



University of Nebraska at Omaha
DigitalCommons@UNO

Student Work

11-19-2010

Constructive Ontology Engineering

William L. Sousan

University of Nebraska at Omaha

Follow this and additional works at: <https://digitalcommons.unomaha.edu/studentwork>



Part of the [Computer Sciences Commons](#)

Recommended Citation

Sousan, William L., "Constructive Ontology Engineering" (2010). *Student Work*. 2861.
<https://digitalcommons.unomaha.edu/studentwork/2861>

This Thesis is brought to you for free and open access by DigitalCommons@UNO. It has been accepted for inclusion in Student Work by an authorized administrator of DigitalCommons@UNO. For more information, please contact unodigitalcommons@unomaha.edu.



Constructive Ontology Engineering

By

William L. Sousan

A DISSERTATION

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Doctor of Philosophy

Major: Information Technology

Under the Supervision of Dr. Qiuming Zhu

Omaha, Nebraska

November 19, 2010

Supervisory Committee:

Prithviraj Dasgupta

William Mahoney

Haifeng Guo

Steve From

UMI Number: 3428432

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI 3428432

Copyright 2010 by ProQuest LLC.

All rights reserved. This edition of the work is protected against unauthorized copying under Title 17, United States Code.



ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Constructive Ontology Engineering

William L. Sousan, Ph.D.

University of Nebraska, 2010

Advisor: Dr. Qiuming Zhu

The proliferation of the Semantic Web depends on ontologies for knowledge sharing, semantic annotation, data fusion, and descriptions of data for machine interpretation. However, ontologies are difficult to create and maintain. In addition, their structure and content may vary depending on the application and domain. Several methods described in literature have been used in creating ontologies from various data sources such as structured data in databases or unstructured text found in text documents or HTML documents. Various data mining techniques, natural language processing methods, syntactical analysis, machine learning methods, and other techniques have been used in building ontologies with automated and semi-automated processes. Due to the vast amount of unstructured text and its continued proliferation, the problem of constructing ontologies from text has attracted considerable attention for research. However, the constructed ontologies may be noisy, with missing and incorrect knowledge. Thus ontology construction continues to be a challenging research problem.

The goal of this research is to investigate a new method for guiding a process of extracting and assembling candidate terms into domain specific concepts and relationships. The process is part of an overall semi-automated system for creating ontologies from unstructured text sources and is driven by the user's goals in an

incremental process. The system applies natural language processing techniques and uses a series of syntactical analysis tools for extracting grammatical relations from a list of text terms representing the parts of speech of a sentence. The extraction process focuses on evaluating the subject-predicate-object sequences of the text for potential concept-relation-concept triples to be built into an ontology. Users can guide the system by selecting seedling concept-relation-concept triples to assist building concepts from the extracted domain specific terms. As a result, the ontology building process develops into an incremental one that allows the user to interact with the system, to guide the development of an ontology, and to tailor the ontology for the user's application needs. The main contribution of this work is the implementation and evaluation of a new semi-automated methodology for constructing domain specific ontologies from unstructured text corpus.

Acknowledgements

First, I would like to thank my dissertation advisor Dr. Qiuming Zhu, of whom I have worked with for over 20 years. He has advised and mentored me while earning my Bachelors, Masters, and now my PhD degrees. In addition, he helped keep me on track and remain focused as well as pushed me at times when I got discouraged. I sincerely appreciate all his support and patience, and look forward to continuing to work with him on future research.

Next I would like to thank Dr. Bill Mahoney for his support in reviewing, critiquing, and providing much needed suggestions for my work along the way and for letting me work with him on several intrusion detection research projects. In addition, I would like to thank Dr. Mahoney and NUCIA for sponsoring parts of my research.

Also, I would like to thank Dr. Zhengxin Chen for his support and collaboration on ontology construction for the avian flu domain and especially for sponsoring a conference paper. In addition, he allowed me to give several talks regarding my research to his classes.

I would also like thank Dr. Robin Gandhi for his support and advisement on ontologies in general. His expertise with ontologies was extremely valuable and his reviews of my work and conference and journal papers are much appreciated.

Also, I especially want to thank my wife, Trish Sousan, for managing the household and family needs while I spent nights and weekends working on my dissertation. In addition, I appreciate my three kids, Lauren, Katie, and Ben for their understanding and patience for me while I worked on my dissertation.

I would also like to thank the members of my Dissertation committee, Dr. Prithviraj Dasgupta, Dr. Haifeng Guo, and Dr. Steve From, for taking the time to review my work and provide much needed critique to help improve my dissertation.

In addition, I appreciate the advice and research collaboration regarding Aspect Oriented Programming I received from Dr. Mansour Zand, Dr. Victor Winter, and Dr. Harvey Siy that resulted in a conference paper and presentation at the AOSD conference workshop.

Finally, I would like to thank my work, Technical Support Inc. and the owner, Rudy Chloupek, for supporting me in obtaining my PhD through tuition re-imbursement, sponsoring academic conferences, and for giving the flexibility to attend classes and research meetings.

Table of Contents

| | | |
|-------|--|----|
| 1 | Introduction..... | 1 |
| 1.1 | Ontologies | 3 |
| 1.2 | Ontology Based Information Systems | 6 |
| 1.3 | Ontology creation..... | 8 |
| 2 | Creating Ontologies from Text <i>Corpus</i> | 10 |
| 2.1 | Corpus creation | 11 |
| 2.2 | Extracting potential domain-relevant terms..... | 12 |
| 2.3 | Conceptualization | 13 |
| 2.4 | Conceptual graph based on taxonomic relations. | 13 |
| 2.5 | Use of Patterns | 14 |
| 3 | Semi-Automated Processes- State of the Art..... | 16 |
| 4 | Research Methodology | 21 |
| 4.1 | Identifying relevant problem for solving | 21 |
| 4.2 | Performing a literature review and investigating theoretical foundation..... | 22 |
| 4.3 | Designing and constructing artifact | 22 |
| 4.4 | Demonstrate usability | 23 |
| 4.5 | Showing research contribution | 24 |
| 5 | Techniques | 25 |
| 5.1 | Process summary | 25 |
| 5.2 | Building Text Corpus..... | 29 |
| 5.3 | Parsing Text Corpus..... | 30 |
| 5.4 | Building Phrase and SPO Database | 32 |
| 5.5 | Seed Ontology..... | 35 |
| 5.6 | Semantic relevance between terms | 37 |
| 5.7 | Ranking SPOs | 41 |
| 5.8 | Creating Term Extraction Patterns..... | 43 |
| 5.9 | Building Term Pools | 44 |
| 5.10 | Conceptualizing Term Pools..... | 46 |
| 5.11 | Ontology Meta-Model | 48 |
| 5.12 | Iteratively repeating the process | 49 |
| 5.13 | Term Pool to SPO Ranking Feed-back Loop | 51 |
| 6 | Experimentation, Results, and Discussion..... | 54 |
| 6.1 | Experimentation..... | 54 |
| 6.2 | Analysis of Boot Strap process | 55 |
| 6.3 | Incremental analysis..... | 61 |
| 6.4 | Analysis of Operations..... | 61 |
| 6.4.1 | Seed Ontology analysis..... | 61 |
| 6.4.2 | Semantic Relevance computation | 62 |
| 6.4.3 | Term Extraction Patterns analysis | 63 |
| 6.4.4 | Pattern ranking and domain term Feedback Loop | 65 |
| 6.4.5 | Conceptualization | 66 |
| 6.4.6 | Learning concept hierarchy..... | 66 |
| 6.4.7 | Ontology Meta-Model | 67 |
| 7 | Conclusions and Future Work | 68 |

| | | |
|-----|--------------------------|----|
| 7.1 | Results..... | 68 |
| 7.2 | Main contributions | 70 |
| 7.3 | Future work..... | 71 |
| 8 | References..... | 73 |

List of Figures

| | |
|--|----|
| Figure 1 - Ontology learning layer cake | 2 |
| Figure 2 - Generalized ontology structure | 5 |
| Figure 3 - Example ontology | 6 |
| Figure 4 - Common ontology construction architecture | 11 |
| Figure 5 - Initial ontology construction process | 27 |
| Figure 6 - Ontology extension and refinement process | 28 |
| Figure 7 - Example output from Stanford Parser..... | 31 |
| Figure 8 - Cyber-attack Seed Ontology | 36 |
| Figure 9 - Example of ranked SPOs for user evaluation | 42 |
| Figure 10 - Pseudo code for building TPs | 45 |
| Figure 11 - Ontology Meta-Model..... | 48 |
| Figure 12 - Pseudo code for conceptualization and hierarchy arrangement updating | 51 |
| Figure 13 - Resulting ontology | 57 |
| Figure 14 - Enlarged section of ontology | 58 |

List of Tables

| | |
|---|----|
| Table 1- Sample text segments showing detected SPOs | 34 |
| Table 2 - Description of seed concepts | 36 |
| Table 3 - SPO ranking before using TP terms | 52 |
| Table 4 - SPO ranking after using TP terms | 53 |
| Table 5 - Initial run <i>corpus</i> extraction statistics..... | 55 |
| Table 6 - Initial run TEPs and terms per concept | 55 |
| Table 7 - Resulting terms for Attack-Agent concept..... | 58 |
| Table 8 - Resulting terms for the Attack-Victim concept..... | 59 |
| Table 9 - Resulting terms for the Attack-Consequence concepts | 60 |
| Table 10 - Resulting terms for the Attack-Means concept | 60 |
| Table 11 - Example Term Pool results from TEPs | 64 |

1 Introduction

The ability to build high quality and practically usable ontologies remains an open research problem. Although there have been increasing improvements in various methods and systems, the process known as “ontology creation,” or also as “ontology learning”, or “ontology construction”, is still cumbersome and difficult. Several methods of creating ontologies using automated and semi-automated techniques from various data sources have been described in literature. Due to the vast amount of available unstructured text residing on the Web, there exists a high degree of motivation for using these techniques for the creation of domain specific ontologies [Sousan et al. 2007].

Ontology construction from text sources generally consists of several processes configured within a pipelined architecture where the output of one process is used as the input to another process. Typically, some of these processes include collecting relevant documents into a domain specific *corpus*, detecting and extracting relevant text terms, clustering the terms into groups that identify a concept, determining names of the identified concepts, determining the semantic distance between concepts, and finally hierarchically arranging the concepts based on their taxonomic and semantic relations. In addition, some applications may require richer ontologies that would also need additional processes to extract non-taxonomic relations, attributes, and axioms. Figure 1 outlines the levels of ontology learning from text in the well-known ontology learning layer cake [Buitelaar et al. 2005].

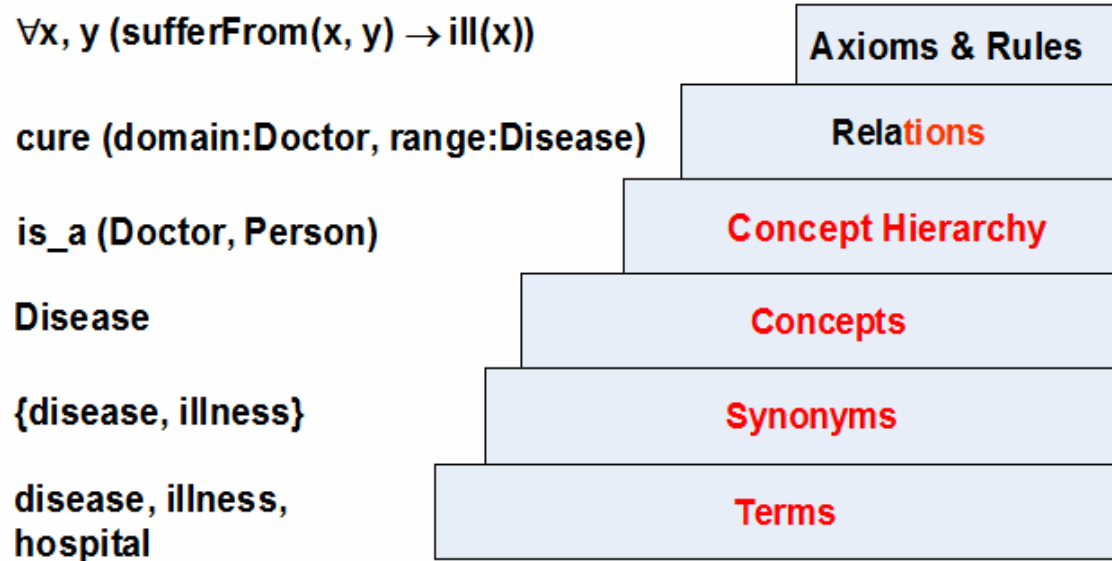


Figure 1 - Ontology learning layer cake

Note that the ontology construction process described in this research work focuses on the lower four layers and part of the fifth layer that consists of “learning terms,” “synonyms,” “concepts,” “concept hierarchies,” and “relations.”

Several problems exist with an entirely automated, unsupervised process of creating ontology from text. In addition to the technical challenges mentioned above, an ontology generated without intensive human intervention is often mixed with noisy, inaccurate, missing, and insufficient concepts and relationships. Furthermore, the ontology resulting from automated processes may lack the necessary properties for the applications that require their accrued knowledge for completing the intended problem-solving tasks. Domain ontologies are also subject to the acceptance from its community of users. Overall, the unsupervised ontology building methods, although they need relatively little or no user assistance, are vulnerable in terms of the algorithm’s ability to find the problem-relevant concepts and mold them correctly into the desired structure.

On the other hand, semi-automated and supervised methods allow for the configuration and adjustment of the ontology construction process which is timely and accurate, but has the obvious drawback of requiring significant user intervention. In addition, noise resulting from the output of any one of the pipelined processes may be propagated up through the pipeline and possibly cause problems with other processes along the way.

These challenges provide the motivation for the research presented in this dissertation. As such, this dissertation focuses on a semi-automated process for ontology construction. Efforts are devoted to the development of a user-feedback guided methodology that attempts to deliver higher quality ontology as compared to fully unsupervised methods, but without the time consuming drawbacks of a fully manual process. The goal is to minimize the needed interactions from the user in guiding and improving the ontology building process.

Thus the research question posed in this work is whether improvements can be made to previous methods of semi-automated ontology construction techniques in order to construct a particular domain-specific ontology. The objective behind this research is to determine useful types of ontology construction parameters that can be defined by the user and used within a formally defined method to construct and incrementally extend a domain specific ontology.

1.1 Ontologies

Ontologies are used for modeling knowledge of a particular domain by using concepts and their relations. These concepts can be real-world entities such as cars, or planes, or abstract things such as emotions, motivation, and others. As Tom Gruber

[Gruber 1993] so eloquently stated, “An Ontology is a formal explicit specification of shared conceptualization”. Thus ontologies provide the capability of formally modeling knowledge with concepts and their semantics that are formally defined together with their corresponding attributes and interrelationships. Ontologies also provide a means for data normalization so that different systems can refer to the same object explicitly and without any ambiguity. Furthermore, the implementation of ontologies in well-defined formal languages allows for the machine readability of ontologies, and thus various applications and software agents can refer to ontologies for a standardized description of things. Moreover, ontologies allow one to infer new information based on implicit information within the structure of the ontology. Overall ontologies are key parts of Semantic Web [Berners-Lee et al. 2001] technologies. They provide the schematics of standardized concepts and their relationships which allow for the automatic linking, processing, and understanding of published data that has been described semantically by an ontology.

Ontologies are made up of several components. They consist of concepts represented by text terms, the concepts themselves, attributes of concepts, relationships between concepts, and axioms that identify constraints amongst the components. Attributes are properties, characteristics, or parameters of concepts such as the color of an object and axioms are constraints on objects such the concept of an airplane could have a rule that airplanes must fly.

In addition, ontologies vary in their structure that depends on the needs of the application that uses the ontology. For example, light-weight ontologies, may simply be a hierarchical taxonomy of concepts that describes the items of discourse within a given domain, which is one of the desired goals of this research. In contrast, more complex

ontologies may contain multi-folds of concepts, relationships, attributes, and constraints/axioms. Figure 2 depicts a generalized structure of an ontology:

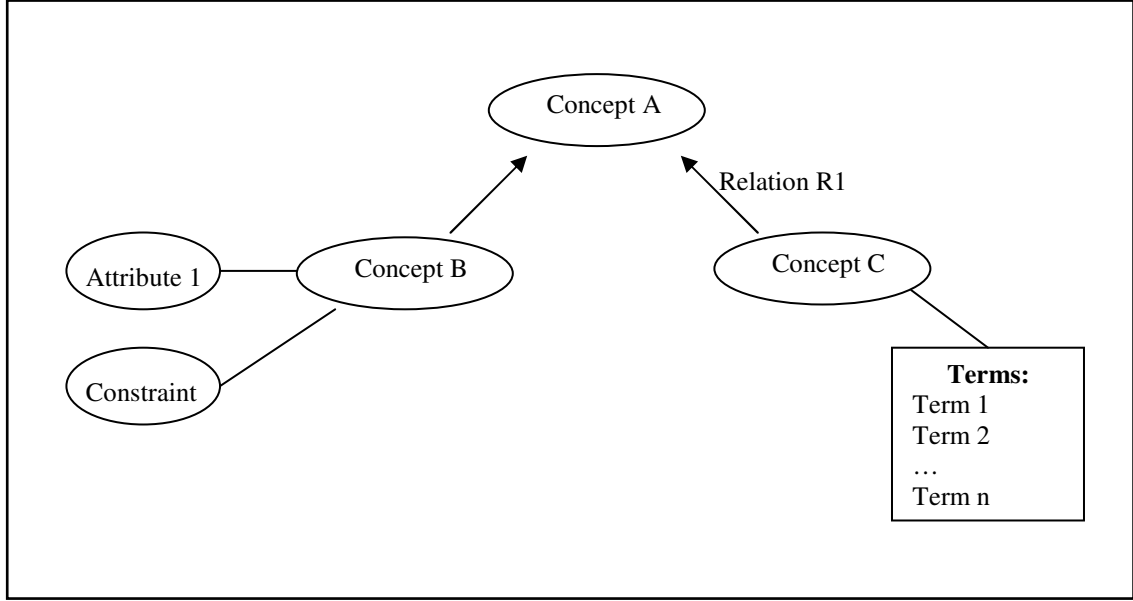


Figure 2 - Generalized ontology structure

The work in this dissertation focuses primarily on terms, concepts, and relationships and their respective hierarchy in building an ontology. The ontological structure is similar to the one used in [Desmontils & Jacquin 2002] that formally describes the ontology as $O(C, R, T)$, where:

- **C:** a set of concepts, $C = \{c_1, c_2, \dots c_n\}$;
- **R:** a set of relationships, $R = \{r_1, r_2, \dots r_n\}$; and
- **T:** a set of terms, $T = \{t_1, t_2, \dots t_n\}$.

