SIMOnt: A Security Information Management Ontology Framework

Muhammad Abulaish[1,#], Syed Irfan Nabi[1,3], Khaled Alghathbar[1] & Azeddine Chikh[2]

[1]Centre of Excellence in Information Assurance, King Saud University Riyadh, KSA
E-mail: {mAbulaish, kAlghathbar}@ksu.edu.sa
[2]College of Computer Science and Information Systems, King Saud University Riyadh, KSA
E-mail: az_chikh@ksu.edu.sa
[3]Faculty of Computer Science, Institute of Business Administration, Karachi, Pakistan
E-mail: snabi@iba.edu.pk

Abstract. In this paper, we have proposed the design of a Security Information Management Ontology (SIMOnto) framework, which utilizes natural language processing and statistical analysis to mine an exhaustive list of concepts and their relationships in an automatic way. Concepts are extracted using TF-IDF and LSA techniques whereas, relations between them are mined using semantic and co-occurrence based analyses. The mined concepts and relations are presented to domain experts for validation before creation of ontology using Protégé.

Keywords: Information security, Knowledge management, Ontology engineering, Ontology learning, Natural language processing.

1 Introduction

Due to existence of network-centric security environment, automatic discovery of resources and the ability to share information and services across different domains are some of the basic important requirements. The first step in fulfilling these requirements is to markup the security-related resources with various metadata in a well-understood and consistent manner [8]. Such annotations will enable resources to be machine-readable and machine-understandable. Using metadata to find distributed resources that meet one's functional requirements is only the first step. Resource requestors may have additional requirements such as security, survivability, or quality of service (QoS) specifications. For example, they may require resources to possess a certain military classification level, to originate from trusted sources, or to be handled according to a specified privacy policy. Therefore, resources need to be sufficiently annotated with security-related metadata so that they can be correctly discovered, compared, and invoked according to security as well as functional requirements of the requestor.

However, the resources are generally written using natural languages which are unstructured in nature. Given the inherent nuances of natural languages and the unstructured nature of text documents, domain knowledge plays a crucial role in improving the quality of text information retrieval. Ontology is a knowledge-management structure that represents domain knowledge in a structured and machine-interpretable form. It is increasingly being accepted as the key technology wherein key concepts and their inter-relationships are stored to provide a shared and common understanding of a domain across applications and hence ideally suited to aid in context analysis tasks. The use of ontological models to access and integrate large knowledge repositories in a principled way has an enormous potential to enrich and make accessible unprecedented amount of knowledge for reasoning [7].

One of the hurdles affecting the design for an exhaustive domain ontology is the absence of an unambiguous list of values that may be used to define concepts and relationships. Since, concept definitions need not be exhaustive as new or modified definitions of concepts may emerge, one of the ways to deal with this problem is to integrate ontology learning and enhancement process with text information retrieval based applications. This can help in learning and enhancing existing ontologies effectively within the rigid structural definition. This technique is suitable for learning concepts automatically from underlying resources and therefore sheds off a definite hurdle that acts as a major bottleneck for designing any ontology-based application. This can also take care of a fast-changing world, where areas of interest, vocabulary everything changes at a very rapid rate. Due to the dynamic nature of the security-related document repository, any agent or system designed for reasoning with these concepts should be able to adapt to changes and an ontology should be upgradeable with information extracted through text mining in the domain.

In this paper, we have proposed the design of a text mining framework to create a Security Information Management Ontology (SIMOnt). SIMOnt organizes security concepts and their inter-relationships in a structured and machine-readable format that can be used to annotate security resources in a consistent and effective manner. Starting with a seed ontology, the system extracts feasible concepts and their relationships in an automatic way from underlying text documents. This takes care of a fast-changing world, where areas of interest, vocabulary everything changes at a very rapid rate. The system is also equipped with an ontology enhancing mechanism, which enriches existing concepts and relationships extracted from resource documents after domain expert validation. Marking up security aspects of resources is a crucial step toward deploying a secure Service Oriented Architecture (SOA) system. Although, a number of researchers have noticed the need for security annotation of services and proposed a set of security-related ontologies, these ontologies lack the ability to express certain important security concepts, contain unnecessary concepts, and are organized in a non-intuitive way [9].

