc. DEXA '07,* Regensburg, Germany.

BioPortal, 2015. http://bioportal.bioontology.org/.

Bookstein A., Klein S.T., 1990. Information Retrieval Tools for Literary Analysis, in *Database and Expert Systems Applications*, edited by A M. Tjoa, Springer Verlag, Vienna 1-7.

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

Choueka Y., Klein S.T., Neuvitz E., 1983. Automatic Retrieval of Frequent Idiomatic and Collocational Expressions in a Large Corpus, *Journal Assoc. Literary and Linguistic Computing*, Vol. 4, 34-38.

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

Cohen J., 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* 20 (1): 37–46. doi: 10.1177/001316446002000104.

Fenz S., Ekelhart A., 2009. Formalizing information security knowledge, in *Proc. of the 4th Int. Symposium on Information, Computer, and Communications Security*, Sydney, Australia: ACM, 183 – 194.

Geller, J., Chun, S., and Wali, A., 2014. A Hybrid Approach to Developing a Cyber Security Ontology. *In: Proc. of the 3rd Int. Conf. on Data Management Technol. and Applicat.*, pp. 377-384, Vienna, Austria.

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

Goodrich M. T., Tamassia R., 2011. *Introduction to Computer Security,* Addison Wesley.

Hearst M. A., 1992. Automatic acquisition of hyponyms from large text corpora, in *Proceedings of the 14th conference on Computational linguistics – Vol. 2* Nantes, France: Assoc. for Computational Linguistics.

Herzog A., Shahmehri N., Duma C.. 2007. An Ontology of Information Security, *Information Security and Privacy*. 1(4), pp. 1-23.

Hindle D., 1990. Noun classification from predicate-argument structures, in *Proc. of the 28th Ann. Meeting of Association for Computational Linguistics* Pittsburgh, Pennsylvania: Ass. for Comp. Linguistics.

Jain P., Hitzler P., Sheth A. P., Verma K., Yeh P. Z., 2013. Ontology alignment for linked open data, in *Proc. of the 9th Int. Conference on the Semantic Web - Volume Part I* Shanghai, China: Springer-Verlag.

Katsurai M., Ogawa T., Haseyama M., 2014. A Cross-Modal Approach for Extracting Semantic Relationships Between Concepts Using Tagged Images. *IEEE Trans. on Multimedia* 16(4):1059-1074.

Kullback S., Leibler R. A., 1951. On information and sufficiency. *Annals of Math. Statistics* 22 (1): 79–86.

Lee C. H., Koo C., Na J. C., 2004. Automatic Identification of Treatment Relations for Medical Ontology Learning: An Exploratory Study. In *Knowledge Organization and the Global Information Society: Proc. of the Eighth International ISKO Conference.* Ergon Verlag, Wurzburg, Germany, pp. 245-250.

Maedche A., Staab S., 2000. Mining Ontologies from Text, in *Knowledge Engineering and Knowledge Management Methods, Models, and Tools, 12th International Conference, EKAW 2000. Lecture Notes in Computer Science, Volume 1937,* pp. 189-202.

Meersman R., Tari Z., Kim A., Luo J., Kang M., 2005. Security Ontology for Annotating Resources, in *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE.* vol. 3761: Springer, pp. 1483-99.

Musen M. A., Noy N.F., Shah N. H., Whetzel P. L., Chute C.G., Story M.A., Smith B., NCBO team, 2012. The

National Center for Biomedical Ontology. *J Am Med Inform Assoc*. Mar-Apr;19(2):190-5.

Novalija I., Mladenić D., Bradeško L., 2011, OntoPlus: Text-driven ontology extension using ontology content, structure and co-occurrence information. Knowledge-Based Systems, 24(8), pp. 1261–1276.

Pattanasri N., Jatowt A., Tanaka K., 2007. Context-aware search inside e-learning materials using textbook ontologies, in *Proc. of the Joint 9th Asia-Pacific Web and 8th Int. Conf. on Web-age Inform. Management*, appeared as: Advances in data and web management, LNCS 4505, Springer-Verlag, pp 658-669.

Salomon D., 2006. *Foundations of Computer Security*, Springer Verlag, London, ISBN 978-1-8462-8193-8.

Turpin A., Scholer F., 2006. User performance versus precision measures for simple search tasks, in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, Seattle, WA, 2006, pp. 11-18.

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

Wali, A., Chun, S. A., Geller, J. 2013. A Bootstrapping Approach for Developing Cyber Security Ontology Using Textbook Index Terms. *Proc. International Conference on Availability, Reliability and Security (ARES),* University of Regensburg, Germany.

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