A given concept may be represented by a group of terms used to lexically identify the concept and the concepts are arranged in a hierarchical structure based on taxonomic relationships between each concept. Note that a term may consist of one or more words, such as $t_n = w_1 + w_n$.

Figure 3 gives an example of an ontology structure that is used within this research.

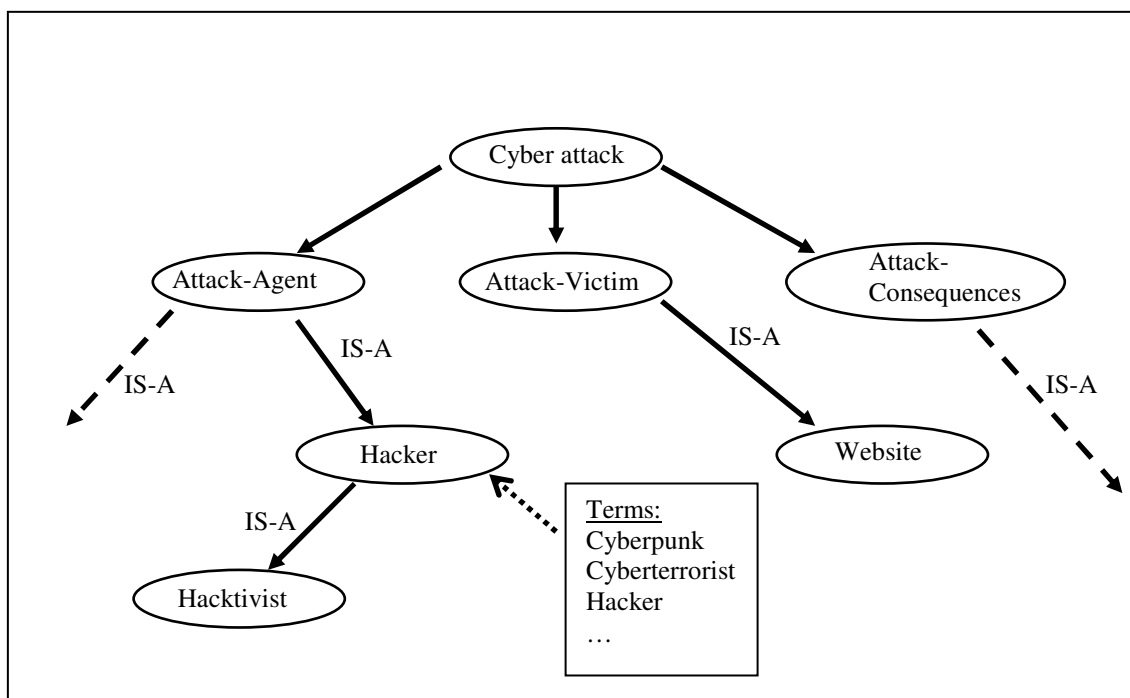


Figure 3 - Example ontology

In figure 3, the ovals represent the concepts; the arrows between the concepts represent the taxonomic relationships between the concepts, where the arrow points to the sub-concept which is more specific than its parent concept. In addition, each concept has a corresponding list of terms that lexically identify the concept. Note that for a given concept, a term associated with the concept and the actual name of the concept itself may be identical.

1.2 Ontology Based Information Systems

Ontology Based Information Systems (OBIS) uses ontologies for knowledge management which provide the solutions to several problems. Ontologies can be used for data fusion purposes, as they can contain synonyms of concepts, or character strings

referred to as terms, that may also be considered as lexical representation of concepts. Therefore different terms that semantically mean the same thing from various documents can be normalized or mapped to the same concept. For example, the terms “whirlybird” and “chopper” may both be used to identify the concept of ‘Helicopter’. Note again that the name of a concept can also be one of its original terms. For example, the concept of ‘Helicopter’ could also have a term of “helicopter”.

In addition, ontologies provide a means for indexing documents based on desired semantic levels and concepts. Documents can have terms annotated that identify concepts within a given ontology. Also, ontologies are a useful means for data retrieval and extraction such that they provide a clear means of query specification for the target information. Ontologies provide a method for semantically annotating text which provides information that can be semantically retrieved by the user’s intent in a WYGIWYN (What you get is what you need) fashion [Sousan et al. 2007]. In addition, they can provide a means for semantically specifying queries in a contextually relevant fashion, versus a simple sequence of keywords or “bag of words” that contain no information regarding the relationships between the words.

There have been several OBIS reported in literature. For example, the ADDminer system [Garcia et al. 2006] performs text mining for creating instances as defined within its ontology. This ontology models the information stored in text-based reports describing incidents occurring on offshore drilling platforms. The Iblogs project from VISTology [Ulicny B. et al. 2007] analyzes online blogs with an ontology for the purposes of the early detection of cyber threats. The Tailored Information Delivery System (TIDS) project [Sousan et al. 2008] uses ontologies for the collection of military intelligence

using ontologies for harvesting information from open source intelligence sources on the Web.

Semantic Web applications rely upon ontologies for their semantic needs for a clear and unambiguous understanding of their data. Therefore there is a need for the construction of high-quality domain specific ontologies to support these applications. In addition, future applications or users may need the ability to generate relatively small ontologies quickly for information retrieval purposes within a narrow domain via an “Ontologies on Demand” [Cimiano et al. 2006] process. These ontologies need to be highly precise and cover only the domain of interest without a lot of irrelevant ontological components.

1.3 Ontology creation

Creating ontologies has been studied from the use of various approaches including manual, semi-automatic, and automatic methods. It has been well documented that ontologies require a great effort in their creation and maintenance, and as such, a considerable amount of research has been dedicated to creating ontology construction systems. When constructing a domain specific ontology, a domain expert and/or domain ontology components are needed as input to the ontology building process. Sources for domain ontology building can come from structured data such as formal databases, semi-structured data such as HTML tables or machine readable dictionaries or unstructured text. The overall objective is to model the desired domain relying on the data sources for its description.

However, ontologies are difficult to build and are extremely labor intensive to create [Gruninger, and Lee, 2002]. There has been a considerable amount of work

performed in the area of ontology construction. For example, there were over 50 systems in 2003 described in [Shamsfard & Barforoush 2003]. In addition, there is no standardized ontology definition or structure and no universally accepted evaluation of quality for a given ontology [Zouaq and Nkambou 2009]. The need for high quality and usable ontologies are often referred to as one of the bottlenecks [Wagner 2006] in the proliferation of the Semantic Web. A considerable amount of research has been performed in the construction of ontologies using automated, semi-automated, and manual processes from various data sources. Current automated and semi-automated methods have some levels of success; however, the resulting ontologies generated from these methods are often noisy with inaccurate, missing, and insufficient concepts and relationships. These generated ontologies may also lack the necessary properties for the applications that require their usage for completing their tasks. In most cases, the generated ontologies may be used as prototype or guide for ontology construction or “cleaned up” by domain experts. As a result, the motivation for creating new methods for ontology construction continues.

As we have seen, the automated methods, although need relatively little or no user assistance, are vulnerable to their algorithm’s ability to find concepts and mold them correctly into the desired ontology. On the other hand, semi-automated methods allow for the configuration and adjustment of the ontology construction process with the obvious drawback of possibly needing too much user input.

2 Creating Ontologies from Text *Corpus*

Current methods of constructing ontologies from text often involve a complex series of processes arranged in a pipeline fashion. Each of these processes has imperfections, and as such, contains a degree of inaccuracy about their outcomes. As a result, much research has been performed in refining and experimenting with various algorithms within these processes. Figure 4 depicts the basic blocks of constructing ontologies from text. These processes are commonly found in systems that attempt to model domains from a given text *corpus*. However, variations exist in literature that adds additional blocks and feedback loops.

For the purposes of analysis of this research work, the ontology construction processes have been classified into four sections; that are 1) *Corpus* creation 2) extracting potential domain-relevant terms 3) Conceptualization 4) Conceptual graph based on taxonomic relations.

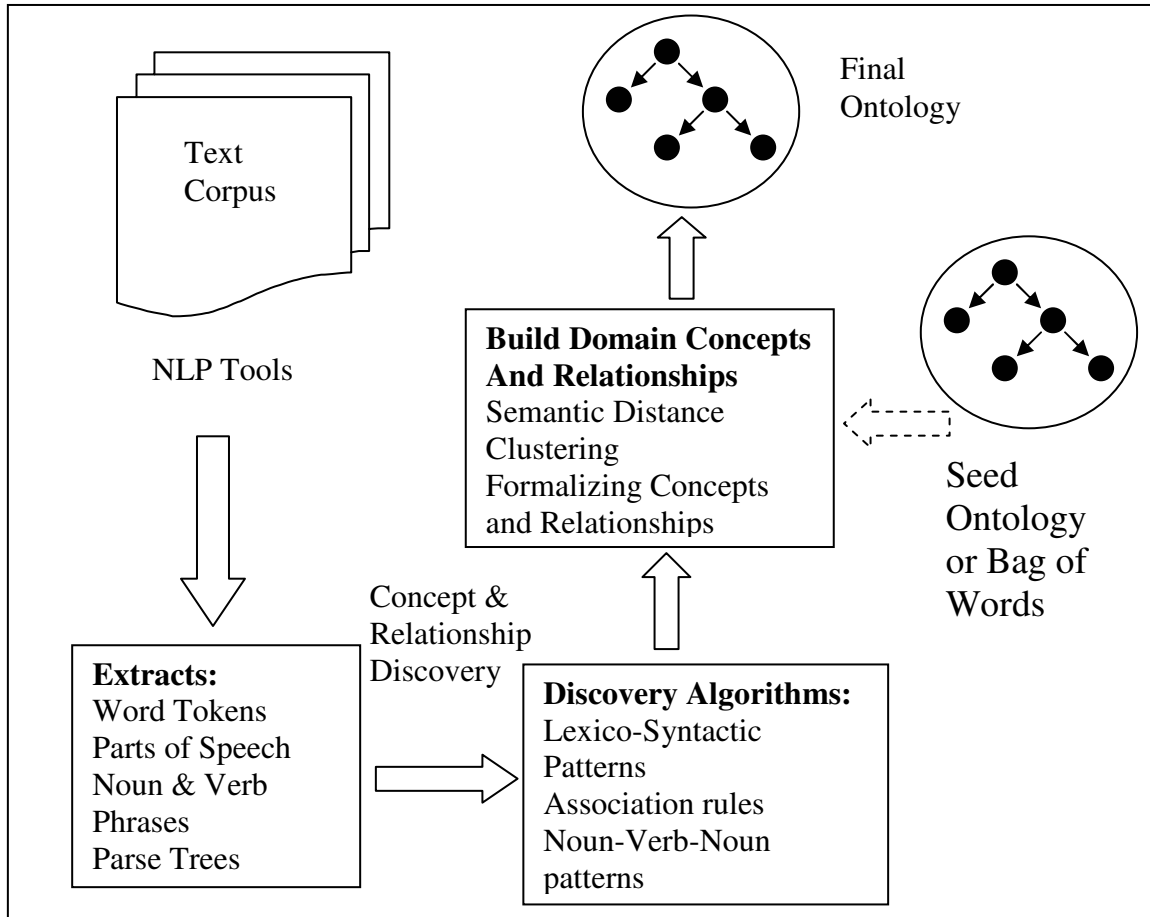


Figure 4 - Common ontology construction architecture

In addition, there does not exist a one-size-fits-all ontology learning process for any domain. There may be ontology building methods that are better for building ontologies for particular domains than others [Zhou 2007]. For example, the authors in [Zhou 2007] recommend that established domain ontologies may be better suited for a top-down learning process as compared to an emerging domain where a bottom-up approach is preferred due to the need to discover new domain knowledge.

2.1 Corpus creation

The initial problem in an ontology construction process is to determine what text articles should be used in the text *corpus*. Note that the better coverage of domain

concepts and relationships, the better the chances are for a well constructed ontology. Some authors have expressed the need for additional supporting data sources from other upper-level ontologies, machine readable dictionaries, and the Web as they feel that text *corpus* may not capture enough of the domain. Furthermore, there are factors regarding whether the document *corpus* should be hand created or retrieved by using keywords or theme extraction. In addition, it may be difficult to justify the sample size in terms of number articles within the text *corpus* needed for a fair evaluation of an ontology construction system.

2.2 Extracting potential domain-relevant terms

Various methods have been described in literatures that are used for finding terms that lexically identify domain relevant concepts and relationships. This process is known as *Terminology Extraction*, and common methods for performing this consists of using statistical, linguistic, or a hybrid of both methods, as described in [Pazienza et al. 2005]. The discovery of candidate concepts and relationships from text are difficult due to the many ways semantics can be expressed and the many different senses of words. Several methods have been tried such as lexico-syntactic patterns [Hearst 1992], noun-verb-noun patterns [Schutz and Buitelaar 2005], association rules [Maedche and Staab 2000], and word frequency.

Terms can be extracted from “scratch” in a sense where, given a *corpus*, the requirement is to mine all the terms that have a high probability of being part of the domain under consideration. This is typically used when sorting a group of documents. However, there may be other types of ontology that need to be more narrowly defined. Thus it may be beneficial for some guidance for identifying the particular domain of the

ontology. Hints may be given to the system in the form of upper-level concepts, an actual skeleton of an ontology, or just a “bag of words”.

2.3 Conceptualization

Once a group of candidate concept and relationship terms have been extracted, it is necessary to group them into similar or identical clusters. Doing so facilitates the task of formally creating concepts, relationships, and their respective synonyms. Thus the problem is how to determine the semantic relevance between terms and what concepts are represented by the terms. Various types of clustering methods have been used to group terms in groups of semantically similar entities. Problems arise with determining the semantic relevance as words can have multiple senses as well as a given concept can be represented with different words. In addition, it may be difficult in the naming of concepts within the clustering process without user intervention. Finally the granularity of the ontology or part of the ontology needs to be constructed appropriately for the given domain.

2.4 Conceptual graph based on taxonomic relations.

Various methods have been reported in literature for hierarchically arranging concepts based on taxonomic relations, which is also known as the learning of concept hierarchies. For example, in [Hearst 1992], the author uses lexico-syntactic patterns, known as “Hearst Patterns”, for discovery of hyponyms from text. As an instance, the follow pattern “NP₀ including NP₁” indicates to look for a sequence of a noun phrase (NP) followed by the word “including” and ending with another noun phrase. In addition, the semantic relationships and concepts need to be defined for the sequence. For this

sequence, NP_0 and NP_1 would be considered concepts related in a hyponym (IS-A) relationship where NP_1 is a hyponym of NP_0 . For instance, the sentence fragment "... all kinds of fish including tuna, halibut..." contains the pattern " NP_0 including NP_1 ", which indicates that tuna is a type of fish. Thus these patterns can be used to identify taxonomic relations between concepts. However, these patterns are reported to have low occurrence rates within test *corpora* [Cimiano et al. 2006]. In order to increase the occurrence rates of the patterns the work in [Cimiano et al. 2004b] uses the Web as *corpus* for matching the Hearst Patterns for finding ontological relations. Note that the work in [Brewster et al. 2002] uses these patterns for the purposes of detecting ISA/hyponymy relations and for detecting new terms too.

Formal Concept Analysis (FCA) [Ganter and Wille 1999] has also been used for determining the hierarchical arrangement of concepts. Using this method requires the determination of concept's characteristics such as attributes and relations to use as a means to determine the semantic similarity as well as the parent-child taxonomic relationships. FCA has been used for learning concept hierarchies in the work of [Cimiano et al. 2005].

2.5 Use of Patterns

Information extraction methods that use linguistic patterns have been described in literature. For example, patterns are used to build a terrorist event dictionary in the Autoslog [Riloff 1993] project. In addition, Subject-Predicate-Object (SPO) triples were used in the EXDISCO system [Yangarger et al. 2000] for information extraction. The authors exploited the use of the triples for finding relevant documents based on their inclusion of designated patterns. The assumption that the subject or object terms of the

SPOs are also viewed as a concept that has multiple terms for lexically representing it. The difference between these systems and the work described in this dissertation is that this work uses the triples for the purposes of extracting domain relevant terms that are later conceptualized. Furthermore, there have been several accounts in literature regarding the use of SPOs for finding candidate non-taxonomic relations. This research uses these sequences for term extraction purposes only.