The rest of the paper is organized as follows. Section 2 presents a review of related work on ontology engineering and learning from text documents. Section 3 proposed the architectural and functional detail of the proposed SIMOnto (Security Information Management Ontology) framework. In section 4, we discuss the experimental setup and evaluation results. Finally, section 5 concludes the paper with future directions.

2 Related Works

Ontology learning is an area that has fascinated many researchers since last decade. Traditionally ontology development is a time consuming process that is not very conducive to changes in the domain knowledge and is not error-free; therefore, numerous frameworks and methodologies have been proposed to reduce the effort and resources required to develop ontology and provide more accurate results.

Luong *et al.* [1] have developed a framework for ontology development using text mining technique on web documents crawled in a focused way. Starting with a manually-created seed ontology, they have proposed the use of Support Vector Machine (SVM) to identify relevant documents for text mining process. Although, the proposed framework is fully automatic, they recommend for the domain experts intervention for improved overall performance. Lenci *et al.* [2] have also proposed a semi-automatic ontology extraction system from text documents. They have used NLP-based hybrid method that bring together linguistic and statistical techniques such that domain terminologies are extracted linguistically while the acquired terms are organized into proto-conceptual structures using NLP. Jiang and Tan [3] have presented an automatic domain ontology learning system that combines both statistical and lexico-syntactic features present in the contemporary systems and uses NLP tools for full text parsing and Parts-Of-Speech (POS) analysis. They have also designed rule-based algorithms to extract both taxonomical and non taxonomical relations. Nie and Zhou [4] have developed a domain-adaptable ontology learning system that uses an automatic seed concept learning by taking an overview of the domain material based on frequency. These core concepts are used to learn new concepts. Parkin et al. [5] have experimented with ontology development for information security based on associating infrastructure properties with human behavior to find the effects of information security controls on human behavior. The approach may be used to better manage information security by predict human behavior.

As can be seen from the literatures discussed above that one of the key issues that is still unresolved is filtering the domain specific terms (representing concepts) and their inter-relationship in an automatic way and therefore sheds off a definite hurdle that acts as a major bottleneck for designing domain ontologies. Further, it can also be observed that most of the authors recommend human intervention at concept validation level to preserve only relevant concepts and relationship rather than preserving everything identified by the system as feasible concepts – as ontology is not a data store rather a knowledge management tool.

3 Proposed SIMOnt Framework

Figure 1 presents the SIMOnt framework proposed for creating security information management ontology. Starting with seed ontology concepts, the system identifies relevant documents on the Web and crawls them on local machine. Thereafter, the documents are processed for cleaning, chunking, and parsing using a statistical parser. The outputs generated by parser are further analyzed to extract concepts and their relationships which are incorporated in seed ontology under expert supervision. The proposed framework follows the iterative process to learn concepts and relationships and continues until there is no change in the underlying ontology. This takes care of a fast-changing world, where areas of interest, vocabulary everything changes at a very rapid rate. The major modules of the framework are – *web crawler*, *document processor*, and *knowledge miner*. The functional details of these modules are discussed in the following sub-sections.

