# Automatic Top-Down Role Engineering Framework Using Natural Language Processing Techniques

Masoud Narouei(✉) and Hassan Takabi

Department of Computer Science and Engineering,
University of North Texas, Denton, TX, USA
masoudnarouei@my.unt.edu, takabi@unt.edu

**Abstract.** A challenging problem in managing large networks is the complexity of security administration. Role Based Access Control (RBAC) is the most well-known access control model in diverse enterprises of all sizes because of its ease of administration as well as economic benefits it provides. Deploying such system requires identifying a complete set of roles which are correct and efficient. This process, called role engineering, has been identified as one of the most expensive tasks in migrating to RBAC. Numerous bottom-up, top-down, and hybrid role mining approaches have been proposed due to increased interest in role engineering in recent years. In this paper, we propose a new top-down role engineering approach and take the first step towards extracting access control policies from unrestricted natural language requirements documents. Most organizations have high-level requirement specifications that include a set of access control policies which describes allowable operations for the system. It is very time consuming, labor-intensive, and error-prone to manually sift through these natural language documents to identify and extract access control policies. We propose to use natural language processing techniques, more specifically Semantic Role Labeling (SRL) to automatically extract access control policies from these documents, define roles, and build an RBAC system. By successfully applying semantic role labeling to identify predicate-argument structure, and using a set of predefined rules on the extracted arguments, we were able correctly identify access control policies with a precision of 79%, recall of 88%, and $F_1$ score of 82%.

**Keywords:** Role Based Access Control · Role engineering · Semantic role labeling · Natural language processing · Privacy policy

## 1 Introduction

In computer security, access control is the selective restriction of access to resource. System administrators at the top of any organization ascertain which individuals will be given access to what type of information. Access Control Policies (ACPs) detail controlling access to information and systems. These controls can be exemplified as the management of a number of key issues, including user access, network access controls, passwords, operating system software controls, and higher-risk system

R.N. Akram and S. Jajodia (Eds.): WISTP 2015, LNCS 9311, pp. 137–152, 2015.
DOI: 10.1007/978-3-319-24018-3_9

access; giving access to files and documents and controlling remote user access; and restricting access.

However, defining proper ACPs is challenging, especially for large organizations. Advanced access control models such as Role Based Access Control (RBAC) [41] promise long-term cost savings through reduced management effort, but manual development of initial policies can be very time consuming, labor-intensive, and error prone [4, 24]. RBAC is the most widely used model for advanced access control in diverse enterprises of all sizes. In RBAC, access permissions are associated with roles instead of users where roles represent functions within a given organization. Users can activate a subset of the roles which they are members of and easily acquire all the required permissions for those roles. Deploying an RBAC system requires identifying a complete, correct and efficient set of roles, and then assigning users and permissions to those roles [44]. This process is known as role engineering and is the most expensive component of an RBAC implementation [19].

There are mainly two approaches to role engineering: the top-down approach and the bottom up approach. The top-down approach takes advantage of a detailed analysis of business processes where organizational business processes are analyzed, particular job functions are defined and decomposed into smaller units. Once the required permissions for performing specific tasks are identified, they can be grouped into appropriate functional roles. The process is repeated until all the job functions are covered. Because of the large number of business processes, users and permissions in an organization, and also as such a process is human-intensive, it is a rather difficult task and hence believed to be slow, expensive, and not scalable. In order to overcome this drawback, researchers have proposed a bottom-up approach to use data mining techniques to discover roles from existing data. Since many organizations already have user-permission assignments defined in some form, it makes sense to identify roles from this existing information. This approach first considers the existing users' permissions before RBAC is implemented, and aggregates them into roles. Such a bottom-up approach is called role mining.

Role mining has raised significant interests in the research community and in recent years, numerous role mining techniques have been developed [17, 18, 23, 34]. While role mining can quickly combine existing permissions into roles, it often leads to roles that are difficult to understand and manage because they don't have business meaning and fail to reflect the business structure of the organization [34]. In order to mitigate this problem, researchers have proposed hybrid role mining techniques that incorporate both top-down and bottom-up approaches [34, 23]. The hybrid role mining approach derives roles not only from the user-permission assignments but also using certain top-down information. This methodology of role development creates roles that are not simply collections of permissions, but are semantically meaningful and bear relevance to the organizational structure. Hybrid role mining generates semantically meaningful roles that are understandable and relevant to practical scenarios and hence makes adoption of RBAC more acceptable to organizations.

In this paper, we propose a substantially different top-down approach to role mining and take the first steps towards using natural language processing techniques to extract policies from unrestricted natural language requirements documents.