The use of patterns for ontology learning has been reported to have a high rate of precision but a low rate of recall [Rastegari et al. 2010]. Although there may be several reasons for the low recall rate, two reported reasons are considered the most important. First, the *corpus* may generally have a low amount of instances of predefined patterns. Second, the patterns may have low flexibility in the ability to match new instances. Patterns are typically defined by identifying positive instances and may be selected by such processes as machine learning.

3 Semi-Automated Processes- State of the Art

Semi-Automatic ontology construction, from a high level perspective, relies on the user to provide guidance and feedback in the ontology building process. It is desired that through minimal user feedback and guidance that a higher quality ontology can be constructed as compared to a totally automated process. The user feedback can be provided through various means and used to adjust different parts of the construction process such as the concept discovery, determining semantic distances between concepts, a concept's relationships and attributes, and in the creation of the concept hierarchy. However, if the amount of guidance and feedback required within the process overburdens the knowledge worker, then the process may not be any better than constructing the ontology manually.

Various semi-automated methods for ontology construction have been described in literature. However, the semi-automated portions of these methods vary greatly amongst reported methods and the interpretation of semi-ness. The following examples of semi-automated ontology construction processes are ones that appear similar to the work within this research and are listed in chronological order.

The work in [Brewster et al. 2002] considers their work as a user-guided process for knowledge management. Their system uses a user-defined seed ontology that requires the identification of at least one term, called the seed term, for each seed concept. In addition, the system uses Natural Language Processing (NLP) tools to parse a domain specific text *corpus* looking for Hearst patterns [Hearst 1992] that contains the defined seed terms. Users validate the identified patterns as a positive or negative example, and the system generalizes the patterns and applies them to the entire *corpus* to find related

concepts and relationships. When the user is satisfied with the pattern configurations, they can cleanup the resulting ontology.

Software Application Programming Interface (API) documentation is used as the domain knowledge resource in a semi-automated ontology building process in [Sabou 2004]. Java programming language method headers, which are described with plain text and field descriptions, are used as domain knowledge input into the system. Using NLP tools, the text fields are tokenized into words and the parts-of-speech of each word are identified. Afterwards, verb-noun pairs are identified from the parts-of-speech breakdown. The verb-noun pairs are lemmatized in order to normalize the words, and ranked based on their significance using different ranking schemes. It is then the job of the knowledge engineer to assign a concept to each verb-noun pair as they deem significant – thus provided the semi-automated component of concept assignments. For example, the authors described the assignment of the verb-nouns pairs of “load graph” and “add model” to the concept of *AddOntology*. But the system relies on the knowledge engineer to manually structure the derived concepts into the corresponding hierarchy.

The work described in [Liu et al. 2005] uses a semi-automated method for constructing and updating an ontology on the domain of climate change. Their system utilizes a seed ontology for designating the upper-level domain ontology that is extended and refined via text data mining from the Web. The terms within the seed ontology are used to mine additional terms through term co-occurrence and the semantic relationships are determined by analyzing WordNet [Fellbaum 1998], which is a general purpose ontology. The use of trigger phrases to identify parent-child relationships is reported in [Joho et al. 2004], where weighted links are used to connect the mined terms to the seed

ontology. Concepts are identified by analyzing the terms in WordNet and using disambiguation processes to determine the correct sense. Afterwards, a process known as “Spreading Activation”, similar to neural networks, is used to determine the degree of term relevancy and the term’s placement into the seed ontology. This process iteratively traverses through the network, analyzes the node and link weights to determine node activation, and spreads the activation across related links for the objective of identifying the most relevant keywords. The confirmation of the semantic relationships is performed by again using a combination of the WordNet with the head nouns and subsumption analysis. Terms that can not be automatically confirmed are evaluated by a domain expert or left alone for another iteration of the “Spread Activation” process using additionally acquired evidence. It appears that the semi-automated portions of this process are the construction of the seed ontology and re-running the process against an updated text *corpus*.

For the purposes of building a topic-specific ontology, the OntoGen tool [Fortuna et al. 2006] assists the user by analyzing a *corpus* of plain text documents and recommending potential new topics and providing a visualization of the currently constructed ontology. The authors define the semi-automated component of the process consisting of the user making all the decisions based on computer generated suggestions on topic names and assigning documents to the created topics. Users can edit existing concepts, further expand topics into subtopics, evaluate suggested subtopics for a given topic, and view related topics to the selected topic. Changes to the ontology are reflected within the ontology visualization window. Subtopics of a selected topic are suggested by the system using Latent Semantic Indexing (LSI) or K-means clustering algorithms that

are applied only to the selected topic's documents. LSI [Deerwester et al. 1990] is applied to textual context to determine similar word meanings and K-means clustering [Jain et al. 1999] is used to iteratively partition data into K similar groups.

The authors in [Zhou et al. 2006] use a semi-automated process for the creation of a domain specific ontology of medicine. Their process requires the specification of a core ontology that serves as the upper-level ontology that will be extended. This core ontology consists of seed concepts that are defined by domain experts. The goal is to first extend the core ontology with additional concepts gleaned from WordNet that may require multiple iterations. Second, they process it to further extend the ontology with more concepts and non-taxonomic relations through event based learning. The authors consider events as a triplet defined by $E(C1, V, C2)$ where $C1$ and $C2$ are concepts, or either $C1$ or $C2$ is a frequently occurring noun, V is a verb linking the $C1$ and $C2$. Note that the events described in this work are closely related to the Subject-Predicate-Object sequences described in this dissertation. Events are extracted from the *corpus* using NLP tools and analyzed for the addition of the verb as a relation between two existing concepts within the ontology and/or one of the frequently occurring nouns within the $C1$ or $C2$ slot as a new concept within the ontology. Thus the semi-automated components consist of users guiding the system by defining the core ontology and potentially running multiple iterations of the two processes after evaluating the output of each process.

Other methods allow for users to encode concepts based on a domain's best practices using case based reasoning as in the OntoCase system [Blomqvist 2007]. In [Blomqvist 2008] the author looks for stored patterns in a given group of extracted concepts and relationships. It focuses on the creation and reuse of these patterns along

with their confidence levels in the semi-automatic process of ontology creation in regards to enterprise domain and application ontologies which are denoted by the author as “enterprise application ontology.” These patterns focus on reusing ontology structures and are less focused on text analysis. Furthermore this work builds on the application of existing state of the art ontology learning algorithms for text processing as a front end to their pattern-based method.

4 Research Methodology

The Constructive Research Methodology [Dodig-Crnkovic 2010] was adhered to for conducting the research within this project. The methodology is built upon the following steps:

- a) Finding and identifying a relevant problem to solve
- b) Performing a literature review and investigating the theoretical foundation
- c) Designing and constructing a solution (artifact)
- d) Demonstrate usability
- e) Showing research contribution

4.1 Identifying relevant problem for solving

Ontologies are often referred to as one of the bottlenecks [Wagner 2006] to the proliferation of the Semantic Web and are also used for a variety of knowledge based applications. Various methods of creating ontologies automatically and semi-automatically has been reported in literature and is still a yet to be a fully solved problem. The reported systems are typically very complex pipelined methods with various techniques for solving different parts of the pipelined methods and have varying levels of success. As a result, there exist several areas with the ontology construction systems for research.

4.2 Performing a literature review and investigating theoretical foundation

Several ontology construction systems are reported in literature. In addition, considerable work has been reported on various components that make up the ontology construction process along with any open issues. A key part of this research is the analysis of what characteristics of an ontology construction process can be configured or tailored to successfully construct and refine the ontology. The following research questions need to be answered:

- Are there ways to guide the construction process to tailor it to a specific domain and application such that only includes those concepts and relationships needed for the application and inhibits and removes unwanted concepts and relationships?
- What techniques can be used to reduce the amount of effort needed to create domain specific ontology?
- How to improve the quality of the ontology?

Finally, the objective of the resulting ontology is not meant as a means for semantically indexing a given domain specific *corpus*, but rather for extracting an explicit set of concepts for a specific domain.

4.3 Designing and constructing artifact

In order to test and evaluate the proposed methods, it was necessary to develop a model and implement the model within a software prototype. The model was developed by analyzing various methods and challenges described in literature, reviewing plain text documents on the Web, performing experiments on the types of semantic information

that could be extracted, and finally researching and experimenting on the types of concept characteristics that could be defined by users. Once the model was completed, it was necessary to realize the model by implementing it into a computer program that would perform as an ontology constructing tool kit – thus a software artifact. This artifact would provide a test bed for experimentation and analysis.

The program is written in Java and leverages open source software packages to reduce the implementation efforts. In addition, state-of-the-art algorithms were used for those components not essential to the novelty of the research. Note however that the goal is not develop the entire system – just those components unique to the goals of the research. Algorithms previously reported in literature that fit the goals of the ontology construction process are used. Furthermore several components are needed to construct ontologies from text and therefore a considerable amount of time was spent on implementing the algorithms.

4.4 *Demonstrate usability*

Experiments are needed to collect data for determining the degree of usability. Therefore a domain needs to be selected for the purposes of experimentation that consists of using the ontology constructing software prototype for creating an ontology of the selected domain. For the purposes of this research, the cyber-attack domain was selected and used in experiments for data collection. A *corpus* of hand selected plain text articles was created for input into the system that provided the necessary background knowledge for the cyber-attack domain. Various experiments were defined that exercised the features and new technologies of the system for data collection. However, it is difficult

to evaluate constructed ontologies as there are currently no universally accepted standards to judge the quality of an ontology.

4.5 Showing research contribution

The final task of the constructive research methodology is to identify those components that contribute to the corresponding research area. The research performed within this dissertation contains several new methods that are considered as contributions and are outlined at the end of the report.

5 Techniques

To test and evaluate the proposed methods, it was necessary to implement them within a prototype system. Note that, as stated before, constructing ontologies is a complex process that typically involves several tasks executed in a pipelined fashion. Also note that some of these processes have been used in previous research projects while others are unique to this research. In an effort to clearly describe the ontology construction and iterative process, all the pipelined tasks are identified with those that are unique to this research. There are two major processes within the ontology constructing process, first the process of building the initial ontology and then the process of incrementally extending the ontology. The following sections describe the processes used within the ontology building system.

5.1 *Process summary*

Typical of other systems, the semi-automatic ontology construction system of this research consisted of a relatively complex pipelined process that contains a mixture of custom developed code in conjunction with various open source packages. Several steps are needed within the process for developing and iteratively refining the ontology. The basic steps of the initial process to build the first ontology are the following:

- 1) Create text *corpus* by manually accumulating various unstructured text articles from the Web into a text *corpus* that supplies ontological knowledge regarding the domain of interest
- 2) Build parsed *corpus* by using NLP tools and parsing each article into an Extensible Markup Language (XML) file which contains a breakdown of the

article into sentences, words, parts of speech, noun and verb phrases, and type dependency lists

- 3) Create phrase and Subject-Predicate-Object (SPO) databases by lemmatizing all the words within the phrases to the root words and extract noun phrases, verb phrases, and build SPO triples
- 4) Build a seed ontology by allowing the user to manually create seed concepts and defining at least one term to identify its corresponding seed concept
- 5) Display lists of SPOs to the user that are ranked in ascending order based on the semantic relevance between a given seed concept and either the SPO's subject or object term
- 6) Allow the user to select SPOs for a given seed concept that will be used as a Term Extraction Pattern (TEP) for identifying domain-specific terms. Thus the user builds a list of TEPs for each seed concept.
- 7) Use the TEPs for extracting domain specific terms from the text *corpus*. Also use the patterns for extracting terms in future articles retrieved from the Web.
- 8) Conceptualize the terms within each term pools into clusters of similar terms that describe a concept. Then hierarchically arrange the terms into an ontology.

Figure 5 depicts the initial process.

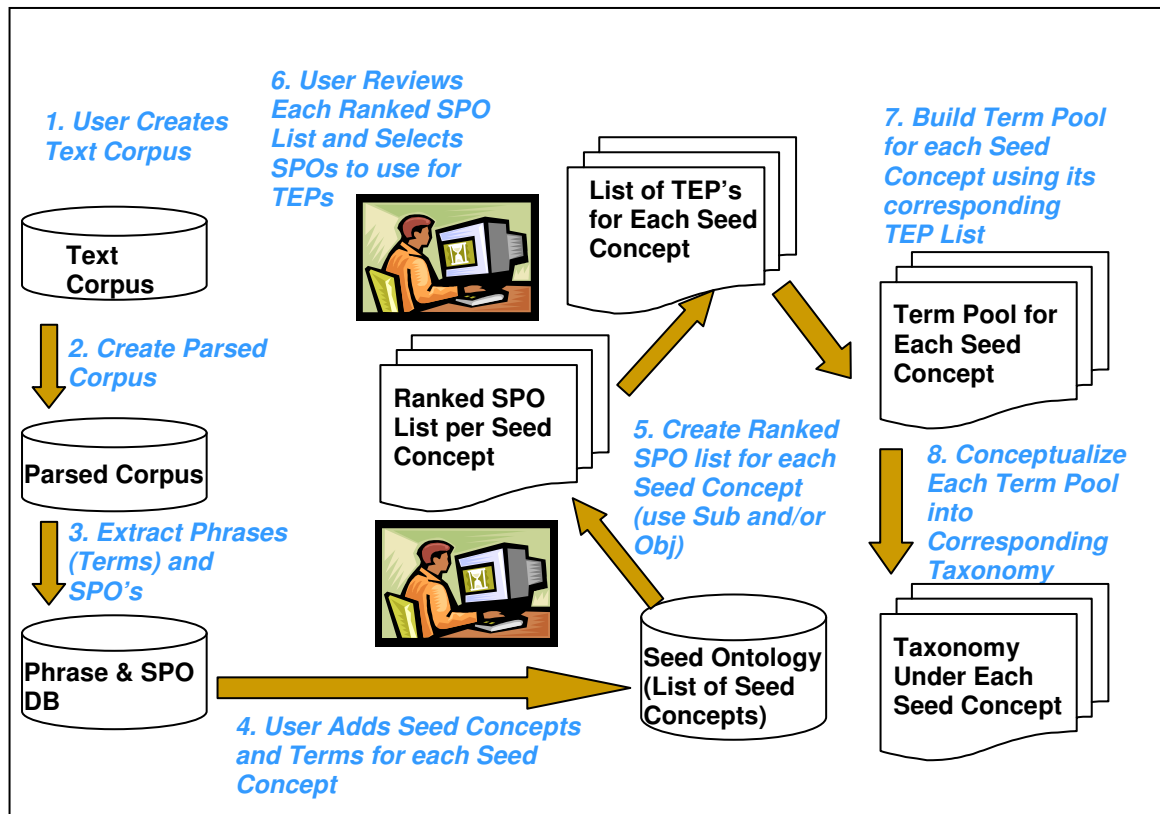


Figure 5 - Initial ontology construction process

Note that steps 1, 2, 3, 4, and step 8 are typical steps that have been implemented in similar work using various algorithms such as in [Zouaq and Nkambou 2009], [Sabou 2004], [Zhou et al. 2006], and others. However, steps 5, 6, and 7 are considered to have new contributions in them as reported in this research.

After the initial ontology has been created, the user can then update the text *corpus* by adding new articles and re-running parts of the process in order to extend and refine the ontology. This is accomplished by the following steps:

- 1) Add text articles to the *corpus*
- 2) Parse added articles only
- 3) Extract noun and verb phrases and SPOs from newly added articles only, and add these to the Phrase and SPO database

- 4) User can review an updated ranked SPO list for each concept. In addition, the terms from the term pool are used to improve the ranking process. Thus SPOs containing Term Pool (TP) terms are pushed higher into the rankings
- 5) Optionally, the user can selected additional SPOs for new Term Extraction Patterns (TEPs) or evaluate existing TEPs for low performance and possibly remove them
- 6) Update each TP based on the current set of TEPs
- 7) Add new terms that were added to the term pools to the existing ontology.