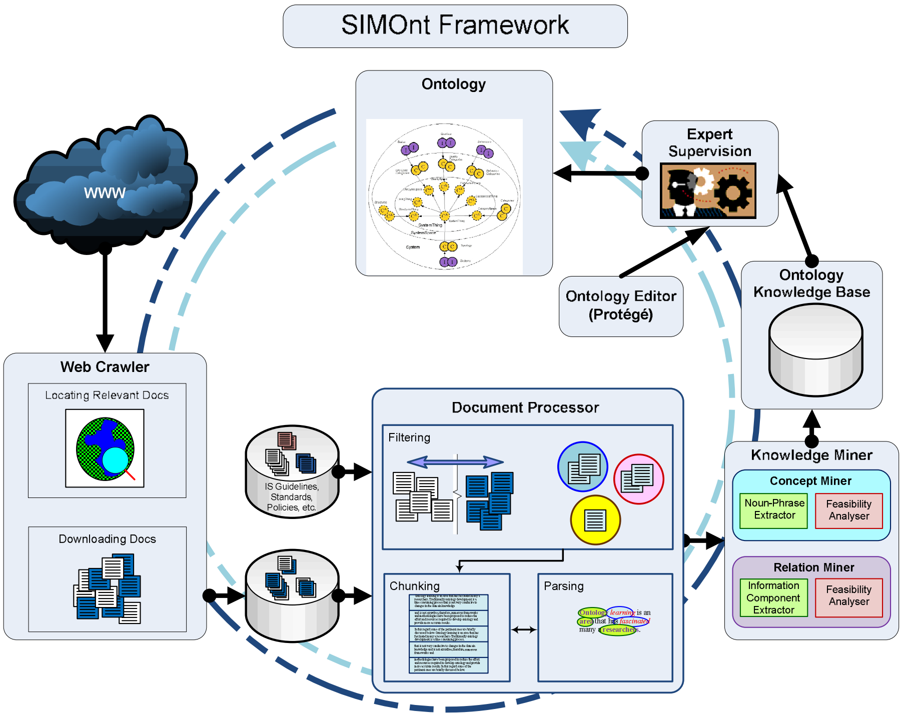

Figure 1. Design of SIMOnt Framework

3.1 Web Crawler and Document Processor

Since a document containing security-related information can be a text document (stored as txt, doc, or pdf files) or a webpage residing on the World Wide Web (WWW), the framework is integrated with a web crawler to crawl relevant web pages from WWW. The web crawler uses the ontological concepts and relations to identify the relevant web pages on WWW and store them on local machine for further processing by document processor to mine new concepts and relations.

The document processor fetches the text documents from local database repository and filters the unwanted texts from them. For example, in case of web pages the HTML tags are filtered out. Similarly, since we are concerned only with textual contents, all the images and their labels are excluded while converting pdf documents into text documents. After filtering process, the documents are divided into record-size chunks which boundaries are decided by the paragraph marks. Finally, the submitted to the Parts-Of-Speech (POS) analyzer which assigns POS tags to every word in a sentence, where a tag reflects the syntactic category of the word. The POS tags are useful to identify the grammatical structure of sentences like noun and verb phrases and their inter-relationships.

3.2 Knowledge Miner

The knowledge miner accepts phrase structure trees output by document processor as input and analyzes them to identify candidate constituents for knowledge-base. The candidate constituents are then analyzed using Latent Semantic Analysis (LSA) technique to compile a list of feasible concepts. Thereafter, the tree is again analyzed to identify relationships between concept-pairs.

3.2.1 Concept Miner

For candidate concepts, we consider only those internal NP (noun phrase) nodes in phrase structure tree whose child nodes appear as a leaf node. If a node NP has single child node tagged as *noun* it is extracted as *term* otherwise, string concatenation is applied to club the child nodes, tagged as *noun* or *adjective*, together and it is identified as *phrase*. The lists of terms and phrases are compiled separately for the purpose of feasibility analysis using LSA. After compiling the lists, the terms having a match in the list of stop-words are filtered out and phrases starting or ending with stop-words are cleaned. For remaining phrases we calculate their weight using term frequency (*tf*) and inverse document frequency (*idf*) in each document of the corpus. The weight of a phrase p_i in j^{th} document, $\omega(p_{i,j})$, is calculated using equations 1 and 2 where, $tf(p_{i,j})$ is the number of times p_i occurs in j^{th} document. |D| is the total number of documents in the corpus, and $|\{d_j : p_i \in d_j\}|$ is the number of documents where p_i appears. While counting frequency of a term or phrase they are stemmed using Porter's stemmer. All those phrases having normalized average weight over all documents above a threshold are retained for feasibility analysis using LSA.