Most organizations have high-level requirement specifications that determine how information access is managed and who, under what circumstances, may access what information [24]. These documents define security policies and include a set of access control policies which describes allowable operations for the system. All US federal agencies are required to provide information security by the "Federal Information Security Act of 2002" [14], and policy documentation is part of that requirement [33]. Although private industry is not required to provide such documentation, the significant cost associated with cyber-attacks has led many companies to document their security policies as well. Besides, having security policies documented makes it much easier for organizations to transition from access control lists (ACLs) into a more robust RBAC infrastructure. We refer to these documents (high-level requirement specifications) as Natural Language Access Control Policies (NLACPs) which are defined as "*statements governing management and access of enterprise objects. NLACPs are human expressions that can be translated to machine-enforceable access control policies*" [24]. However, NLACPs are not directly implementable in an access control mechanism as they are normally expressed in human understandable terms. They are unstructured and may be ambiguous and thus hard to convert to formally actionable elements, so the enterprise policy may be difficult to encode in a machine-enforceable form. It is very time consuming, labor-intensive, and error-prone to manually sift through these existing natural language artifacts to identify and extract the buried ACPs. Properly enforcing these security policies requires the ACPs to be translated to machine-readable policies which has been done manually and is a very labor intensive and error prone process [4, 25]. Our goal is to automate this process to reduce manual efforts and human errors. We propose to develop techniques and tools that will support effective development of trustworthy access control policies through automatically extracting formal access control policies from unrestricted natural language documents and transforming them to enforceable policies. Our goal is to allow organizations to use existing, unconstrained natural language texts such as requirements documents for inferring ACPs. Our approach consists of five main steps: (1) apply linguistic analysis to parse natural language documents and annotate words and phrases in sentence (lexical parser), (2) identify whether a sentences contain potential ACP content or not (ACP sentence identification) , (3) infer semantic arguments of each predicate in each sentence using annotated words and phrases (semantic parser), (4) transform these semantic arguments into ACPs (postprocessor), and (5) aggregate the extracted ACPs into roles (role extractor). Our approach automatically generates machine enforceable ACPs and could be used as standalone top-down approach or as hybrid approach in combination with bottom-up role mining approaches.

In this paper, we limit our discussions to the linguistic analysis of natural language documents and extracting semantic arguments of each predicate from each sentence. We also present initial results of applying the technique to a sample of our policy dataset. To the best of our knowledge, there is not much work in the literature that addresses this issue and this is the first report on effectiveness of applying semantic role labeling to large and diverse set of ACPs.

Our contributions in this paper are three-fold:

- We propose an automated top-down role engineering approach;
- We apply semantic role labeling to identify access control policies in unrestricted natural language documents;
- We perform experiments to show efficiency of the proposed approach. Our evaluation results show that the proposed approach can effectively identify access control policies with a precision of 79%, recall of 88%, and $F_1$ score of 82%.

The rest of this paper is organized as follows: We start with an overview of previous literature in section 2. In section 3, we present our proposed approach and its components. The experiments and results are presented in section 4, and finally, conclusion and future works wraps up the paper.

## 2 Background and Related Work

This section describes the state of the art in NLP techniques and their application for access control policies and related areas as well as hybrid role mining approaches in the literature.

### 2.1 Natural Language Processing (NLP)

Most modern NLP semantic parsers include several tasks such as tokenization, sentence segmentation, part-of-speech (POS) tagging, lemmatization, named-entity recognition, syntactic parsing, semantic role labeling, event recognition, and coreference resolution. *Tokenization* detects individual words, punctuation, and other items from the text. *Sentence segmentation* identifies the boundaries of sentences. *Part-Of-Speech (POS) tagging* determines the POS tags such as noun, verb, adjective, etc. for each token. The current state-of-the-art POS taggers achieve 97.3% accuracy for individual tokens [30]. *Lemmatization* generates the common root word for a group of morphologically related words. For instance, sang, sung, and sings are all forms of a common lemma “sing”. The state of the art achieves around 99% accuracy for the English language [20]. *Named-entity recognition (NER)* aims to classify phrases into entity types such as people, organizations, locations, times, vehicles, and events [26]. The state-of-the-art for the NER general task has a $F_1$ score of around 89% [29]. *Syntactic parsing* generates a parse-tree structure for a sentence [26]. The tree structure provides a basis for other tasks within NLP such as question answering, information extraction (IE), and machine translation. State-of-the-art parsers have a $F_1$ score of around 90% [53]. *Coreference resolution* determines whether or not two expressions in a document refer to the same entity or event. A common subset of this problem occurs within extracting ACPs from NLACP texts in that pronouns must be resolved to their antecedents (the actual role or resource). *Kennedy et al.* introduced an algorithm to resolve pronoun anaphora resolution (match the correct noun to a pronoun) that does not require parsing and achieves 75% accuracy on their test set [28].

## 2.2 Information Extraction (IE)

Information extraction creates structured data from text [26]. Common targets of IE applications include named-entities, other entities of specific types, relations, events, and their attributes. A relation expresses the relationship among two or more items. Common relation types include "is-a" and "part-of". For example, "a doctor is a healthcare professional (HCP)" is represented by *is_a(doctor, HCP)* and "medical records contain family history" is represented by *contains(medical record, family history)*. State-of-the-art systems for relation extraction (RE) typically have around 85% precision and 70% recall [38]. Another IE technique is shallow parsing (semantic role labeling) which involves identifying the different predicates (verbs) in a sentence along with their semantic arguments [21].