Note that parts of the ontology may be restructured due to adding new terms

Figure 6 depicts the ontology extension and refinement process:

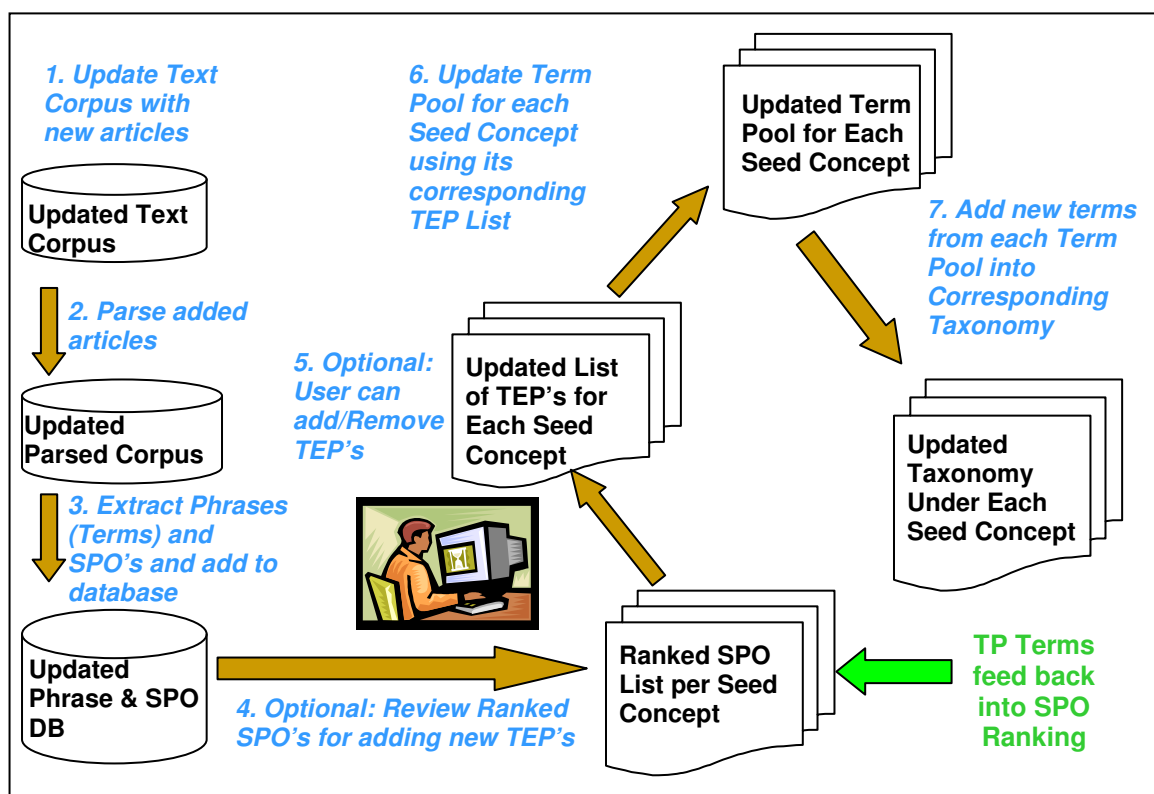


Figure 6 - Ontology extension and refinement process

Note that the above ontology extension and refinement process can be performed periodically as needed to update the resulting ontology. In addition, note that steps 4, 5, and 6 are considered to have new contributions in them reported in this research.

5.2 Building Text Corpus

The text *corpus* forms the foundation of domain-relative terms that may correspond to domain concepts and relations. Thus it is critical that the selection of text articles placed in the *corpus* be articles that represent text passages that contain domain-related information. For the purposes of this research, the articles were hand selected from the Internet.

This text *corpus* was iteratively expanded with additional text articles to simulate the process of adding new domain relative information to the *corpus* for the purposes of updating the ontology with new concepts and relationships. Thus the objective is to use the system in an ongoing fashion, so that the ontology would continue to be updated over time to express the changes and new information reported in the text articles.

The domain of cyber-attacks was chosen to help solve some of the challenges of cyber security for the purposes of determining the probability of an impending cyber-attack due to events that occur across the world. These events are classified based on Social, Political, Economical, and Cultural (SPEC) disturbances in the physical world [Gandhi et al. 2010]. Within the SPEC events, certain types of attacks may occur that consists of various types of characteristics. These attacks may be motivated by amateur hacking, insider retaliations, psychopathic obsessions, social protests, personal gain, commercial competitiveness, organized crime, terrorism, and national/international interests. Furthermore, each attack contains certain attributes such as the type of attacker,

the means of the attack, the consequences of the attack, the victims of the attack, and others. By collecting and analyzing historical data found on the Web within reported events, it may be possible to determine the probability of a future attack based on SPEC events as well as determining the corresponding attack characteristics. The characteristics described in SPEC events provide the motivation for this research to develop an ontology to model the characteristics within SPEC events.

5.3 *Parsing Text Corpus*

In order to analyze noun and verb phrases and their respective relationships, the text articles need to be parsed into words, their respective parts of speech, and grammatical relationships. To accomplish this, open source Natural Language Processing (NLP) tools were used. This project used the Stanford NLP tools [Klein and Manning 2003] that parse English language sentences into various types of formats. This parser was chosen due to its recommendation at related conferences and to its continual development efforts. As of this writing, version 1.6.4 was released on 8/20/2010.

The format used in this research, known as the “typed dependency representation”, identifies the grammatical relations between words and their parts of speech. Figure 7 is a sample of the output from the parsing of the following sentence: “A cyber attack disrupted access to several popular Web sites Tuesday morning, including Yahoo, Google, Microsoft, and Apple”

```

det(attack-3, A-1)
nn(attack-3, cyber-2)
nsubj(disrupted-4, attack-3)
doobj(disrupted-4, access-5)
amod(sites-10, several-7)
amod(sites-10, popular-8)
nn(sites-10, Web-9)
prep_to(disrupted-4, sites-10)
nn(morning-12, Tuesday-11)
tmod(disrupted-4, morning-12)
nn(Microsoft-16, Yahoo-14)
nn(Microsoft-16, Google-15)
prep_including(disrupted-4, Microsoft-16)
conj_and(Microsoft-16, Apple-18)

```

Figure 7 - Example output from Stanford Parser

Note that each line contains a dependency relation such that the first string identifies the grammatical relation between the pair of words separated by a comma within the parenthesis. The number after the dash at the end of each word identifies its sequence within the sentence. For example, *nsubj* (nominal subject), identifies a noun-verb type of relation between two words where the right side word in the parentheses is the noun/object and left side word is the verb/predicate. So the following relation

nsubj (disrupted-4, attack-3)

signifies a noun-verb relation between the words “attack” and “disrupted” where the word “attack” is the third word in the sentence and the word “disrupted” is the fourth word in the sentence. The complete list of relations is described in [Marneffe et al. 2006].

Each article stored in the *corpus* is parsed and its output is stored into an XML file with the same name as the original article. The new XML file has an .xml file extension appended to the end of the file name of the original article file.

5.4 Building Phrase and SPO Database

Each parsed article has its corresponding typed dependency output list scanned for noun phrases, verb phrases, and Subject-Predicate-Object sequences. Similar work on the analysis of the typed dependency has been performed in [Zouaq and Nkambou 2009], for the purposes of extracting subject-verb-object triples by transforming grammatical structures into semantic ones. The work done in this dissertation also uses grammatical structures for extracting noun phrases, verb phrases, and subject-predicate-object triples. The extraction process is done by searching for specific grammatical relation types and then finding connecting words to extract the phrases and SPO sequences.

The process first scans the typed dependency list for the purposes of extracting noun phrases. This is done by looking for the specific grammatical relations of “nn” (noun compound modifier) and “amod” (adjectival modifier). The “nn” relation identifies a pair of connecting nouns, and the “amod” relation identifies an adjective of a noun. Thus a noun phrase can be constructed by scanning the typed dependency list for these sequences.

After the first scan that creates the noun phrases, a second scan is performed for finding subject-verb-object sequences. This is done by looking for sequence of “nsubj” following by a “dobj” (direct object). Thus these two grammatical relations identify a subject-verb-object triple. Note however that other grammatical relations may also identify subject-verb-object relations but this work only focuses on the nsubj-dobj pairs.

Also note that the process creates a list of verbs that are found within these sequences, but only the verb is stored without any connecting relations such as adverbs.

These phrases and SPO sequences are then lemmatized (words converted to their lemma or base form), filtered by a stop-word list, and then stored within a database for use later in the system. Also, the number of occurrences within the text *corpus* of each phrase is tracked for statistical purposes. The phrase and SPO database is extended as new articles are parsed and their corresponding phrases and SPOs are appended to the database.

Note that the parts of speech and their relations can express potential concepts and relationships and as well as attributes. Noun Phrases (NP) are potentially lexical identifiers of corresponding concepts, and Verb Phrases (VP) are potential identifiers of relations between two concepts. Each NP and VP can be viewed as a term. A term in this sense is a lexical indicator of a concept or relationship. For example, the term “chopper” may be the lexical indicator of the concept ‘Helicopter’. Sometimes a term may be the same character string or sub-string as the concept name, but this is not always the case. In addition, terms may consist of multiple words and as such, need to be evaluated to determine if all the words should remain in the term, or if there are intra-term relationships between the words. The breakdown of a multi-word noun phrase can potentially produce taxonomic relationships. Noun phrases that consist of multiple words may also contain multiple nouns and adjectives as well. For example, the sentence fragment of “U.S. soldiers continue to come under attack” has as its first noun phrase or subject the multiword term “U.S. soldiers”. This term consists of two words that are each nouns in that what could be interpreted as “U.S. soldiers” are a subclass or IS-A relation

to the concept of ‘Soldiers’. Research using these methods has been performed in [Buitelaar et al. 2004]. However, this is not always the case and may not be the desired goal of the domain being modeled.

Patterns of NP-VP-VP can be examples of semantic information described in Subject-Predicate-Object (SPO) sequences that represent potential concept-relationship-concept triples. These triples are formally stated as $c_1-r_1-c_2$ where c_1 is a member of the domain of relationship r_1 and c_2 is a member of the range of r_1 . Thus each triple provides the potential for supplying two concepts and a relationship between them for the ontology construction. Previous work in [Schutz and Buitelaar 2005], [Villaverde et al. 2009], and others have been performed in extracting relationships from SPO sequences.

A sample of text segments used for extracting SPO sequences is shown in the Table 1. These articles are indicative of the types of articles accumulated within the *corpus*.

Table 1- Sample text segments showing detected SPOs

| |
|--|
| <p>Muslim hackers hit 3,000 Danish Web sites NCHRO - Feb 22, 2006</p> <p>http://www.nchro.org/public_html/index.php?option=com_content&view=article&id=1653:muslim-hackers-hit-3000-danish-websites&catid=57:press&Itemid=37</p> <p><i>Muslim hackers</i> angered by the publication of cartoons of the Prophet Mohammed have <u>defaced</u> nearly 3,000 <u>Danish Web sites</u> over the past month in the biggest politically motivated cyber attack long-time observers have ever seen.</p> <p>Experts say that the world-wide protests over a Danish newspaper's decision to publish the caricatures ...</p> |
| <p>Hactivism: An Emerging Threat to Diplomacy - Dorothy E. Denning – Sept 2000</p> <p>http://www.afsa.org/fsj/sept00/Denning.cfm</p> <p>... <u>Hactivists</u> have also <u>defaced Web sites</u> belonging to the U.S. embassies in Belgium and in Bosnia-Herzegovina. Doctor Nuker, a founder of the Pakistan Hackerz Club, claimed credit for the attacks and posted images with messages "Stop the Indians" and "Save Kashmir." In these cases, it was obvious to any observer that the defacements were the work of hackers, ...</p> |

China blames US cyber attack for Iran unrest – News.scotsman.com – Jan 25, 2010

<http://news.scotsman.com/world/China-blames-US-cyber-attack.6009653.jp>

Zhou mentioned an outage suffered by Chinese search engine Baidu on 12 January, but did not mention that it was attacked by the Iranian Cyber Army, which had previously attacked Twitter, nor that Chinese hackers launched retaliatory attacks on Iranian sites the next day.

A Brief History of Cybercrime – Time – June 1, 2009

<http://www.time.com/time/nation/article/0,8599,1902073,00.html>

... A 15-year-old Canadian with the handle "mafiaboy" launched the first documented DoS attack in 2000, against numerous e-commerce sites, including eBay and Amazon.com, shutting some down and wreaking havoc that cost an estimated \$1.7 billion. In 2007, entities believed to have been associated with the Russian government or its allies launched a DoS attack against ...

5.5 Seed Ontology

In order to better define the needed groups of concepts to be extracted from the domain of interest, a seed ontology construct is used. The seed ontology allows for the user to restrict the concept groups to the particular ones that apply to the domain ontology. This helps to prevent the influence of popular terms not related to the domain. These seed concepts represent the core concepts within the domain specific ontology and provide the scaffolding for the ontology structure. The user is required to define at least one term for each seed concept that is used as a lexical identifier that represents the seed concept. These terms will later be used to extract domain-specific terms by helping to rank term extraction patterns for the user to select from in order to define term extraction patterns for a given seed concept.

Seed ontologies have been used before as in the work of [Liu et al. 2005] in which a seed ontology was used to create an ontology for climate change that consisted of concepts used for global warming”, “nuclear winter”, “greenhouse gas” and others. Similarly, the work described in [Brewster et al. 2002] uses a user-defined seed ontology with corresponding seed terms for mining Hearst Patterns. These Hearst patterns, which

are used to find taxonomic relations, are rated by the user as positive or negative examples and then used to form general patterns for the purposes of finding new ones for the same taxonomic relation.

The seed ontology used in this research consisted of the following concepts that were developed and described in the work of [Gandhi et al. 2010] as shown in Figure 8:

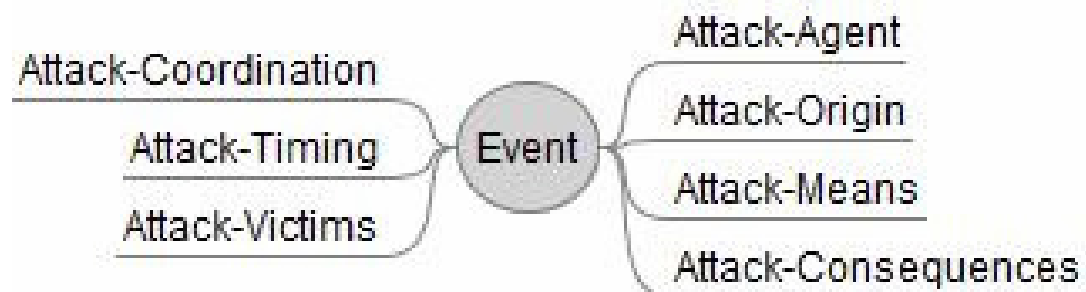


Figure 8 - Cyber-attack Seed Ontology

Note that in the seed ontology the attack-motives concept as described in [Gandhi et al. 2010] is not used. Table 2 describes the concepts from Figure 8:

Table 2 - Description of seed concepts

| Seed Concept | Description |
|---------------------|---|
| Attack-Agent | Type of attacker such as hacker, hacktivist, mercenaries, etc... |
| Attack-Coordination | How the attack was organized, un-organized, chat-room organized, cybermilitia, cyber-vigilante, etc... |
| Attack-Origin | The origin of the attack such as malware victims, malicious agents, etc... |
| Attack-Timing | The timing of the attack such as immediately following an attack, in parallel with an attack, timed activation of planted malware, etc... |
| Attack-Means | The means of the attack such as Denial of Service, spread of malware, SQL or code injections, etc.... |
| Attack-Victims | The victims of the attack such as military, government, businesses, individuals/civilians, and others. |
| Attack-Consequences | The results of the attack, financial loss, information loss, mass panic, etc... |

The program allows the user to define as many seed concepts as they need. Within the definition of each seed concept, the user is required to enter at least one term that will be used later for finding relevant SPOs. The user can also specify a specific word-sense within WordNet [Fellbaum 1998] for each seed concept's term. This helps to improve the ranking process by reducing the amount of word-sense ambiguity problems during the ranking of the SPOs as WordNet is used in the semantic relevance computations.

5.6 Semantic relevance between terms

Several processes within the program require the need for determining the semantic relevance between terms as well as determining the concept a given term represents. Various methods of determining the semantic relevance between terms have been reported in literature. Although the ability to determine semantic relevance was needed for several tasks in the ontology construction process, it was not the focus of this research. As such, the method of using a general purpose ontology was selected along with a corresponding method of using the ontology's concepts and terms to determine semantic relevance. Thus in this research, the WordNet [Fellbaum 1998] was selected along with using the open source package WordNet Similarity Package [Pedersen et al. 2004], which implements four different semantic relevance algorithms. From this set of algorithms, the Lin's [Lin 1998] method for semantic similarity was selected as a result of its evaluation in [Budanitsky and Hirst 2001]. Lin's method provides a degree of semantic evidence that two terms may represent similar concepts; the range is from 0.0 to 1.0 where 1.0 is the highest degree of similarity. In addition, Lin's method is based on the idea of using the Least Common Subsumer (LCS) [Colucci et al. 2008] and the

Information Content (IC) [Resnik 1995] of a concept. The LCS is the most specific concept that subsumes a group of concepts (i.e. most specific superclass). The IC of a concept is the inverse of the probability of encountering an instance of the concept. That is the more generalized a concept is the less information it will contain. Note that the use of IC is an improvement over using the simple counting of link distances due to problems with taxonomies not representing uniform distances between parent/child relationships. Also note that Lin's algorithm is not specific to WordNet and is a generalized method that can be applied to any given taxonomy. The variation of Lin's method used in the WordNet similarity package is:

$$2 * IC(LCS) / (IC(synset1) + IC(synset2)).$$

Where:

IC = Information Content

LCS = Least Common Subsumer

synset = WordNet synset

Note that a WordNet synset is basically group of synonyms that define a concept. The IC is computed based on taking the $-\log$ of the probability of a concept/term appearing in a *corpus*. The probability used in the WordNet Similarity package is the frequency of a term divided by the total number of terms. In addition, the probabilities are computed based on a group of text *corpora*. However, the implementation will return a zero value for an IC that can not be computed based on a missing term frequency value resulting from the term not being found in any of the *corpora*. As a result, the semantic relevance may not be computable for all terms found in WordNet.

To enhance the semantic relevance computation, a method was added in order to determine the taxonomic direction between two words as defined in WordNet. This was needed to determine the direction of the semantic similarity between two terms so that it would be indicated if one of the semantically similar pair of terms is more generalized or specialized than the other. The taxonomic direction is indicated by adding a positive or negative sign to indicate the direction of generalization to specialization. For example, if the two terms “vehicle” and “truck” are analyzed with the above method, a semantic relevance value of +0.7858 is returned indicating the magnitude of relevance. Since the value is positive, it indicates the term “truck” is more specific than the term “vehicle”. The following formula, which is referred to as *semrel*, is used for determining the semantic relevance between two terms t_1 and t_2 :

$$simrel(t_1, t_2) = lins(t_1, t_2) * wpathdir(t_1, t_2) \quad (1)$$

Where:

- $simrel(t_1, t_2)$ is the semantic relevance between t_1 and t_2
- $lins(t_1, t_2)$ is the semantic relevance between terms t_1 and t_2 using Lin’s algorithm within the WordNet taxonomy using the first sense of both terms
- $wpathdir(t_1, t_2)$ is the hyponym/hypernym relation between t_1 and t_2 . If t_2 is lower than or equal to t_1 within the WordNet taxonomy (more specific), then a positive value of 1.0 is returned, else -1.0 is returned to indicate t_2 is more general or abstract than t_1 .