$$\omega(p_{i,j}) = tf(p_{i,j}) \times idf(p_i) \tag{1}$$

$$idf(p_i) = \log\left(\frac{|D|}{\left|\left\{d_j : p_i \in d_j\right\}\right|}\right) \tag{2}$$

$$(p_{i,j}) = \begin{cases} idf\,(t_i), & if\ i = j\ and\ \ j \le m \\ idf\,(t_i), & if\ \ j > m\ and\ \ t_i\ is\ a\ substring\ \ of\ the\ phrase \\ 0, & otherwise \end{cases} \quad (3)$$

LSA is a technique which is used to analyze relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms [6]. For LSA each document d is represented as a feature vector $\vec{d} = \left(w_{t_1}, \cdots, w_{t_m}\right)$, where m is the number of terms, and w_{t_i} is the weight of term t_i in document d calculated using equation 1. Feature vector for each document in the corpus is used to generate *term-document* matrix A by composing feature vectors of all the documents in the corpus. In this matrix, a column vector represents a document and a row vector represents a term as document's feature. In matrix A all column vectors are normalized so that their length is 1. Thereafter, Singular Value Decomposition (SVD) is applied on A which breaks it into three matrices $U, S,$ and V such that $A = USV^T$. SVD translates the term and document vectors into a concept space. The first r columns of U (where r is $A's$ rank) form an orthogonal basis for the matrix $A's$ term space. Therefore, basis vectors, which are column vectors in U, represent abstract terms of corresponding document. In practice, it is not possible to take all r abstract terms. Therefore we take a threshold value, θ, and find the number of singular values (say k) in matrix S that is higher than this θ. Then, we use U_k, which consists of first k columns of U as shown in figure 3(e), to obtain k most important terms for the document corpus. At the time of identification of important terms and phrases we consider only magnitude therefore we take absolute value of U_k. Since the column vectors in U represent the importance of the terms for the document corpus, we also use U to evaluate the importance of phrases. For this, we construct a matrix P of order $m\times(m+p)$, where m and p represent the number of terms and phrases respectively. In matrix P, each row represents a term and columns represent terms as well phrases. Elements of matrix P are computed using equation 3. Like term-document matrix A, the column vector lengths in P are also normalized to 1. Thereafter, the matrix $abs(U_k^T)$ is multiplied with P to get matrix M which represents the importance of terms and phrases. In matrix M, the highest value in each row is identified and the corresponding term or phrase is extracted as feasible key concept.

3.2.2 Relation Miner

Once the list of feasible concepts is compiled, it is presented to the domain expert for validation. The human intervention is suggested just to ensure that ontology is not a generally data store rather a knowledge management tool. For each pair of validated concepts, we mine two different types of relations – structural relations and generic relations. The structural relations (IS-A, HAS-PART, etc.) are also called conceptual-semantic and lexical relations and extracted using WordNet, which is a large lexical database of English words. In WordNet, nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. The generic relations are extracted using co-occurrence based analysis.

4 Experimental Evaluation

In this section, we present the experimental details and performance evaluation results of the proposed SIMOnt framework. For performance evaluation a prototype model of SIMOnt is implemented using Java programming language. The prototype uses a number of in-built tools for different purposes. For example, it uses Stanford Parser for POS analysis and phrase structure tree creation. Similarly, it uses Porter Stemmer to stem the phrases while counting their frequencies, checking string containments, etc.