## 2.3 NLP Techniques for Privacy Policies

*Breaux et al.* have manually analyzed privacy policies to map natural language policy statements into frame-based and first-order logic representations [5, 6, 8]. They have also analyzed regulatory text and developed natural language heuristics, some expressible as simple regular expressions, that can be used to identify frame-based representations of actions [7] and whether actions on information are permitted, required or prohibited with various conditions, exceptions and purposes [9]. *Ammar et al.* conducted an experiment to use NLP methods and crowdsourced annotations from the "Terms of Service; Didn't Read" project [47] to train a classifier to answer a single question: whether a privacy policy is considered clear by humans about a particular set of procedures pertaining to sensitive user data [1]. The ongoing Usable Privacy Policy Project aims to build on recent advances in NLP, privacy preference modeling, crowdsourcing, and formal methods to semi-automatically extract key privacy policy features from natural language website privacy policies [48]. The focus of this project is website privacy policies while our project is focused on access control policies.

## 2.4 Controlled Natural Language (CNL) and Access Control

*Schwitter* defined a Controlled Natural Language (CNL) as "an engineered subset of a natural language whose grammar and vocabulary have been restricted in a systematic way in order to reduce both ambiguity and complexity of full natural language." [42]. While a CNL provides semantic interpretations, it limits policy authors to the defined grammar and requires language-specific tools to stay within the language constraints. The SPARCLE Policy Workbench [10, 27, 11] employs shallow parsing technology [35] to extract privacy policies based on a pre-defined controlled grammar for forming policies in a structured form. The policies are then translated to a machine-readable form, such as EPAL [2] and XACML [36]. *Inglesant et al.* proposed a similar tool, PERMIS, which used a role-based authorization model [25]. However, they reported issues with users not comprehending the predefined "building blocks" imposed by using a CNL. *Shi et al.* presented their approach to authoring policies using a CNL and showed the improved usability of CNL interface [43]. However, their approach is limited in the complexity of the rules that could be created since their supporting tool did not support conditions such as previous actions that must be taken before a user could access data.

## 2.5 NLP and Access Control

NL sources have been analyzed to infer and generate ACPs. *Fernandez et al.* presented a basic overview of extracting RBAC from use cases [15]. *Fontaine* proposed an approach based upon goal-based requirements engineering to extract authorization and obligation rules from NL texts into a policy language [16]. *He et al.* proposed an approach to generate ACPs from NL based upon available project documents, database design, and existing rules [22]. Using a series of heuristics, developers manually analyze the documents to find ACPs whereas our approach seeks to automatically extract ACPs. *Xiao et al.* proposed Text2Policy for automated extraction of ACPs [49]. It first uses shallow parsing techniques with finite state transducers to match a sentence into one of four possible access control patterns. If such a match can be made, Text2Policy uses the annotated portions of the sentences to extract the subject, action, and object from the sentence. *Slankas et al.* proposed Access Control Rule Extraction (ACRE) [45] which applies inductive reasoning to find and extract ACRs while Text2Policy applies deductive reasoning based upon existing rules to find and extract ACRs. While this work these two early works are promising, they suffer from several weaknesses. ACRE uses a supervised learning approach to identify sentences containing ACRs which requires a labeled dataset similar in structure and content to the document being analyzed. This data is hard to come by. Text2Policy does not require a labeled data set but it misses ACRs that do not follow one of its four patterns. It is reported that only 34.4% of the identified ACR sentences followed one of Text2Policy's patterns [44]. Additionally, Text2Policy's NL parser requires splitting longer sentences as the parser cannot handle complicated sentence structures. These approaches assume all necessary information for an ACP is contained within the same sentence, and they do not handle resolution issues. Neither one of these approaches take into account the presence of contextual information or environment conditions which is a very challenging task.

## 2.6 Top-Down Role Engineering

*Roeckle et al.* described a process oriented approach for role-finding to implement Role-Based Security Administration. The core of their work is presenting the data model, which integrates business processes, role based security administration and access control. Moreover, a structured top-down approach is outlined which is the basis for derivation of suitable business roles from enterprise process models [40]. *Baumgrass et al.* identified several tasks in role engineering that are monotonous, time-consuming, and can get tedious if conducted manually. These tasks include the derivation of candidate RBAC artifacts from business processes and scenario models. They presented an approach to automatically derive role engineering artifacts from process and scenario models. They especially discuss the derivation of role engineering artifacts from UML activity models, UML interaction models, and BPMN collaboration models. In particular, they use the XMI (XML Metadata Interchange) representation of these models as a tool and vendor independent format to identify and automatically derive different role engineering artifacts [3]. *Molloy et al.* propose a hybrid role engineering approach where a set of roles has already been derived by the top-down approach and the remaining roles are defined following a bottom-up technique [34]. In this approach, roles that correspond to sensitive job responsibilities are specified manually by the top-down procedure. The proposed hybrid approach combines traditional role mining techniques with an approach that optimizes an existing set of roles. *Hernandez et al.* propose a hybrid role mining method which creates roles from both top-down and bottom-up information collected from a number of sources [23]. Two criteria, one based on the top-down information and the other based on the bottom-up information are used to assign roles to the appropriate users. The bottom-up information is the user permission assignments whereas the top-down information is the various attributes of users.