Within the WordNet structure, terms can have multiple senses and therefore problems exist with determining the related concept or WordNet synset of a given term from a text passage without analyzing the context of the term. This causes problems with

word sense ambiguity when the *semrel()* formula is used for computing semantic relevance. In order to reduce the negative affects with word-sense ambiguity, some heuristics are used. Normally, if both terms have multiple senses, then the most common sense of each term is used in the semantic relevance computation. However, if either term has a single sense, then the semantic relevance computation computes the relevance for all combination of senses and selects the highest relevance computation. By doing so, it is assumed that due to the fact that the *corpus* is domain relevant, that if the sense of one term is known, there is a high probability that the closest sense of the term being compared is probably the correct sense. This same heuristic is used if one of the terms has a designated sense. WordNet senses can be designated when the user defines seed terms.

Note that the work in this research does not modify Lin's algorithm in the WordNet Similarity Package. Lin's method is simply used as the initial foundation and the word senses that are used as inputs to Lin's method are designated through using the above word sense heuristics.

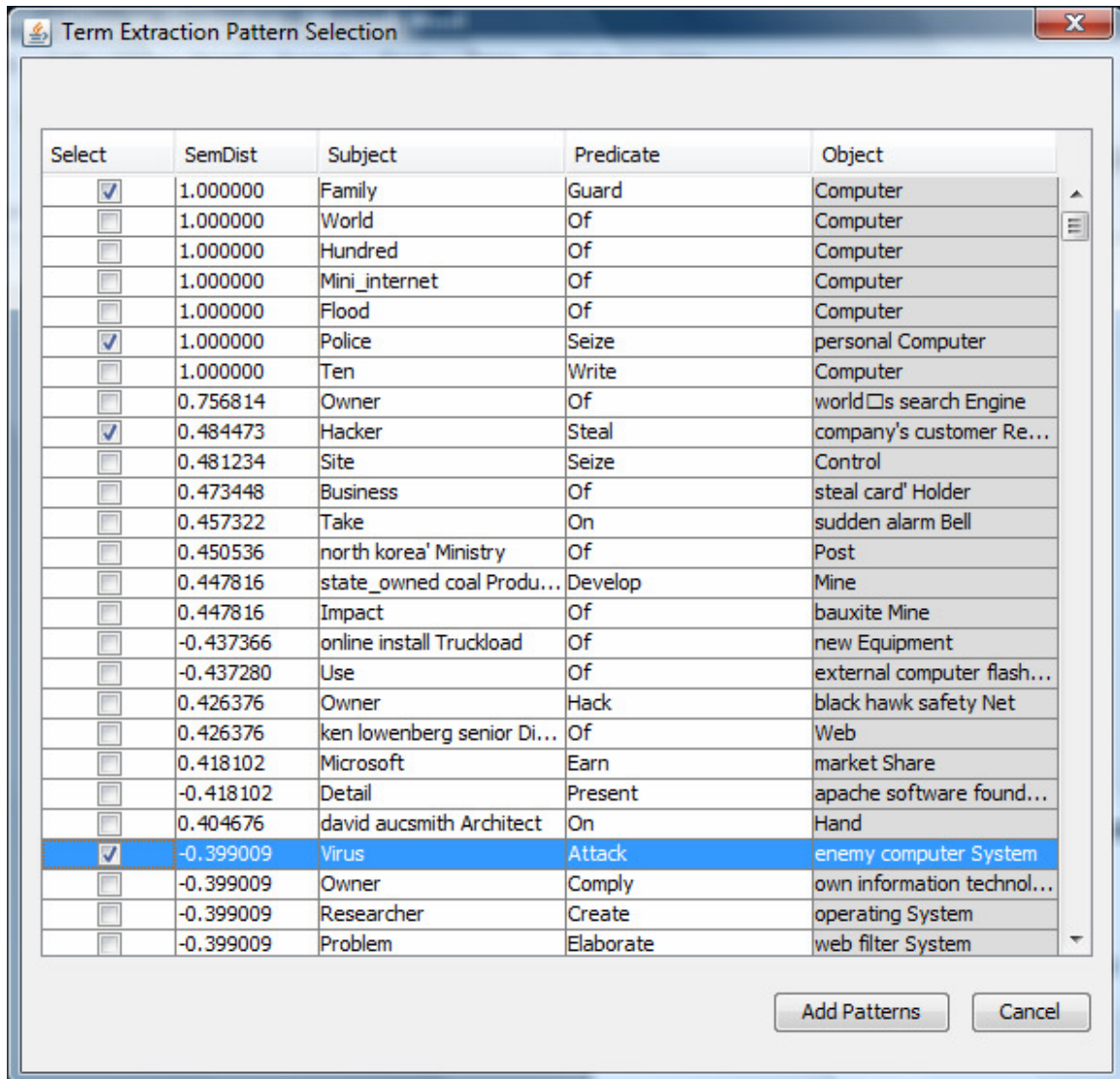
In addition to using the above heuristic, terms are searched for in the WordNet by using the most number of words within in multi-word term. That is the words in the multi-word term as used from left to right. For example, the term "financial institution" is searched for by first looking for the word "institution" which is known as the head noun. If the word "institution" is found within WordNet, then the process goes to the next word within the term, which in this case would be "financial institution". If the term "financial institution" is found then it is used in the semantic relevance computation, otherwise just the word "institution" is used in the computation.

5.7 *Ranking SPOs*

After the database of NP, VP, and SPOs entities is created, and the initial seed ontology has been built, a list of ranked SPOs can be presented to the user. This list is ranked based on the semantic relevance between a given seed concept's term and either the subject or object phrase within the stored SPOs. Thus the user selects a specific seed concept and indicates either the subject or object field to be used in the sorting. The ranked list is presented to the user in ascending order based on semantic relevance of the selected fields. The user can select SPOs which appear to be good candidates for detecting domain specific terms. It is critical to rank the SPOs well to help reduce the workload on the knowledge engineer by limiting the amount of SPOs that has to be analyzed. Note that the SPOs are ranked first by their semantic relevance, and if the relevance value is zero, then the SPOs are ranked alphabetically by their predicate values. Ranking has been performed previously in literature in projects such as [Sabou 2004] for ranking verb-noun pairs list for the purposes of concept assignment to each pair. In [Sabou 2004] the verb-noun pairs are ranked based pair frequency, term weights, or number of occurrences within an API. In contrast, the ranking performed in work is based on semantic relevance between the subject or object term compared to a given seed concept's term.

The SPOs are formally represented as $s p o_i = \langle s_i, p_i, o_i \rangle$, where s_i is the subject term, p_i is the predicate term, and o_i is the object term. To create an SPO list for a given concept c_j , either s_i or o_i is semantically compared against the terms listed for concept c_j and are presented in ranked order based on the semantic relatedness of either s_i or o_i to concept c_j . The closer the two terms are, the higher in the list they will be placed. In

addition, the degree of computed semantic relevance is shown for each ranked SPO. The following diagram in Figure 9 is a screen shot from the program that displays a ranked SPO list.



| Select | SemDist | Subject | Predicate | Object |
|-------------------------------------|-----------|----------------------------|-----------|----------------------------|
| <input checked="" type="checkbox"/> | 1.000000 | Family | Guard | Computer |
| <input type="checkbox"/> | 1.000000 | World | Of | Computer |
| <input type="checkbox"/> | 1.000000 | Hundred | Of | Computer |
| <input type="checkbox"/> | 1.000000 | Mini_internet | Of | Computer |
| <input type="checkbox"/> | 1.000000 | Flood | Of | Computer |
| <input checked="" type="checkbox"/> | 1.000000 | Police | Seize | personal Computer |
| <input type="checkbox"/> | 1.000000 | Ten | Write | Computer |
| <input type="checkbox"/> | 0.756814 | Owner | Of | world's search Engine |
| <input checked="" type="checkbox"/> | 0.484473 | Hacker | Steal | company's customer Re... |
| <input type="checkbox"/> | 0.481234 | Site | Seize | Control |
| <input type="checkbox"/> | 0.473448 | Business | Of | steal card' Holder |
| <input type="checkbox"/> | 0.457322 | Take | On | sudden alarm Bell |
| <input type="checkbox"/> | 0.450536 | north korea' Ministry | Of | Post |
| <input type="checkbox"/> | 0.447816 | state_owned coal Produ... | Develop | Mine |
| <input type="checkbox"/> | 0.447816 | Impact | Of | bauxite Mine |
| <input type="checkbox"/> | -0.437366 | online install Truckload | Of | new Equipment |
| <input type="checkbox"/> | -0.437280 | Use | Of | external computer flash... |
| <input type="checkbox"/> | 0.426376 | Owner | Hack | black hawk safety Net |
| <input type="checkbox"/> | 0.426376 | ken lowenberg senior Di... | Of | Web |
| <input type="checkbox"/> | 0.418102 | Microsoft | Earn | market Share |
| <input type="checkbox"/> | -0.418102 | Detail | Present | apache software found... |
| <input type="checkbox"/> | 0.404676 | david aucsmith Architect | On | Hand |
| <input checked="" type="checkbox"/> | -0.399009 | Virus | Attack | enemy computer System |
| <input type="checkbox"/> | -0.399009 | Owner | Comply | own information technol... |
| <input type="checkbox"/> | -0.399009 | Researcher | Create | operating System |
| <input type="checkbox"/> | -0.399009 | Problem | Elaborate | web filter System |

Figure 9 - Example of ranked SPOs for user evaluation

Note that the highlighted column indicates that the object field was used in the comparison. Checkboxes are provided for the user to indicate which SPOs are to be

transformed into term extraction patterns. Only unique TPs are stored so if multiple SPOs are transformed into the same TP, only a single copy of the TP will be saved.

For a given seed concept, the user can select multiple SPOs to be used for detecting domain specific concepts. In addition, selected SPOs can also be removed in later steps within the process to allow for refining of the ontology construction parameters.

5.8 Creating Term Extraction Patterns

The purpose of selecting SPOs from the ranked list is for the creation of Term Extraction Patterns (TEPs). For a given concept, multiple SPOs are selected for and stored in the concept's parameter list. These SPOs form a set of TEPs for detecting semantically relevant terms for the given concept. The TEPs can be viewed as a form of linguistic patterns used for candidate term detection for a domain-specific vocabulary.

Since our goals of the ontology construction process is to strive for generating a higher quality ontology, it is necessary to focus on quality instead of quantity for the term extraction process. A careful creation and selection of the TEPs is needed as the system requires specific groups of concepts and relationships. There may be a considerable amount of information that others may consider as being part of a cyber-attack ontology but which are not what is required in this particular case. In other words, it may be difficult to find text articles that strictly use terms within the sought after domain-specific vocabulary. Therefore this system relies on the TEPs to discover terms that have a high-probability of domain relevancy.

The SPOs are transformed into a TEP by identifying either the subject or object position within a selected SPO as a “slot” that will later be filled with a candidate

domain relevant term. For example, suppose the user selects the following SPO “*Cyber-weapon -> target -> Critical Infrastructure*” and chose the object slot as the domain relevant term. The generated pattern would be “*Cyber-weapon -> target -> **” and would match any SPO containing the phrase *cyber-weapon* in the subject position and the verb *target* in the predicate position. This pattern would assume the term found in the object slot (asterisk) is a term that is semantically similar to the seed concept that was assigned this pattern. As a result, the list of TEPs for given seed concept are used in harvesting domain-specific terms by applying the TEPs to every SPO stored within the SPO database. As new SPOs are added to the SPO database, the TEPs can be re-applied to harvest additional domain specific terms. Thus a pool of domain specific terms is formed for each seed concept.

Note that the TEPs actually serve for two purposes: first they represent the linguistic patterns for detecting potential domain-specific terms, and second, they indicate relations that belong to the corresponding concept. Thus when the user selects SPOs for a given concept, they are also selectively indicating non-taxonomic relationships that belong to the concept. Through the combination of multiple iterations selecting SPOs and specifying terms for the concepts, the user executes a process of describing features of a concept. This in turn, is used to mine semantically related concepts by the user describing the seed concept’s relationships and then in using these relationships to find terms that have the same relationships as the seed terms.

5.9 Building Term Pools

Term Pools (TPs) are the resultant pools of candidate domain specific terms that are extracted by applying the TEPs against the database of SPOs. A TP is created and

updated for each seed concept within the ontology. Thus the process to create or update the TPs consists of scanning the database of SPOs looking for SPOs that match one of the TEPs defined for a given seed concept. Note that presently the comparison against the subject or object term is compared based on using the head-noun of the terms. For example, suppose a given concept has the TEP of “successful cyber-attack -> on -> *” where * is the slot that contains candidate terms that map to the concept. So this TEP would match to the SPO of “sophisticated cyber-attack -> on -> US defense department” because only the head-nouns, “cyber-attack”, of each term is compared. The following pseudo-code in Figure 10 describes the algorithm for building the TPs:

```

FOR each seed concept,  $c_i$  DO:
  FOR each SPO,  $spo_j$  DO:
    FOR each TP,  $tp_z$  DO:
      IF predicate( $spo_j$ ) == predicate( $tp_z$ )
        IF useSubjectSlot == TRUE
          IF headNounObj( $spo_j$ ) == headNounObj( $tp_z$ )
            addSubTermToPool( $spo_j$ ,  $c_i$ )
          ELSE
            IF headNounSub( $spo_j$ ) == headNounSub( $tp_z$ )
              addObjTermToPool( $spo_j$ ,  $c_i$ )
        DONE
      DONE
    DONE
  DONE

```

Figure 10 - Pseudo code for building TPs

The TPs are maintained for each seed concept and are updated as new terms are discovered during the TP updating process. Statistical data such as the frequency of occurrence within the *corpus* are maintained for both the initial TP creation and updates to the TP. In addition, each term within the TP has a link to its corresponding TEP that generated the term. This is needed for evaluation purposes for determining the quality of the TEPs.

5.10 Conceptualizing Term Pools

The final stage in building the ontology is the conceptualization of the TPs and hierarchically arranging the concepts. This is done by semantically comparing the terms to one another and creating clusters of semantically similar terms based on a given threshold. Thus each cluster forms a concept. Later the clusters are arranged hierarchically based on their semantic relevance to one another and the taxonomic relationships between one another.

The Hierarchical Agglomerative Clustering (HAC) is used for clustering similar terms into corresponding concepts and to link pairs of parent-child concepts together in a bottom-up fashion. The HAC method has been used in previous work for hierarchical arrangement as described in [Cimiano et al. 2004]. The core components of this method are the construction of a two-dimensional term relevance matrix and a method of determining the semantic distance between every pair of terms. The process continuously reduces the rows and columns of the matrix by combining the current iteration's semantically closest concepts and terms building the hierarchy from bottom up. Within the process, two experimentally derived thresholds are used. The first threshold, known as *SynonymThreshold*, identifies if two terms are semantically close enough to be considered synonyms of the same concept. The second threshold, known as *TaxonomicThreshold*, is used to identify if two terms are not close enough to be considered synonyms but close enough to have a parent/child taxonomic relationship. If the semantic distance value is less than both of these values, then the term pairs being analyzed are considered non-related. The computation described previously in (1) is used for all semantic relevance computations and for determining the taxonomic relationships.

Also note that the first term found for a new concept is used as the name of the concept. WordNet is queried with the term, and only the words within multi-word terms found in WordNet are used. For example, if the term “Nuclear Power Plant” is designated as the concept’s name, and WordNet only lists a “Power Plant” term, then the string “Power Plant” will be used as the concept’s name. Similarly, if a term found for a new concept is not listed within WordNet, including the term’s head noun, then the full term is used as the concept’s name. A head noun is the main noun in the term. For example, in the term “big brown cow” the adjectives “big” and “brown” modify the head noun of “cow”.

A final step is added in that after the above two steps, any concepts that contain a term that is contained in WordNet and has a single sense listed, will have an abstract concept added to it if WordNet has a hyponym listed. Only words with a single sense are analyzed to reduce the introduction of noise due to word-sense ambiguity. More generalized concepts are added to the ontology to help in categorizing the more specific concepts because most terms found in text tend to be more specific than generalized.

The HAC algorithm actually serves two purposes. First, it conceptualizes the terms into concepts by clustering the terms into semantically similar groups. Second, it arranges the concepts into a hierarchy based on taxonomic relationships. However, within the semantic distance matrix, there may be terms that are not listed in WordNet and thus pockets of hierarchies may develop within the matrix.

The open source software package of Graphviz (<http://www.graphviz.org/>) is used to convert the ontology meta-model to a graphical format for analysis. The concepts are

converted to nodes with the parent-child relationships converted to links. Added generalized concepts from WordNet are identified by showing dashed ovals.