The performance of the framework is analyzed by taking into account the performance of the knowledge miner. The quality of knowledge mining can be measured by comparing its information extraction accuracy against human curation. Since, our intention was to study the applicability of the knowledge extraction from unstructured texts related to information security, we considered "the standard of good practice for information security" document which has been produced by the Information Security Forum (ISF). ISF is an international association of over 300 organizations which fund and co-operate in the development of a practical research programme in information security. As the ISF's document contains 372 pages and available as a pdf file, first it is converted into text file. Thereafter, it is divided into 4881 smaller files on the basis of paragraph markers and stored separately. The knowledge mining algorithm is applied on these files and the list of extracted concepts having relevance value (obtained through LSA) greater than or equal to a given threshold is presented for evaluation. We manually inspected these documents to build a complete compilation of all concepts to be extracted. Since, the number of documents are large, we have applied sampling to evaluate the results manually. The performance of this module is computed using standard measures of *precision* and *recall*, which are defined in equation 4 & 5 respectively. In these equations, *TP* indicates *true positive* which is defined as the number of correct concepts the system identifies as correct, *FP* indicates *false positive* which is defined as the number of incorrect concepts the system falsely identifies as correct, and the *FN* indicates *false negatives* which is the number of correct concepts the system fails to identify as correct.

$$precision = \frac{TP}{TP + FP} \tag{4}$$

$$recall = \frac{TP}{TP + FN} \tag{5}$$

The precision value of the system reflects its capability to identify a concept relevant to the domain. The precision of the proposed system is found to be 72.2%, which needs improvement. The precision value can be improved by tightening the rule set and enhancing the list of stop words. Recall value reflects the capability of the system to locate all instances of a concept within the corpus. The recall value of the knowledge miner module is 93.8%.

5 Conclusion and Future Works

In this paper, we have proposed the design of a text mining based Security Information Management Ontology (SIMOnt) framework. Starting with a seed ontology, SIMOnt retrieves relevant documents and applies natural language processing and statistical techniques to extract concepts and relationships from them. The mined concepts and relationships are then used to enhance the underlying ontology under expert supervision. For ontology editing, we have used protégé 4.1. Presently, we are enhancing the relation-mining technique to consider more complex relations that do not necessarily appear with relating concept-pair. We are also enhancing the framework to facilitate automatic annotation of resources using ontological concepts and answer security-related queries over them.

References

1. Luong, H. P., Gauch, S., & Wang, Q. (2009). Ontology Learning Through Focused Crawling and Information Extraction. In *Proceedings of the 2009 International Conference on Knowledge and Systems Engineering* (pp. 106-112). IEEE Computer Society. doi:10.1109/KSE.2009.28.
2. Lenci, A., Montemagni, S., Pirrelli, V., & Venturi, G. (2009). Ontology learning from Italian legal texts. In *Proceeding of the 2009 conference on Law, Ontologies and the Semantic Web: Channelling the Legal Information Flood* (pp. 75-94). IOS Press.
3. Jiang, X., & Tan, A. (2010). CRCTOL: A semantic-based domain ontology learning system. *J. Am. Soc. Inf. Sci. Technol.*, *61*(1), 150-168. doi:10.1002/asi.v61:1
4. Xuejun Nie and Jingli Zhou, “A Domain Adaptive Ontology Learning Framework,” *Networking, Sensing and Control, 2008. ICNSC 2008. IEEE International Conference on*, 2008, pp. 1726-1729.
5. S. E. Parkin, A. V. Moorsel, and R. Coles, “An information security ontology incorporating human-behavioural implications,” *Proceedings of the 2nd international conference on Security of information and networks*, Famagusta, North Cyprus: ACM, 2009, pp. 46-55.
6. Landauer, T., Foltz, P., and Laham, D.: Introduction to Latent Semantic Analysis, *Discourse Processes, 25,* pp. 259–284, 1998.
7. Fensel, D., Horrocks, I., Harmelen, F. van, McGuinness, D. L. & Patel-Schneider, P.: 2001, March/ April, OIL: Ontology Infrastructure to Enable the Semantic Web, *IEEE Intelligent Systems 16(2)*, 38-45.
8. Anya Kim, Jim Luo, Myong Kang, Security Ontology for Annotating Resources, Naval Research Lab, Washington DC, August 31, 2005.
9. Denker, G., Kagal, L., Finin, T., Paolucci, M., and Sycara, K. (2003). Security for DAML Web Services: Annotation and Matchmaking. In *Proc. of the 2nd International Semantic Web Conference (ISWC2003):* Sanibel Island, Florida.