*Frank et al.* propose a probabilistic method for hybrid role mining [18]. Their proposed method consists of two steps - (i) identification of business information relevant to the existing user-permission assignments and (ii) including this business information in the process of role mining. Incorporation of business information is achieved by satisfying two objectives: (i) finding a decomposition consisting of UA and PA that best describes the UPA even if new users are added and (ii) agreement of the resulting role assignments with the relevant business information.

## 3 The Proposed Framework

In order to construct a formal model for an NLACP, we must extract the necessary elements of ACPs from the natural language document. The ACPs describes who has access to what resource in what way. By processing these formal models, our technique will generate corresponding machine readable and enforceable policies. An overall view of the proposed system is shown in Figure 1. In the following sections, we describe each of these steps in details.

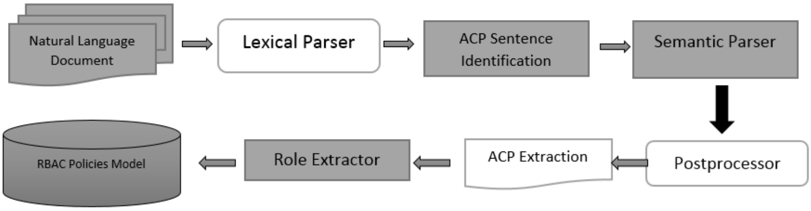

**Fig. 1.** Overview of the Proposed Framework

## 3.1 Lexical Parser

We first read the entire NLACPs and perform sentence segmentation and tokenization. Sentence segmentation identifies the boundaries of sentences whereas Tokenization detects individual words, punctuation, and other items from the text. In order for the input text to be ready for evaluation, the first step is to identify all sentences and separating these sentences by a period and a carriage return so that each sentence will be on a separate line. For this purpose, we use CoreNLP tool kit [31].

**Coreference Resolution.** Coreference resolution (sometimes written co-reference) determines whether or not two expressions in a document refer to the same entity or event. The goal is identify all expressions that refer to the same entity in a text. For example, consider the following sentence where HCP stands for healthcare professional:

The **HCP** opens the message to which **he** or **she** wishes to reply.

Here, "HCP," "he," and "she" all refer to the same entity. The goal of coreference resolution here is to replace all of pronouns with their corresponding referents. Because each sentence will be evaluated separately, having a clear idea of each pronoun is a key point in identifying the correct ACP elements. We adopt the approach proposed in [12], which is a fast and robust algorithm for this purpose.

## 3.2 ACP Sentence Identification

Often time NLACPs contain contents that describe functional requirements and are not necessary related to ACPs. Although these documents also contain ACPs, attempting to extract ACPs from the whole document is an error prone and tedious process. To correctly extract ACPs from NLACPs, it is very important to find out those sentences that have potentially ACP content and then perform further analysis only on those sentences to extract ACP elements.

*Slankas et al.* proposed a *k*-Nearest Neighbors (*k*-NN) classifier to identify sentences containing ACPs. *k*-NN is an instance based classifier that attempts to locate the *k* nearest neighbors of an instance in an instance space and labeling that instance with the same class as that of most neighbors. As our focus is on correctly identifying ACP elements, we employ the same approach as the one used in [45].

## 3.3 Semantic Parser

We use semantic role labeling (SRL) to automatically identify predicate-argument structure in ACP sentences. SRL, sometimes also called shallow semantic parsing, is a task in natural language processing consisting of the detection of the semantic arguments associated with the verb (or more technically, a predicate) of a sentence and their classification into their specific roles. It labels verb-argument structure using the notation defined by Propbank [37] project, identifying who did what to whom by assigning roles to constituents of the sentence representing entities related to a specific verb. Recognizing these semantic arguments is a key task in finding the answer to the questions like: "Who," "When," "What," "Where," "Why", etc., which are especially in use by analyzers trying to extract access control policies from sentences. The following sentence, exemplifies the annotation of semantic roles:

[$_{\text{Arg0}}$ John] [$_{\text{ArgM-MOD}}$ can] [$_{\text{V}}$ **assign**] [$_{\text{Arg1}}$ clerk ] [$_{\text{Arg2}}$ to users from department A]

Here, the roles for the predicate **assign** (assign.01, that is, the *roleset* of the predicate) are defined in the PropBank Frames scheme as:

**V:** verb
**ArgM-MOD:** modal
**Arg0:** (assigner)
**Arg1:** (thing assigned)
**Arg2:** (assigned to)

SRL is very important in making sense of the meaning of a sentence. Such semantic representation is at a higher-level of abstraction than a syntax tree. For instance, the sentence "A professor can review the same project at most one time" has a different syntactic form, but the same semantic roles to "The same project can be reviewed by a professor at most one time".

In general, given any sentence, the task of SRL consists of analyzing the propositions expressed by all target verbs in a sentence and for each target verb, all the constituents in that sentence which fill a semantic role of the verb, will be extracted. Here, we use the following notation to describe ACPs:

$$\{A;\ B;\ C\}$$

Where *A* stands for *Argument0*, *B* stands for *predicate* and *C* stands for *argument1*. *Argument0* usually denotes agent or experiencer for that predicate and *Argument1* denotes theme (where predicate affects). In this paper, we use Senna (Semantic/syntactic Extraction using a Neural Network Architecture) semantic role labeler [13], which performs sentence-level analysis. Senna is a multilayer neural network architecture that can handle a number of NLP tasks such as part-of-speech tagging, chunking, named entity recognition, and semantic role labeling with both speed and accuracy. Senna relies on large unlabeled dataset(s) and allows the training algorithm to discover internal representations that prove useful for the requested task. Senna is fast because it uses a simple architecture, it is self-contained because it does not rely on the output of another system, and it is accurate because it offers state-of-the-art or near state-of-the-art performance.