5.11 *Ontology Meta-Model*

The seed concepts, TEPs, and TPs, are all stored as part of the ontology construction parameters. These parameters are structured into a model that is considered an ontology meta-model or goal tree that is used to create and extend/refine the ontology. Thus users can refine the parameters stored within the goal-tree to improve the ontology. Figure 11 depicts the data structure of the meta-model for the ontology construction process:

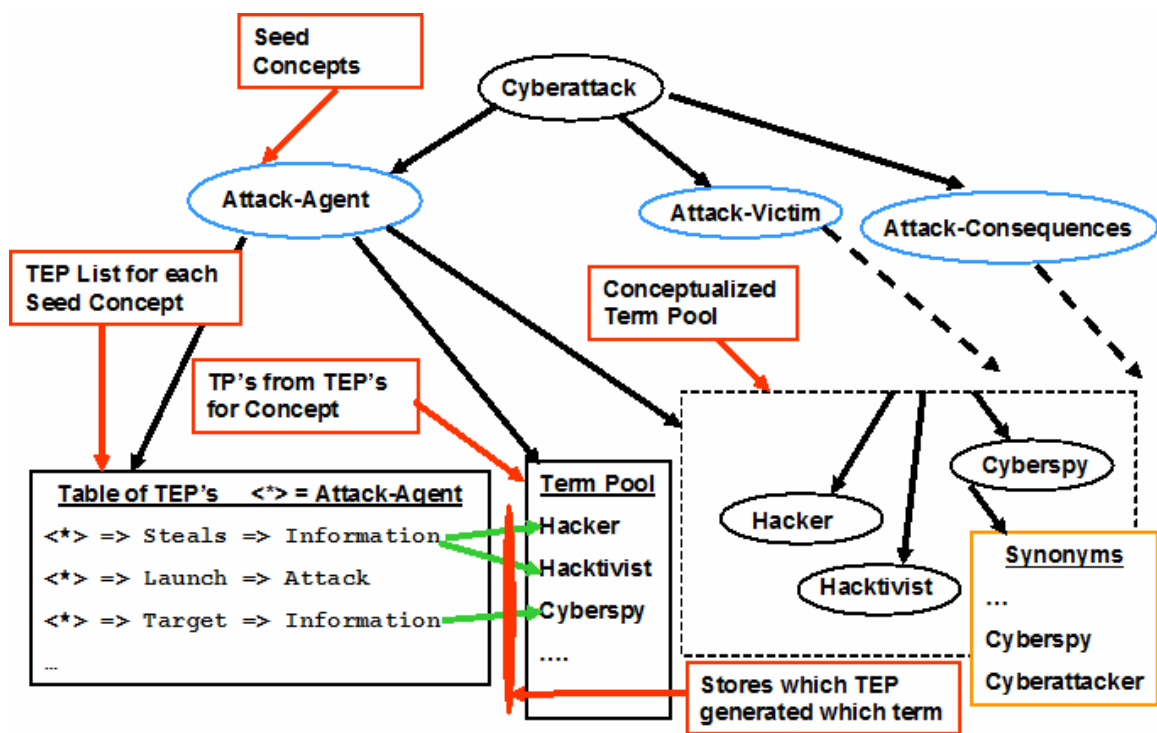


Figure 11 - Ontology Meta-Model

Note that the breakdown is such that for each seed-concept, there exists a list of TEPs for extracting semantically related terms for the seed-concept. In addition, each seed concept contains a corresponding TP of semantically related terms that are or will be

conceptualized and hierarchically arranged into an ontology whose root node is the seed concept. Each term within the TP also is linked to the TEP that generated the respective term. This is necessary so that the user can evaluate the performance and quality of each TEP. Also, the terms that make up each created concept are stored along with their respective concept. This is also beneficial for later ontology construction needs, in order to determine if the terms within a given concept must be split up into additional concepts and further sub-divided.

5.12 Iteratively repeating the process

Once the initial process is complete and the first ontology is constructed, the user can iteratively add text articles for phrase and SPO extraction to be followed by updating the TPs and eventually updated the ontology. In addition, users can evaluate the generated TPs and ontology and add/remove TEPs to refine the ontology refinement process.

The iterative process consists of following steps:

- a. **Step 1:** Increase and update the text *corpus* by adding more domain specific text files that may include updated ontological knowledge
- b. **Step 2:** Parse only the added text files and update the parsed article database.
- c. **Step 3:** Update the phrase and SPO database by only adding those phrases and SPOs that were found in the added parsed articles. This is critical to maintain the correct frequency counts (statistical data) on each unique phrase and SPO as well as to avoid redundant computations.
- d. **Step 4:** Apply the currently defined TEPs against the entire *corpus* generating a new TP. However, the new terms will be merged into the current TPs to

preserve the statistical data on each term and avoiding creating duplicate terms. Each term within the TP will be unique and the number of occurrences within the *corpus* will be updated to reflect the correct value.

- e. **Step 5:** Execute the ontology update method such that terms that are in the TP and not accounted for in the ontology will be conceptualized and added to the ontology. Note that this process is different than the conceptualization and hierarchically arranging of terms within the initial process. This process consists of extending the ontology as compared to the initial process of constructing the ontology.
- f. **Step 6:** (Optional) The user can review currently constructed ontology, TPs, and TEPs for the purposes of determining the quality of the TEPs. New ranked lists of SPOs can be viewed for the possibility of creating new TEPs as well as the possibility of removing TEPs that are not operating correctly.

The following pseudo code in Figure 12 describes the ontology extension method:

```

FOR each seed concept,  $sc_i$  DO:
  FOR each term  $t_j$  in  $TP_i$  DO:
    added = FALSE
    // Look at each concept and determine if
    // this term should be assigned to concept
    FOR each concept  $c_i$  in hierarchy  $h_i$  DO:
      sd = semdist( $t_j$ ,  $c_i$ )
      IF (abs(sd) >= 0.95)
        addTermToConcept( $t_j$ ,  $c_i$ )
        added = TRUE
    DONE
    // if term was not assigned to concept, look
    // if this term represents a new concept
    IF added == FALSE
      FOR each concept  $c_i$  in hierarchy  $h_i$  DO:
        sd = semdist( $t_j$ ,  $c_i$ )
        IF (abs(sd) >= 0.50)
          addSubSuperToConcept( $t_j$ ,  $c_i$ , sd)
          added = TRUE
      DONE
      // If term not added, then by default make it a
      // subconcept of the root node
      IF added = FALSE
        addSubToRoot( $t_j$ , root)
    DONE
  DONE

```

Figure 12 - Pseudo code for conceptualization and hierarchy arrangement updating

5.13 Term Pool to SPO Ranking Feed-back Loop

As part of an incremental building process, the terms generated within a given TP are fed back into the ranking process to help push potentially relevant SPOs closer to the top of the rankings, thus helping to reduce the workload on the knowledge engineer. A fixed semantic relevance value of 0.9988 is assigned to an SPO subject or object term whose head noun matches a head noun from one the TP terms. As a result, as the number of terms within the TPs increase, potentially more relevant SPOs can be pushed higher into the rankings. Note that the highest value of semantic relevance is 1.0, and as such

0.9988 was selected to indicate SPO terms that may not be an “exact” synonym. In addition, this value makes it easy for the knowledge worker to spot SPO ranking values that identify a TP term match.

The following example highlights the benefits from the use of the feed-back loop. Suppose a seed ontology is created that contains a seed concept of Attack-Agent that is designated with the term of “hacker”. This seed concept’s purpose is to contain those entities that attack things. The user can view a SPO list based on this seed concept whose ranking will show terms similar to “hacker” near the top of the list. Terms with head nouns of “hacker” will appear closest to the top of the list. Table 3 shows an example excerpt of the SPO rankings based on the subject term:

Table 3 - SPO ranking before using TP terms

| Semantic Relevance | Subject | Predicate | Object |
|---------------------------|----------------|------------------|---------------------|
| ... | ... | ... | ... |
| 1.0000 | Hacker | Create | Countermeasure |
| 1.0000 | Muslim Hacker | Deface | Danish Web Site |
| 1.0000 | Hacker | Deface | Individual Web Site |
| 1.0000 | Hacker | Demand | Apology |
| ... | ... | ... | ... |
| 0.0000 | Hacktivist | Deface | Web Site |
| ... | ... | ... | ... |

Note that the term “Hacktivist” is also in the list, but is near the bottom, as WordNet does not contain the term “Hacktsivist”. However, suppose the user selected the SPO of “Hacker => Deface => Individual Web Site” which would transform into a TEP of “* => Deface => Site”. Afterwards, when the TPs are generated from this TEP, the “Hacktivist” term would be extracted and placed into the TP. Thus future SPO rankings

for the Attack-Agent seed concept would place SPO's with the "Hactivist" term higher in the list as shown in the excerpt in Table 4.

Table 4 - SPO ranking after using TP terms

| Semantic Relevance | Subject | Predicate | Object |
|---------------------------|----------------|------------------|----------------|
| ... | ... | ... | ... |
| 1.0000 | Hacker | Target | Credit card |
| 1.0000 | Hacker | Trigger | Cascade Effect |
| 1.0000 | Hacker | Turn | Attention |
| 1.0000 | Teenage Hacker | Wreak | Virus |
| 0.9988 | Hactivist | Deface | Web Site |
| 0.9988 | Hactivist | Expose | Strategy |
| ... | ... | ... | ... |
| 0.9988 | Hactivist | Steal | Documents |
| ... | ... | ... | ... |

Now the term "Hactivist" appears higher up in the rankings and from observation, the SPO of "Hactivist => Steal => Documents" appears to make a good TEP for mining more semantically similar terms to the Attack-Agent concept for entities that "steal documents".

6 Experimentation, Results, and Discussion

Presently there is no universally accepted standard for judging the quality of an ontology. However, there are various methods described in literature for ontology evaluation. One possible method is the evaluation of the generated ontology by domain experts. Multiple experts may use a common criteria and rating system for evaluating the ontology such as in [Caraballo 1999]. Another method is to compare the generated ontology to a manually created “golden” standard of the domain being compared. Various ontology similarity and structure measures such as AKTiveRank as described in [Alani and Brewster 2006] may be used. Finally, a simple and lightweight solution would be to simply analyze the generated ontology for incorrect concepts and relationships. For example, a military domain containing the concept of “tank” linked to the sub concept of “Fish Tank” would indicate a probable incorrect relationship and concept, as fish tanks are most likely not part of the military domain.

6.1 Experimentation

For the purposes of evaluating the developed methodology, experiments were performed using the cyber-attack domain previously described. Due to the lack of existing related text *corpora*, cyber-attack related articles were manually selected from various news sites across the Web in this research. The selected articles were stored in plain text documents and stored as one article per text file within a single directory. The statistics of the experiment are reported in terms of number of articles processed, number of phrases and SPOs extracted, number of TEPs created, size of TPs, number of created concepts, and finally a graphical visualization of the resulting ontology.

6.2 Analysis of Boot Strap process

The initial iteration was executed by defining the seed ontology, selecting TEPs for each seed concept, and then running the process for building the TP and for conceptualization and hierarchy arrangement. Note that although the cyber-attack ontology proposed contained seven seed concepts, we selected four of the seven for testing. This was needed due to the lack of sufficient information sources in finding TEPs for three of the seven. So for the analysis, the seeds concepts of Attack-Agent, Attack-Victims, Attack-Consequences, and Attack-Means were evaluated.

A total of 191 plain text articles were present within the text *corpus*. Table 5 indicates the number of phrases and SPOs that were extracted from this corpus:

Table 5 - Initial run *corpus* extraction statistics

| # Documents | #Noun Phrases | #Verb Phrases | # SPOs |
|-------------|---------------|---------------|--------|
| 191 | 13,002 | 681 | 4,716 |

Table 6 indicates the statistics for each seed concept. Note that the last column labeled "Added Concepts" are the count of concepts added from WordNet in an attempt to inject additional higher level concepts for better classification.

Table 6 - Initial run TEPs and terms per concept

| Seed Concept | #TEPs | Term Pool Size | Concepts | Added Concepts |
|---------------------|-------|----------------|----------|----------------|
| Attack-Victims | 18 | 30 | 23 | 5 |
| Attack-Agent | 11 | 78 | 48 | 16 |
| Attack-Consequences | 12 | 29 | 21 | 7 |
| Attack-Means | 3 | 4 | 2 | 0 |

From the above table it can be seen that there are more terms than TEPs, and one could theorize that if the same TEPs were used with a larger text *corpus*, the number of generated terms would increase due to a larger base of potential terms. Note that the Attack-Means TEPs did not generate very many concepts. This may be an indication that the TEPs that were chosen were poor, that the concept may be difficult to represent with TEPs, or that the frequency of SPOs that can be used to detect terms for Attack-Means is low.

The initial output from the ontology construction is shown in Figure 13. The Graphviz package is used to convert the meta-ontology model to a graph where the ovals represent the concepts and the arrows or edges between the ovals represent the taxonomic relationships. The hierarchy of the structure flows from left to right where the left side is more general terms and the right side are more specific. The farthest left node is the top of the tree, which is the main node of Cyber-Attack. Next to it are the four seed concepts shown with each of their respective term pool clusters. Note that there are also some concepts/terms that are generated from more than one seed concept's TEP. Abstract concepts that are added from WordNet are indicated using dashed ovals. Also note that there are few sibling nodes. Further work is needed in clustering the generated concepts and using additional more generalized concepts in classifying the concepts.

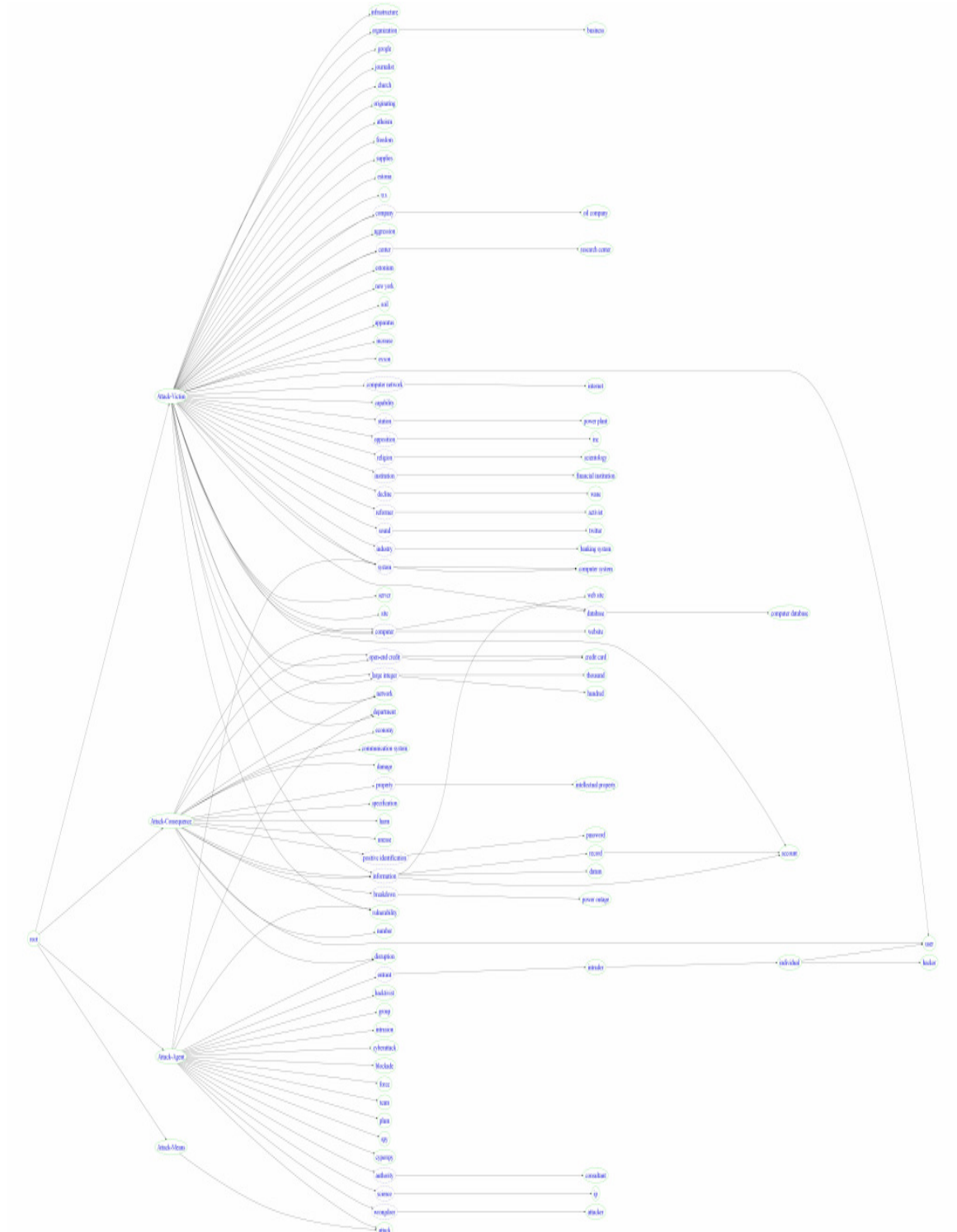


Figure 13 - Resulting ontology

Figure 14 shows an enlarged section of the resulting ontology.