### 3.4 Postprocessor

After generating predicate-arguments using SRL tool, additional processing is required on the output. This is due to the fact that the NLACPs are typically stated by managers using their own language and grammar because they do not have the technical knowledge of the system, and this makes ACP extraction from their stated sentences more complicated. In order to increase accuracy of the extracted ACPs, we apply named entity recognition and argument expansion to the SRL output as described below.

**Named Entity Recognition.** Named entity recognition (NER) is the task of identifying named entities and sequences of words in a text belonging to predefined categories such as the names of persons, locations, organizations, expressions of times, quantities, monetary values, percentages, etc. The task here is to produce an annotated text that highlights the names of entities such as the following example:

[$_{\text{ORGANIZATION}}$ Customer Service Reps], [$_{\text{PERSON}}$ Pharmacists], and [$_{\text{ORGANIZATION}}$ Billing Reps] can collect and use customer name and [$_{\text{TIME}}$ date of birth] to help confirm identity.

In this example, *Customer Service Reps* is an organization consisting of three tokens, *Pharmacists*, is a person consisting of one token, *Billing Reps* is an organization consisting of two three tokens, and finally *date of birth* is a time consisting of three tokens.

**Argument Expansion.** ACPs usually do not conform to a predefined template unless there is controlled grammar being used. Most of the time ACPs are stated by managers using their own language because they do not have the technical knowledge of the system and this makes their stated sentences complicated. One of these complications is that sometimes more than one ACP is stated in a given sentence. Consider the following sentence for example:

*Customer Service Reps, Pharmacists, and Billing Reps can collect and use customer name and date of birth to help confirm identity.*

There are 15 different ACPs associated with this sentence:

- {customer service rep; collect; customer name}
- {customer service rep; collect; customer date of birth}
- {customer service rep; use; customer name}
- {customer service rep; use; customer date of birth}
- {pharmacist; use; customer name}
- {pharmacist; use; customer date of birth }
- {pharmacist; collect; customer name}
- {pharmacist; collect; customer date of birth }
- {billing rep; collect; customer name}
- {billing rep; collect; customer date of birth }
- {billing rep; use; customer name}
- {billing rep; use; customer date of birth }
- {customer service rep; confirm; identity}
- {pharmacist; confirm; identity}
- {billing rep; confirm; identity}

Now consider the following list of the extracted roles for predicate Collect:

[$_{\text{Arg0}}$ Customer Service Reps, Pharmacists, and Billing Reps] [$_{\text{ArgM-MOD}}$ can] [$_{\text{v}}$ **collect**] and use [$_{\text{Arg1}}$ customer name and date of birth] [$_{\text{ArgM-PNC}}$ to help confirm identity].

As a comparison between the generated semantic arguments and the actual ACPs shows, SRL's output can be interpreted as an abstract form for ACPs, so we have to expand this abstract form to generate all of the related ACPs. This expansion could be in the form of extracting all named entities or other standalone nouns in Arg0 as the possible agents and also extracting independent entities from Arg1 as themes. For example, in this case, Named Entity Recognizer identifies *Customer Service Reps* and *Pharmacists* as organizations. After extracting all of these entities, we list all of the combinations of these entities based on each predicate. For example this verb-argument listing can be expanded as the following rules:

- {customer service rep; collect; customer name}
- {customer service rep; collect; customer date of birth}
- {pharmacist; collect; customer name}
- {pharmacist; collect; customer date of birth }
- {billing rep; collect; customer name}
- {billing rep; collect; customer date of birth }

### 3.5 Role Extractor

When the SRL tool extracts the ACP components, role extractor utilizes the extracted information to define roles. Then, these roles can be used to build an access control models such as RBAC. The ACPs are in the form of {subject, object, operation} and many of the extracted subjects correspond with the job functions within organization (e.g. doctor, pharmacist, nurse, healthcare professional, etc.) which could represent roles.

A naïve approach would be to just look at the ACPs and find the ones with the same subject and group them together in one role and use all the ACPs with that subject to build the role permission assignment relationships. The object and operation elements of the ACPs are used to define permissions in RBAC and then assign those permissions to roles based on that specific subject using ACP associations. Another approach is to use classifier such as *k*-Nearest Neighbors (*k*-NN) classifier or Naive Bayes classifier to extract roles from the ACPs. However, we leave this to future work as it is not focus of this paper.

## 4 Experimental Results

In this section, we perform experiments to answer the research question of how effectively the subject, object, and operation elements of ACPs are extracted. In the followings, we explain the datasets and the evaluation criteria used in our experiments. Then, we present the experimental results.