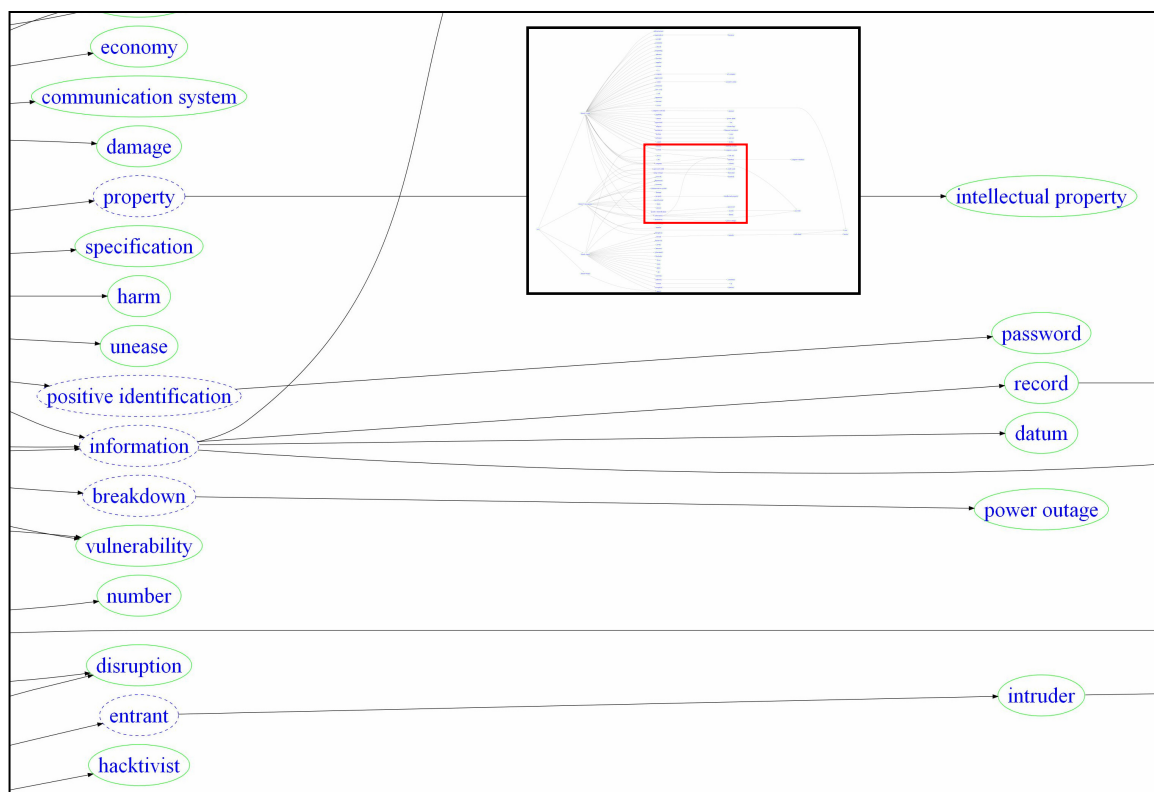


Figure 14 - Enlarged section of ontology

The following tables list all of the resulting terms from the four seed concepts from the boot strap process test. Results from the experiments show that the terms for the attack consequences would be better conceptualized using the predicates from the TEPs.

Table 7 - Resulting terms for Attack-Agent concept

| | |
|----------------------------------|-------------------------|
| Israeli Hacker | Russian Hacker |
| wahhabi Hacker | Intruder |
| Hacker | computer System |
| Group | computer Hacker |
| Intrusion | Attack |
| Cyberattack | allege Hacker |
| Disruption | Team |
| Hactivist | Plum |
| Muslim Hacker | FBI computer Consultant |
| rant Blockade | Russian Spy |
| Chinese Hacker | IP |
| So-called patriotic-hacker Group | Cyperspy |

| | |
|----------------------|--------------------|
| very few Individual | Attacker |
| Korean internet User | Department |
| Russian Force | same Vulnerability |

Table 8 - Resulting terms for the Attack-Victim concept

| | |
|---------------------------------------|--|
| Google | Twitter |
| estonian government Network | Infrastructure |
| Journalist | own bureaucracy's information technology Infrastructure |
| Church | non_governmental Infrastructure |
| Server | member state Estonium |
| fra's Website | new York |
| us defence Department | polish government System |
| corporate infrastructure Originating | small Network |
| Network | Critical infrastructure System |
| power Plant | computer System |
| Country's Network | web Site |
| google Inc | shia Website |
| Internet | american Soil |
| Atheism | telecommunications computer Network |
| Freedom | Nation's technology Apparatus |
| Scientology | Sunday Website |
| uk's computer Network | america's Infrastructure |
| oil Supplies | us Increase |
| major financial Institution | shiite_related Site |
| vulnerable Business | other shiite Site |
| Montenegrin companies' Website | Blogger's Site |
| Internet System | Exxon |
| other Network | oil Company |
| government Server | Website |
| Estonia | retail System |
| U.S. | Banking System |
| library Site | enemy computer System |
| US military Computer | restricted computer Database |
| Wane | Chechen governmental youth Website |
| Non_governmental Organization | Database |
| defense Company | System |
| Google's Server | higher_balance Account |
| human_rights Activist | home User |
| computer Network | credit Card |
| NATO's unfair Aggression | Vulnerability |
| public Site | Hundred |
| India's Bhabha Atomic Research Center | Georgian government web Site |

| | |
|---------------------|----------------------------------|
| Center | Danish web Site |
| Government Computer | Indian show trademark Capability |

Table 9 - Resulting terms for the Attack-Consequence concepts

| | |
|---------------------------------|------------------------|
| electronic communication System | Gmail Account |
| greater Damage | valuable Information |
| serious Disruption | social security Number |
| company's customer Record | phone Number |
| card Number | Network |
| intellectual Property | Higher-balance Account |
| Specification | home User |
| Information | credit Card |
| Password | web Site |
| sensitive Datum | US Economy |
| customer Information | entire Economy |
| classified Information | power Outage |
| Users' Datum | Harm |
| Thousand | public Unease |
| proprietary Information | |

Table 10 - Resulting terms for the Attack-Means concept

| |
|----------------------|
| cyber Attack |
| Attack |
| retaliatory Attack |
| low_intensity Attack |

It's difficult to formally evaluate the generated terms as correct or incorrect as there is no golden standard to compare too that aligns with the goals of this ontology. A few cyber-attack related ontologies were found during the literature review such as the ones in [Prueitt & Stephenson 2005], [Simmonds et al. 2004], and [Shiva et al. 2009]. But they do not deal with the concepts needed by this research.

In addition, the author of this work is not considered a domain expert of the desired cyber-attack ontology. However, for future work, a fair evaluation may be one of

how well the ontology works for its corresponding application, which in this case is the CyCast application.

6.3 Incremental analysis

After the initial boot strap process is complete, the user can incrementally continue to build and refine the ontology. If desired, the user can leave the initial set of TEPs alone, and continue to add articles with updated information and process them into their respective term pools and have them added to the ontology. This cycle can continue building and extending the ontology. The user can re-run parts of the process with new adjustments, such as adding new TEPs and re-running them over the current text *corpus*. Also if desired, the user can add new seed concepts and continue to run the system periodically to add the processing of new articles.

6.4 Analysis of Operations

The following sections describe the results and analysis from the various components of the ontology construction processes.

6.4.1 Seed Ontology analysis

The seed ontology allows the user to establish the upper-level concepts that will be further extended and defined during the ontology constructing process. Challenges arise in determining how specific or abstract to make the seed concepts. In addition, it may be difficult to designate a single term that can identify a relatively abstract concept. For example, the Attack-Victim concept used in the cyber-attack domain is a relatively abstract concept that would be difficult to describe using a single term. The purpose of

the Attack-Victim concept is to describe victims of a cyber-attack that could be such entities as a network, a computer, or higher level ones such as governments, types of businesses, or critical infrastructure systems. Thus with these variations of sub-concepts it would be difficult to describe all of them with a single or handful of terms such as other systems do [Liu et al. 2005]. This provides the motivation for using patterns for mining terms related to the abstract seed concepts. Although this system requires a user defined term for each seed concept, the term's purpose is not to find semantically similar terms, but to aid in finding patterns that can possibly locate a larger quantity of semantically similar terms. Thus it is just part of the boot-strapping process. The patterns are actually what are used to find related terms. Similar work in using terms defined for seed concepts for pattern mining have been reported in [Brewster et al. 2002]. However, they look for Hearst-style patterns [Hearst 1992] instead of the SPO patterns used in this work.

Note however in this research that the seed concepts can only be defined on a single level. That is there presently is no support for defining hierarchical seed concepts, which may be supported in future versions. There may be tradeoffs in the amount of work versus reward that would result in defining these hierarchical seed concepts. Obviously an increase in the number of hierarchical seed concepts defined would result in a decrease in the amount of programmatic decisions needed by the system in the process of conceptualizing and arranging concepts.

6.4.2 Semantic Relevance computation

The computation of the semantic relevance is based on using a general purpose ontology in combination with Lin's formula for semantic relevance. As mentioned before there are problems with using a general purpose ontology such as WordNet for

determining semantic relevance. Problems such as multiple word senses and lack of support for many domain specific terms cause problems with determining semantic similarity. In addition, the concepts and structures within the general purpose ontology may not completely match or align with the domain specific ontology that is being built. Note that the addition of other forms of semantic relevance may be combined with the use of WordNet for improved semantic relevance. Work described in literature in [Cimiano et al. 2003] describes using multiple forms of semantic relevance.

6.4.3 Term Extraction Patterns analysis

Observations of the results confirm the notion that the ratio for the number of usable patterns compared to the available patterns is relatively low, which has also been observed in literature [Rastegari et al. 2010]. In other words, pattern instances generally have high precision and low rates of recall. Another property of patterns to consider is the level of generalization versus the recall and precision rates. Patterns that are more generalized may result in more terms extracted and also larger amounts of incorrect terms. In contrast, more specific patterns may have a higher rate of precision and lower recall rate of extracted terms. These characteristics provide motivation for combining the use of patterns along with the use of other methods for term extraction such as term co-occurrence [Liu et al. 2005]. Also, the amount of relevant terms that a given pattern may generate needs to be considered. That is, two or more patterns may generate the same terms, so it would be better for efficiency purposes to choose the pattern that generates the greatest amount of domain relevant terms. In addition, there may be some concepts where no SPO patterns can be found that would generate related terms. Finally, the use of patterns for finding domain related terms is preferred over statistical methods for

computational efficiency when processing newly added articles. Pattern usage requires the system only to analyze the data from new articles as compared to statistical means, which would require an additional analysis of the entire text *corpus*.

Table 11 shows the results from the generation of a TP for the Attack-Agent seed concept and the corresponding TEPs.

Table 11 - Example Term Pool results from TEPs

| TEP | Candidate <i>Attack-Agent</i> Term |
|------------------------------|---|
| * => Attack => Site | Machine |
| * => Attack => Site | virtual Sit_ins |
| * => Attack => Site | computer Hacker |
| * => Attack => System | Virus |
| * => Attack => System | big economic Collapse |
| * => Attack => Website | israeli Hacker |
| * => Attack => Website | wahhabi Hacker |
| * => Breach => System | Hacker |
| * => Breach => System | Group |
| * => Claim => Responsibility | Post |
| * => Create => Attack | sophisticated Attacker |
| * => Cripple => Economy | cyber Attack |
| * => Cripple => Site | Attack |
| * => Deface => Site | Hactivist |
| * => Deface => Site | muslim Hacker |
| * => Deface => Site | rant Blockade |
| * => Deface => Website | hacker Group |
| * => Deny => Attack | chinese foreign Ministry |
| * => Deny => Involvement | Moscow |
| * => Deny => Involvement | chinese Official |
| * => Disrupt => Server | individual Acting |
| * => Disrupt => Site | Software |
| * => Gain => Access | Team |
| * => Gain => Access | Plum |
| * => Gain => Access | fbi computer Consultant |
| * => Gain => Access | russian Spy |
| * => Gain => Access | Ip |
| * => Gain => Access | Cyperspy |
| * => Gain => Access | Attacker |
| * => Gain => Access | Department |
| * => Gain => Access | same Vulnerability |
| * => Hack => Website | Specialist |

| | |
|---------------------------|----------------------------------|
| * => Launch => Attack | chinese Hacker |
| * => Launch => Attack | so_called patriotic_hacker Group |
| * => Launch => Attack | very few Individual |
| * => Launch => Attack | korean internet User |
| * => Launch => Attack | 15_year_old Canadian |
| * => Launch => Attack | north Korea |
| * => Penetrate => System | Ghostnet |
| * => Steal => Information | Intruder |
| * => Steal => Information | computer System |

6.4.4 Pattern ranking and domain term Feedback Loop

The ranking of the SPOs by semantically comparing either the subject or object term of each SPO to a selected seed concept's term was beneficial in reducing the amount of work needed to find potential TEPs for term extraction. In general, there are patterns located near the top of the list which were suitable. However, due to terms that were not correctly ranked by the semantic calculations, potentially useful patterns are also found considerably farther down the list. Problems with the semantic distance calculations are due to such issues as word sense ambiguity, words not being found in WordNet, and the differences of relationships between terms in WordNet and terms in the constructed ontology.

In order to continually improve the number of usable SPOs appearing near the top of the ranked SPO list, the technique of using a feedback loop of comparing the designated terms against terms within the corresponding term pool is used. This appears to help rank potentially usable SPOs whose terms are not identified in WordNet higher in the list. However, the evaluation of the feedback loop has to take into consideration the ratio of irrelevant terms versus relevant terms fed back into the ranking.

6.4.5 Conceptualization

This work considers the process of conceptualization as the task of creating concepts from the terms extracted by the patterns. These terms are a lexicon representation of concepts to be entered into the resulting ontology. A general purpose ontology, WordNet, was used to determine what concepts were represented by which terms as well as the taxonomic relations between terms. In addition, the HAC algorithm is used to cluster semantically similar terms into the concepts. For simplicity, the string of the first term found for the cluster is used as the concept's name which may or may not be appropriate for the given domain. Terms that were extracted but not found in WordNet were designated as new concepts and named from the corresponding term.

Problems arise with determining the relevance between terms as well as what concept a term represents due to multiple senses of words. In addition, the term and concept structure within WordNet may not be the correct one for the desired domain. Recall though that this research focuses on investigating and developing user guided ontology construction processes, and thus this task was not taken into main consideration and was simplified by using WordNet in order to focus the research efforts on user-guided technology.

6.4.6 Learning concept hierarchy

The HAC algorithm was used to hierarchically arrange the concepts with taxonomic relationships between them. However, due to the use of WordNet for determining semantic relevance, terms that do not appear in WordNet could not have their semantic relevance value determined and thus had a zero value. This created pockets of semantically related clusters, but not a complete tree. For those terms that

were not defined in WordNet, the terms are simply designated as stand alone concepts and made a direct child concept of the seed concept. As in the conceptualization process, the concept hierarchy learning was simplified in order to focus the research on the user guided technology. In other words, the conceptualization and concept hierarchy learning tasks were simplified but needed to help in the analysis and research of the user-guided technology. These processes can be improved by integrating more of the state-of-the-art methods.

6.4.7 Ontology Meta-Model

The ontology meta-model is the center of the novelty of this work and contains the user preferences and incremental additions and refinements. Within this model the upper-level seed concepts are stored that basically defines sub-ontologies within the ontology. The initial term for the seed concept is used to find patterns that in turn find more terms for the seed concept that find more patterns, and so on. In addition, the TEPs that generated the terms is saved for TEP evaluation. Note that the TEPs could also be used to determine relations and concept ranges for each term.

7 Conclusions and Future Work

The goals of this work were to research and implement a user-guided methodology for the purposes of constructing and incrementally refining a domain specific ontology. This included the exploration of the types of ontology construction parameters as well the types of user input for designating and refining the ontology construction parameters. In addition, research was performed on how to model the ontology construction and refining processes. In order to evaluate the feasibility and performance of these proposed methods, a software prototype was created to provide a means of evaluation. Experiments were performed based on the selected domain of cyber-attacks and the resulting terms, concepts, and taxonomic relationships were shown.

7.1 Results

Results of this research lie within the performance and new techniques of the developed software prototype for constructing and refining ontologies. The evaluation of the tool may be expressed in such parameters as (1) ease of use in terms of the number of semi-automatic ontology building functions provided and the amount of required user input, and (2) correctness of resulting ontology in terms of extracted terms, concepts, and relationships as well as the amount of missed concepts and relationships.

In terms of “ease of use”, several methods were developed and implemented within this research to reduce the burden of the knowledge worker. First, the use of the seed ontology was developed in order to allow the user to guide the development of a domain specific ontology by designating the upper-level concepts. Although seed

ontologies have been used in previous work, this work builds on the seed concepts by allowing the user to assign and refine parameters used in shaping the seed concept's child concepts. By doing so, it eases the burden of selecting candidate terms for a given seed concept.

Second, a ranked list of potential patterns is shown to the user for their evaluation for mining semantically related terms with respect to a given seed concept. The ranking helps reduce the amount of patterns the knowledge worker needs to evaluate. In addition, terms that are extracted from current patterns are fed back into the ranking process to further improve the ranking. Thus a continuous feed back loop develops to improve the ranking and further reduce the workload on the knowledge worker.

Third, each term that is extracted causes the corresponding pattern to be saved. This allows for a listing of which patterns generated what terms. Users can then evaluate the performance of the patterns in order to remove relatively “noisy” patterns or selected patterns that generate a greater number of semantically related terms. As a result, the user can optimize the patterns and thus help to reduce the amount of cleanup for the knowledge worker after the ontology is generated.

The correctness of the ontology is difficult to express as there does not exist a “golden” standard for the cyber-attack ontology. Note that the evaluation of the quality of ontologies is a well recognized research problem. By the very nature of the diversity of ontologies, the evaluation of them is highly subjective as different evaluators have different beliefs on the model of a given domain. Other methods for evaluating ontology quality are to have a group of domain experts judge the constructed ontology. Also, the

structure of the ontology may be evaluated for such errors as duplicate concepts or self-referencing concept where a parent concept is a parent of itself.