### 4.1 Datasets

We use documents from multiple domains such as electronic healthcare, educational, and conference management for the experiments. For the electronic health care domain, we use iTrust [32], which is an open source healthcare application that includes various features such as maintaining medical history of patients, identifying primary caregivers, storing communications with doctors, and sharing satisfaction results. For the educational domain, we employ use cases from the IBM Course Registration System used in previous research [39]. For the conference management domain, we use documents from CyberChair [46], which has been used by hundreds of different conferences and workshops. We also use a combined document of 115 ACP sentences collected from 18 sources (published papers, public web sites, etc.) that has been used in previous research [49, 45]. In this paper, we only consider those sentences that are labeled as access control sentence and ignore the rest as ACP identification is not the focus of this paper. For our evaluation, we use the iTrust data set that was used by *Xiao et al.* [27]. This version includes the preprocessed iTrust data set consisting of simplified sentences. For iTrust, there are 418 sentences identified as containing ACP content. The second dataset, IBM Course Registration System consists of eight use cases and there are 169 ACP sentences. The CyberChair dataset consists of 139 ACP sentences and the for Collected ACP documents, there are 115 ACP sentences.

### 4.2 Evaluation Criteria

We want to know how effectively the semantic arguments of each predicate are extracted from ACP sentences. The results are evaluated with respect to *recall*, *precision*, and the $F_1$ measure of the predicate arguments. To calculate these values, we categorize the extractions into four categories: false positives (FP) are cases where we mistakenly identify a word as an ACP element when it is not, false negatives (FN) occur when we fail to correctly extract an actual ACP element, true positives (TP) are correct extractions, and true negatives (TN) are cases where we correctly identified that a word in the sentence was not an ACP element. From these values, we define precision as the proportion of correctly extracted ACP elements against all extractions against the test data. We also define recall as the proportion of ACP elements found for the current data under test. The $F_1$ score is the harmonic mean—a weighted average of precision and recall—giving an equal weight to both recall and precision. $F_1$is computed by equation 1.

$$F_1 = 2 \times \frac{Recall \ \times Precision}{Recall + Precision} \tag{1}$$

In order to evaluate the effectiveness of our proposed approach, we use the datasets that were manually labeled by *Slankas et al.* [45]. They were able to find a total of 1070 ACPs in iTrust dataset, 375 ACPs in IBM Course Registration System dataset, 386 ACPs in CyberChair dataset and 258 ACPs in Collected ACP documents. More details on how the labeling was done can be found in [45]. The evaluation results as well as comparison with the most recent system (ACRE) are presented in Table 1. As the results show, our approach based on semantic role labeling performs very well and outperforms the ACRE approach in most cases. The algorithm used in ACRE requires repetition in sentence structure as well as subjects and resources throughout the document to perform well. This algorithm performed best on iTrust because it contained repetitions throughout the document but performed poor on the Collected ACP document. That's because there are not enough repetition in that document for finding initial set of known subjects and resources and expanding the patterns. However, semantic role labeling does not require repetition as every sentence will be considered separately, independent of the other sentences. As long as there are role sets defined for that predicate, semantic role labeling can find most of the arguments. This is why the results for semantic role labeling are stable throughout all documents and it provides good results regardless of the structure of the document.

**Table 1.** Comparison of ACP extraction between ACRE and the proposed system (ICM: IBM Course Registration, CC: CyberChair and CAD: Collected ACP Documents).

| Dataset | ACRE | | | SRL | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | $F_1$ | Precision | Recall | $F_1$ |
| iTrust | 80% | 75% | 77% | 75% | 88% | 80% |
| IBM | 81% | 62% | 70% | 54% | 87% | 58% |
| CC | 75% | 30% | 43% | 46% | 84% | 59% |
| CAD | 68% | 18% | 29% | 79% | 86% | 82% |

In terms of precision, however, our approach does not perform very well. One issue with using semantic role labeling is that it extracts all arguments for all of the verbs in a sentence. Sometimes only a portion of these verbs such as (set, add, etc) describe access control policies. Consider the following example:

*Only the manager [$_v$ **is**] [$_v$ **allowed**] to [$_v$ **add**] a new resident to the system and to [$_v$ **start**] or [$_v$ **update**] the care plan of a resident.*

Here, only three of the verbs, namely *add*, *start* and *update* address ACPs. In the experiments, we eliminate "To Be" and "Modal" verbs because usually they are part of other verbs such as *can assign* and do not express ACPs on their own. There are also other verbs such as *click*, *include*, etc., that do not express ACPs and hence increase the false positive rate. In the future, we plan to create a dictionary of the verbs that are associated with ACPs and will only consider those verbs which will improve the results significantly.

Another issue with our approach is that sometimes the SRL tool is unable to correctly identify all predicates and their arguments. This is due to complex structure of some sentences. We plan to define specific rules to handle this issue and improve the precision.

Although our approach does not perform very well in terms of precision, if we consider the $F_1$ scores, we can see that our approach outperforms the ACRE and for some dataset(s) the difference is very significant (82% compared to 29% for the Collected ACP documents). Only for the IBM Course Management dataset, SRL is outperformed by ACRE and it is because precision is very low which leads to lower $F_1$ score. In addition to offering better recall and $F_1$ score, another advantage of our approach over the ACRE is that it does not require any labeled data set whereas ACRE uses a supervised learning approach and requires a labeled dataset similar in structure and content to the document being analyzed to setup the classifiers. One technical challenge concerning the use of SRL is that sometimes our tool is unable to find the predicate-arguments in some sentences. The reason is that SRL tools are often trained on publicly available corpora such as the Wall Street Journal. This means that the predicate-argument frames are usually general and not well suited for processing information such as access control requirements documents. In the future, we plan to address this issue by adapting the SRL tool to ACP domain so that it can identify all predicate-arguments.

## 5 Conclusion and Future Work

In this paper, we proposed a new top-down approach towards role engineering in order to extract access control policies from unstructured natural language documents. We applied semantic role labeling techniques to extract policies from natural language requirements documents. The semantic role labeling allowed us to identify predicate-argument structure and by applying a set of predefined rules on the extracted arguments, we were able to successfully identify ACP elements with a recall of 88%, precision of 79%, and $F_1$ score of 82%. The performance of our system depends on the predefined role sets for each predicate. Currently, the proposed approach considers all predicates in the sentence which results in large number of false positives. In the future, we plan to create a dictionary of the verbs that are usually associated with access control policies and only consider those verbs to improve the results. We also plan to implement the complete system including implementing ACP sentence identification step and role extractor components.

## References

1. Ammar, W., Wilson, S., Sadeh, N., Smith, N.: Automatic Categorization of Privacy Policies: A Pilot Study. School of Computer Science, Language Technology Institute, Technical Report CMU-LTI-12-019, December 2012
2. Ashley, P., Hada, S., Karjoth, G., Powers, C., Schunter, M.: Enterprise Privacy Architecture Language (EPAL 1.2) (2003). http://www.w3.org/Submission/EPAL/
3. Baumgrass, A., Strembeck, M., Ma, S.R.: Deriving role engineering artifacts from business processes and scenario models. In: Proceeding of ACM SACMAT 2011, June 15–17, Innsbruck, Austria, pp. 11–20 (2011)
4. Beckerle, M., Martucci, L.A.: Formal definitions for usable access control rule sets from goals to metrics. In: Proceedings of the Ninth Symposium on Usable Privacy and Security (SOUPS), pp. 2:1–2:11. ACM (2013)
5. Breaux, T.D., Antón, A.I.: Deriving semantic models from privacy policies. In: 6th IEEE International Workshop on Policies for Distributed Systems & Networks, pp. 67–76 (2005)
6. Breaux, T.D., Antón, A.I.: Analyzing goal semantics for rights, permissions and obligations. In: Proc. IEEE 13th International Requirements Engineering Conference (RE 2005), Paris, France, pp. 177–186, August 2005
7. Breaux, T.D., Antón, A.I.: Analyzing regulatory rules for privacy and security requirements. IEEE Transactions on Software Engineering, Special Issue on Software Engineering for Secure Systems (IEEE TSE) **34**(1), 5–20 (2008)
8. Breaux, T.D., Antón, A.I., Doyle, J.: Semantic parameterization: a process for modeling domain descriptions. ACM Transactions on Software Engineering Methodology (ACM TOSEM) **18**(2), Article 5 (2008)

9. Breaux, T.D.: Legal Requirements Acquisition for the Specification of Legally Compliant Information Systems. Ph.D. Thesis, North Carolina State University, April 2009
10. Brodie, C.A., Karat, C.-M., Karat, J., Feng, J.: Usable security and privacy: a case study of developing privacy management tools. In: Proc. SOUPS 2005 (2005)
11. Brodie, C.A., Karat, C.-M., Karat, J.: An empirical study of natural language parsing of privacy policy rules using the SPARCLE policy workbench. In: Proc. SOUPS 2006, pp. 8–19 (2006)
12. Charniak, E., Elsner, M.: EM works for pronoun anaphora resolution. In: Proceedings of the European Chapter of the ACL (2009)
13. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural Language Processing (Almost) from Scratch. Journal of Machine Learning Research (JMLR) (2011)
14. Federal information security management act of 2002. Title III of the E-Government Act of 2002 (Public Law 107-347) (2002)
15. Fernandez, E.B., Hawkins, J.C.: Determining role rights from use cases. In: Proc. ACM Workshop on Role-Based Access Control 1997, pp. 121–125 (1997)
16. Fontaine, P.J.: Goal-Oriented Elaboration of Security Requirements. Université catholique de Louvain (2001)
17. Frank, M., Basin, D., Buhmann, J.M.: A class of probabilistic models for role engineering. In: Proc. 15th ACM Conference on Computer and Communications Security (CCS) 2008, pp. 299–310 (2008)
18. Frank, M., Buhmann, J.M., Basin, D.A.: Role mining with probabilistic models. ACM Transactions on Information and System Security **15**(4), 1–28 (2013)
19. Gallagher, M.P., O'Connor, A.C., Kropp, B.: The economic impact of role-based access control. Planning report 02-1, National Institute of Standards and Technology (2002)
20. Gesmundo, A., Samardžić, T.: Lemmatisation as a tagging task. In: Proc. ACL 2012, pp. 368–372 (2012)
21. Gildea, D., Jurafsky, D.: Automatic Labeling of Semantic Roles. Computational Linguistics **28**(3), 245–288 (2002)
22. He, Q., Antón, A.I.: Requirements-based Access Control Analysis and Policy Specification (ReCAPS). Information and Software Technology **51**(6), 993–1009 (2009)
23. Hernandez, M.H., Laredo, J.A., Mandala, S., Ruan, Y., Sreedhar, V.C., Vukovic, M.: System and Method for Hybrid Role Mining, May 2, 2013. http://www.google.com/patents/US20130111583. US Patent App. 13/283,371
24. Hu, V.C., Ferraiolo, D., Kuhn, R., Friedman, A.R., Lang, A.J., Cogdell, M.M., Schnitzer, A., Sandlin, K., Miller, R., Scarfone, K.: Guide to attribute based access control (abac) definition and considerations (final draft). NIST Special Publication 800-162, National Institute of Standards and Technology, September 2013. http://csrc.nist.gov/publications/drafts/800-162/sp800_162_draft.pdf
25. Inglesant, P., Sasse, M.A., Chadwick, D., Shi, L.L.: Expressions of expertness: the virtuous circle of natural language for access control policy specification. In: Proc. SOUPS 2008, pp. 77–88 (2008)
26. Jurafsky, D., Martin, J.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Pearson (2009)
27. Karat, J., Karat, C.-M., Brodie, C., Feng, J.: Designing natural language and structured entry methods for privacy policy authoring. In: Costabile, M.F., Paternó, F. (eds.) INTERACT 2005. LNCS, vol. 3585, pp. 671–684. Springer, Heidelberg (2005)
28. Kennedy, C., Boguraev, B.: Anaphora for everyone: pronominal anaphora resoluation without a parser. In: Proc. Coling 1996, pp. 113–118 (1996)