7.2 Main contributions

The main contributions to this work are defined in three areas. These areas are (1) a new method of semi-automatic ontology construction, (2) developed techniques for dynamical construction of domain ontology, and (3) applied the method and techniques for an experimental process of ontology construction in a cyber security domain.

This work is considered as a method of semi-automatic ontology construction by its technique for allowing the user to define initial ontology parameters and later to add and refine the ontology construction parameters. They include:

- a. **A streamlined process:** The process goes in a sequence of: Seed ontology specification → Term extraction patterns (TEP) → Text *corpus* processing → TEP Ranking and selection → Term Pool formation → Formal concept identification → Ontology updating
- b. **An incremental learning approach:** The system makes conceptualization of ontological terms based on the existing ontology structure and concept, starting from a seed ontology, continual in user-guided iterations
- c. **A user-in-the-loop control mechanism:** Check and feedback points are provided along the construction path for users to interact with the software systems
- d. **A goal tree representation scheme:** The system effectively stores and manages domain-specific expert knowledge and user preferences for guidance of the ontology construction process

In conjunction with the above methods, the following techniques were implemented and used:

- a. Seed ontology specification using goal tree
- b. Term extraction using text *corpus*
- c. Concept analysis using Subject-Proposition-Object (SPO) triples
- d. Concept selection using Term Extraction Patterns (TEP)
- e. Semantic relevance computation using a combination of WordNet and domain knowledge
- f. Concept generalization using tree hierarchy climb-up inference
- g. Concept specialization using tree hierarchy crawl-down inference

7.3 Future work

Several areas exist for improvement. First, in the system's present form, the seed ontology can only contain a single level of seed concepts. That is any defined seed concept is a sibling of all the other seed concepts. Future work may include the ability to define a hierarchy of seed concepts. In addition, currently only one term can be defined for each seed concept. Thus, it may improve the initial SPO ranking process to allow the user to define multiple terms per seed concept.

Second, there are problems determining the semantic relevance between terms for ranking the SPOs. The semantic relevance computation in current implementation is based on WordNet which inherently has problems due to words with multiple senses and terms that may not be listed in WordNet. Thus support for word sense disambiguation can be integrated to reduce the amount of inaccuracies due to the selection of incorrect word senses. In addition, other statistical methods for determining semantic relevance

could be combined with the WordNet based method to further improve the semantic matching. Also, Hearst patterns could be combined with the semantic relevance computation. The incorporation of weights may have to be applied with multiple forms of semantic relevance inputs to determine the significance of each input. Furthermore, the use of other knowledge resources such as Wikipedia could be considered as an additional measurement for semantic relevance. Wikipedia may be beneficial as it is a collaborative effort from Internet users and may be more update to date in some areas.

Third, improvements to the criteria for ranking the SPOs presented to the user may reduce the workload to the knowledge engineer. Perhaps additional ranking parameters such as head-noun frequency or verb frequency may be considered.

Lastly, improvements to the processes of conceptualization and hierarchical arrangements would result in an improved ontology. Presently, the initial semantic relevancy matrix used in the HAC algorithm may contain zero values due to the occurrence that terms are not found in WordNet and thus can not have their relevance computed.

8 References

- [Alani and Brewster 2006] Alani, H., and C. Brewster. "Metrics for Ranking Ontologies." *4th Int. EON Workshop, 15th Int. World Wide Web Conference*. 2006.
- [Berners-Lee et al. 2001] Berners-Lee, Tim, Jim Hendler, and Ora Lassila. "The Semantic Web: A New Form of Web Content That Is Meaningful to Computers Will Unleash a Revolution of New Possibilities." *Scientific American* 284 (2001): 28-37.
- [Bloehdorn et al. 2005] Bloehdorn, Stephan, Philipp Cimiano, and Steffen Staab. "An Ontology-based Framework for Text Mining." *LDV Forum* 20.1 (2005): 87-112.
- [Blomqvist 2007] Blomqvist, Eva. "Ontocase - A Pattern-Based Ontology Construction Approach." *Proceedings of OTM 2007: ODBASE - The 6th International Conference on Ontologies, DataBases, and Applications of Semantics, Vilamoura*. 2007.
- [Blomqvist 2008] Eva Blomqvist. "Pattern Ranking for Semi-Automatic Ontology Construction." *Proceedings of the 2008 ACM symposium on Applied computing (SAC '08)*. 2008.
- [Brewster et al. 2002] Brewster, Christopher, Fabio Ciravegna, and Yoric Wilks. "User-Centred Ontology Learning for Knowledge Management." *NLDB '02: Proceedings of the 6th International Conference on Applications of Natural Language to Information Systems-Revised Papers*. 2002. 203-07.
- [Budanitsky and Hirst 2001] Budanitsky, Alexander, and Graeme Hirst. "Semantic Distance in {WordNet}: An Experimental, Application-oriented Evaluation of Five Measures." *Proceedings of the Workshop on WordNet and Other Lexical Resources, Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-2001)*. 2001. 24-29.
- [Buitelaar et al. 2004] Buitelaar, Paul, Daniel Olejnik, and Michael Sintek. "A Protégé Plug-in for Ontology Extraction from Text Based on Linguistic Analysis." *The Semantic Web: Research and Applications, First European Semantic Web Symposium(ESWS)*. 2004. 31-34.
- [Buitelaar et al. 2005] Buitelaar, Paul, Philipp Cimiano, and Bernardo Magnini. "Ontology Learning from Text: An Overview." *Ontology Learning from Text: Methods, Evaluation and Applications*. Vol. 123. IOS, 2005. 1-10.
- [Caraballo 1999] Caraballo, Sharon A. "Automatic Construction of a Hypernym-labeled Noun Hierarchy from Text." *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*. College Park: Association for Computational Linguistics, 1999. 120-26.

- [Cimiano et al. 2003] Cimiano, Philipp, Aleksander Pivk, Lars Schmidt-Thieme, and Steffen Staab. "Learning Taxonomic Relations from Heterogeneous Sources of Evidence." *Ontology Learning from Text: Methods, Evaluation and Applications*. Vol. 123. IOS, 2005. 59-76.
- [Cimiano et al. 2004a] Cimiano, Philipp, Andreas Hotho, and Steffen Staab. "Comparing Conceptual, Divisive and Agglomerative Clustering for Learning Taxonomies from Text." *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI)*. 2004. 435-39.
- [Cimiano et al. 2004b] Cimiano, Philipp, Siegfried Handschuh, and Steffen Steffen. "Towards the Self-annotating Web." *Proceedings of the 13th International Conference on World Wide Web*. New York: ACM, 2004. 462-71. WWW '04.
- [Cimiano et al. 2005] Cimiano, Philipp, Andreas Hotho, and Steffen Staab. "Learning Concept Hierarchies from Text Corpora Using Formal Concept Analysis." *Journal of Artificial Intelligence Research* 4 (2005): 305-39.
- [Cimiano et al. 2006] Cimiano, Philipp, Johanna Völker, and Rudi Studer. "Ontologies on Demand? - A Description of the State-of-the-Art, Applications, Challenges and Trends for Ontology Learning from Text." *Information, Wissenschaft Und Praxis* 257.6-7 (2006): 315-20.
- [Colucci et al. 2008] Colucci, Simona, Eugenio Di Sciascio, and Francesco M. Donini. "Partial and Informative Common Subsumers of Concepts Collections in Description Logics." *Proceedings of the 21st International Workshop on Description Logics (DL 2008)*. Vol. 353. 2008.
- [Deerwester et al. 1990] Deerwester, Scott, Susan T. Dumais, George W. Furnas, and Thomas K. Landauer. "Indexing by Latent Semantic Analysis." *Journal of the American Society for Information Science* 41 (1990): 391-407.
- [Desmontils & Jacquin 2002] Desmontils, E., and C. Jacquin. "Indexing a Web Site with a Terminology Oriented Ontology." *The Emerging Semantic Web*. Vol. 75. IOS, 2002. 181-98.
- [Dodig-Crnkovic 2010] Dodig-Crnkovic, Gordana. "Constructivist Research and Info-Computational Knowledge Generation." *Model-Based Reasoning In Science And Technology - Abduction, Logic, and Computational Discovery Conference*. Vol. 314. 2009.
- [Fellbaum 1998] Fellbaum, Christiane. *WordNet: An Electronic Lexical Database*. Cambridge: MIT, 1998.

- [Fortuna et al. 2006] Fortuna, Blaz, Marko Grobelnik, and Dunja Mladeni. "Semi-automatic data-driven ontology construction system." *Proceedings of the 9th International Multi-conference Information Society IS-2006, Ljubljana, Slovenia*. 2006.
- [Gandhi et al. 2010] Gandhi, Robin, Anup Sharma, William Mahoney, William Sousan, Qiuming Zhu, and Phillip Laplante "The Cultural, Social, Economic, and Political Dimensions of Cyber Attacks", *IEEE Technology and Society*, In print, 2010.
- [Ganter and Wille 1999] Ganter, Bernhard, and Rudolph Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer Verlag, 1999.
- [Garcia et al. 2006] Garcia, Ana Cristina Bicharra, Inhaúma Neves Ferraz, and Fernando Bicharra Pinto. "The Role of Domain Ontology in Text Mining Applications: The ADDMiner Project." *Proceedings of the Sixth IEEE International Conference on Data Mining - Workshops*. Washington, DC: IEEE Computer Society, 2006. 34-38.
- [Gruber 1993] Gruber, Thomas R. "A Translation Approach to Portable Ontology Specifications." *Knowledge Acquisition* 5.2 (1993): 199-220.
- [Grüninger and Lee 2002] Grüninger, Michael, and Jintae Lee. "Introduction: Ontology Applications and Design." *Communications of the ACM* 45.2 (2002): 39-41.
- [Hearst 1992] Hearst, Marti A. "Automatic Acquisition of Hyponyms from Large Text Corpora." In *Proceedings of the 14th International Conference on Computational Linguistics*. 1992. 539-45.
- [Jain et al. 1999] Jain, Anil K., M. Narasimha Murty, and Patrick J. Flynn. "Data Clustering: A Review." *ACM Computing Surveys* 31.3 (1999): 264-323. Print.
- [Joho et al. 2004] Joho, Hideo, Mark Sanderson, and Micheline Beaulieu. "A Study of User Interaction with a Concept-based Interactive Query Expansion Support Tool." *Advances in Information Retrieval, 26th European Conference on Information Retrieval*. 2004. 42-56. Print.
- [Klein and Manning 2003] Dan Klein, Dan, and Christopher D. Manning. "Accurate Unlexicalized Parsing." *Proceedings of the 41st Meeting of the ACL-2003*. 2003. 423-30.
- [Lin 1998] Lin, Dekang. "An information-theoretic definition of similarity." *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*. 1998. 296-304.
- [Liu et al. 2005] Liu, Wei, Albert Weichselbraun, Arno Scharl, and Elizabeth Chang. "Semi-Automatic Ontology Extension Using Spreading Activation." *Journal of Universal Knowledge Management* 0.1 (2005): 50-58.

- [Marneffe et al. 2006] De Marneffe, Marie-Catherine, Bill MacCartney, and Christopher D. Manning. "Generating Typed Dependency Parses from Phrase Structure Trees." *LREC 2006*. 2006.
- [Missikoff et al. 2002] Missikoff, Michele, Roberto Roberto Navigli, and Paola Velardi. "The Usable Ontology: An Environment for Building and Assessing a Domain Ontology." *Proceedings of the International Semantic Web Conference (ISWC)*. Vol. 2342. Springer-Verlag, 2002. 39-53.
- [Pazienza et al. 2005] Pazienza, Maria Teresa, Marco Pennacchiotti, and Fabio Massimo Zanzotto. "Terminology Extraction: An Analysis of Linguistic and Statistical Approaches." *Knowledge Mining Series: Studies in Fuzziness and Soft Computing*. Springer Verlag, 2005.
- [Pedersen et al. 2004] Pedersen, Ted, Siddharth Patwardhan, and Jason Michelizzi. "WordNet: : Similarity - Measuring the Relatedness of Concepts." *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-2004)*. 2004. 1024-025.
- [Prueitt & Stephenson 2005] Prueitt, Paul, and Peter Stephenson. "Towards a Theory of Cyber Attack Mechanics." *First IFIP 11.9 Digital Forensics Conference*. 2005.
- [Rastegari et al. 2010] Rastegari, Y., M. Sayadiharikandeh, and B. Zibanezhad. "Lexical Pattern Generalization for Ontology Learning and Population: A Survey." *Lecture Notes in Engineering and Computer Science*. 1st ed. Vol. 2180. 2010. 551-54.
- [Resnik 1995] Resnik, Philip. "Using Information Content to Evaluate Semantic Similarity in a Taxonomy." *In Proceedings of the 14th International Joint Conference on Artificial Intelligence (AAAI)*. Montreal, 1995. 448-53.
- [Riloff 1993] Riloff, Ellen. "Automatically Constructing a Dictionary for Information Extraction Tasks." *Proceedings of the Eleventh National Conference on Artificial Intelligence*. AAAI, 1993. 811-16. Print. AAAI'93.
- [Sabou 2004] Sabou, Marta. "Extracting Ontologies from Software Documentation: a Semi-Automatic Method and Its Evaluation." *In Proceedings of the ECAI-2004 Workshop on Ontology Learning and Population (ECAI-OLP)*. 2004.
- [Schutz and Buitelaar 2005] Schutz, Alexander, and Paul Buitelaar. "RelExt: A Tool for Relation Extraction from Text in Ontology Extension." *International Semantic Web Conference (ISWC 2005)*. Springer, 2005. 593-606.
- [Shamsfard & Barforoush 2003] Shamsfard, Mehrnoush, and Ahmad Abdollahzadeh Barforoush. "The State of the Art in Ontology Learning: a Framework for Compariso." *The Knowledge Engineering Review* 8.4 (2003): 293-316.

- [Shiva et al. 2009] Simmons, Chris, Charles Ellis, Sajjan Shiva, Dipankar Dasgupta, and Qishi Wu. *AVOIDIT: A Cyber Attack Taxonomy*. Tech. no. CS-09-003. Memphis, 2009.
- [Simmonds et al. 2004] Simmonds, Andrew, Peter Sandilands, and Louis Van Ekert. "An Ontology for Network Security Attacks." *Proceedings of the 2nd Asian Applied Computing Conference (AACC'04)*, LNCS 3285. Springer-Verlag, 2004. 317-23.
- [Sousan et al. 2007] Sousan, William L., Matt Payne, Ryan Nickell, and Qiuming Zhu. "MetaData (Ontology) Incremental Building and Refinement Agents." *International Conference on Integration of Knowledge Intensive Multi-Agent Systems*, 2007. 2007. 127-32. KIMAS 2007.
- [Sousan et al. 2008] Sousan, William L., Qiuming Zhu, Ryan Nickell, William Mahoney, and Peter Hospodka. "Collecting Open Source Intelligence via Tailored Information Delivery Systems." *Journal Of Information Warfare* 2008th ser. 7.2 (2008).
- [Sousan et al. 2009] Sousan, William L., Kristina L. Wylie, and Zhengxin Chen. "Constructing Domain Ontology from Texts: A Practical Approach and a Case Study." *Fifth International Conference on Next Generation Web Services Practices*, 2009. 2009. 98-101.
- [Sousan et al. 2010] Sousan, William L., Qiuming Zhu, Robin Gandhi, William Mahoney, and Anup Sharma. "Using Term Extraction Patterns to Discover Coherent Relationships from Open Source Intelligence." *Second International Conference on Social Computing (SocialCom)*, 2010 IEEE. 2010. 967-72.
- [Ulicny et al. 2007] Ulicny, Brian, Chris Matheus, Mitch Kokar, and Ken Baclawski. "Uses of Ontologies in Open-Source Blog Mining." *Proceedings of the Second International Ontology for the Intelligence Community Conference (OIC-2007)* 2007.
- [Villaverde et al. 2009] Villaverde, Jorge, Agustin Persson, Daniela Godoy, and Analia Amandi. "Supporting the Discovery and Labeling of Non-taxonomic Relationships in Ontology Learning." *Expert Systems with Applications* 36.7 (2009): 10288-0294.
- [Wagner 2006] Wagner, Christian. "Breaking the Knowledge Acquisition Bottleneck Through Conversational Knowledge Management." *Information Resources Management Journal*, 19.1 (2006):70-83.
- [Yangarber et al. 2000] Yangarber, Roman, Ralph Grishman, Pasi Tapanainen, and Silja Huttunen. "Automatic Acquisition of Domain Knowledge for Information Extraction." *Proceedings of the 18th International Conference on Computational Linguistics*. 2000. 940-46. COLING 2000.
- [Zhou 2007] Zhou, Lina. "Ontology Learning: State of the Art and Open Issues." *Information Technology and Management* 8.3 (2007): 241-52.

- [Zhou et al. 2006] Zhou, Wen, Zongtian Liu, Yan Zhao, Libin Xu, Guang Chen, Qiang Wu, Mei-li Huang, and Yu Wen Qiang. "A Semi-automatic Ontology Learning Based on WordNet and Event-based Natural Language Processing." *International Conference on Information and Automation, 2006*. 2006. 240-44. ICIA 2006.
- [Zouaq and Nkambou 2009] Zouaq, Amal, and Roger Nkambou. "Evaluating the Generation of Domain Ontologies in the Knowledge Puzzle Project." *IEEE Transactions on Knowledge and Data Engineering* 21.11 (2009): 1559-572.