29. Language-Independent Named Entity Recognition (2003). http://www.cnts.ua.ac.be/conll2003/ner
30. Manning, C.D.: Part-of-speech tagging from 97% to 100%: is it time for some linguistics? In: Gelbukh, A.F. (ed.) CICLing 2011, Part I. LNCS, vol. 6608, pp. 171–189. Springer, Heidelberg (2011)
31. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The stanford CoreNLP natural language processing toolkit. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55–60 (2014)
32. Meneely, A., Smith, B., Williams, L.: iTrust electronic health care system: a case study. In: Software System Traceability (2011)
33. Minimum security requirements for federal information and information systems. Technical report, National Institute of Standards, March 2006. FIPS Pub 200
34. Molloy, I., Chen, H., Li, T., Wang, Q., Li, N., Bertino, E., Calo, S., Lobo, J.: Mining Roles with Multiple Objectives. ACM Transactions on Information and System Security **13**(4), Article 36 (2010)
35. Neff, M.S., Byrd, R.J., Boguraev, B.K.: The Talent System: TEXTRACT Architecture and Data Model. Nat. Lang. Eng. **10**(3–4), 2004 (2004)
36. OASIS. Privacy Policy Profile of XACML v3.0. (2010). http://docs.oasis-open.org/xacml/3.0/xacml-3.0-privacy-v1-spec-cs-01-en.pdf
37. Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: An annotated corpus of semantic roles. Comput. Linguist. **31**(1), 71–106 (2005). ISSN: 0891-2017
38. Poibeau, T., Piskorski, J., Yangarber, R.: Information extraction: past, present, and future. In: Multi-source, Multilingual Information Extraction and Summarization, pp. 23–50. Springer, Heidelberg (2013)
39. Roeckle, H., Schimpf, G., Weidinger, R.: Process-oriented approach for role-finding to implement role-based security administration in a large industrial organization. In: Proceedings of the Fifth ACM Workshop on Role-Based Access Control, RBAC 2000, pp. 103–110. ACM, New York (2000)
40. Sagar, V.V.B.R., Abirami, S.: Conceptual modeling of natural language functional requirements. Journal of Systems and Software **88**, 25–41 (2014)
41. Sandhu, R., Coyne, E., Feinstein, H., Youman, C.: Role-based access control models. IEEE. Computer **29**(2), 38–47 (1996)
42. Schwitter, R.: Controlled natural languages for knowledge representation. In: Proc. CICLing 2010, pp. 1113–1121 (2010)
43. Sinha, A., Sutton Jr., S.M., Paradkar, A.: Text2test: automated inspection of natural language use cases. In: Proc. ICST, pp. 155–164 (2010)
44. Slankas, J., Xiao, X., Williams, L., Xie, T.: Relation extraction for inferring access control rules from natural language artifacts. In: Proceedings of the of 2014 Annual Computer Security Applications Conference (ACSAC 2014), New Orleans, LA (2014)
45. Socher, R., Bauer, J., Manning, C.D., Ng, A.Y.: Parsing with compositional vector grammars. In: Proc. ACL 2013 (2013)
46. Tan, L., Yuan, D., Krishna, G., Zhou, Y.: 21st SOSP 2007, pp. 145–158 (2007)
47. Terms of Service, Didn't Read project. http://tosdr.org/
48. Xiao, X., Paradkar, A., Thummalapenta, S., Xie, T.: Automated extraction of security policies from natural-language software documents. In: Proc. 20th FSE, November 2012